

AFRL-IF-RS-TR-2003-85
Final Technical Report
April 2003



**INTERNET PROTOCOL-HYBRID OPTO-
ELECTRONIC RING NETWORK (IP-HORNET): A
NOVEL INTERNET PROTOCOL-OVER-
WAVELENGTH DIVISION MULTIPLEXING (IP-
OVER-WDM) MULTIPLE-ACCESS
METROPOLITAN AREA NETWORK (MAN)**

Stanford University

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. J955

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2003-85 has been reviewed and is approved for publication.

APPROVED: 
PAUL SIERAK
Project Engineer

FOR THE DIRECTOR: 
WARREN H. DEBANY, Technical Advisor
Information Grid Division
Information Directorate

REPORT DOCUMENTATION PAGEForm Approved
OMB No. 074-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE APRIL 2003	3. REPORT TYPE AND DATES COVERED Final Jun 00 – Dec 02	
---	-------------------------------------	--	--

4. TITLE AND SUBTITLE INTERNET PROTOCOL-HYBRID OPTO-ELECTRONIC RING NETWORK (IP-HORNET): A NOVEL INTERNET PROTOCOL-OVER-WAVELENGTH DIVISION MULTIPLEXING (IP-OVER-WDM) MULTIPLE-ACCESS METROPOLITAN AREA NETWORK (MAN)	5. FUNDING NUMBERS C - F30602-00-2-0544 PE - 62301E PR - J955 TA - 21 WU - A1
--	---

6. AUTHOR(S) Leonid G. Kazovsky, Ian White, Matt Rogge, Kapil Shrikhande, Erie Hu, and Yu-Li Hsueh
--

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Stanford University Electrical Engineering Optical Communications Research Laboratory 350 Serra Mall, Room 362 Stanford California 94305	8. PERFORMING ORGANIZATION REPORT NUMBER
---	---

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Advanced Research Projects Agency AFRL/IFGA 3701 North Fairfax Drive Arlington Virginia 22203-1714	525 Brooks Road Rome New York 13441-4505	10. SPONSORING / MONITORING AGENCY REPORT NUMBER AFRL-IF-RS-TR-2003-85
--	---	--

11. SUPPLEMENTARY NOTES AFRL Project Engineer: Paul Sierak/IFGA/(315) 330-7346/ Paul.Sierak@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.	12b. DISTRIBUTION CODE
---	-------------------------------

13. ABSTRACT (Maximum 200 Words) After significant funding reduction this research program: pursued the design, implementation, and optimization of MAC protocols for variable sized IP packets; and experimentally investigated a laboratory model of a 2 fiber bi-directional path switched ring implementation of the HORNET architecture. The application arena for a IP-HORNET is the metropolitan area.

14. SUBJECT TERMS Optical Ring Networks, Optical Path Switched Networks, IP-HORNET, Metropolitan Optical Networks	15. NUMBER OF PAGES 322
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL
--	---	--	---

Table of Contents

1	Introduction and Motivation	1
1.1	Next-Generation Metro Area Networks	1
1.2	Current Approach: SONET Ring	3
1.3	Alternate Approach: Ethernet in the Metro Area	8
1.4	Emerging Solution: Resilient Packet Ring	9
1.5	RPR-over-WDM	11
2	<i>HORNET</i>: a Next-Generation Network	15
2.1	<i>HORNET</i> Architecture	15
2.2	<i>HORNET</i> Subsystems	17
2.2.1	Fast-Tunable Packet Transmitter	19
2.2.2	Asynchronous Packet Receiver	20
2.2.3	Linear Optical Amplifier	21
2.3	<i>HORNET</i> Media Access Control (MAC)	22
2.3.1	Potential Solutions	23
2.3.2	The <i>HORNET</i> MAC Protocol	26
2.4	<i>HORNET</i> Fairness Control	30
2.4.1	Unfairness of the <i>HORNET</i> Architecture	30

2.4.2	<i>HORNET</i> Fairness Control Protocol: <i>DQBR</i>	34
2.5	<i>HORNET</i> Survivability	38
2.5.1	Conventional Survivable Architectures	38
2.5.2	<i>HORNET</i> Survivable Architecture: 2FBPSR	42
2.6	<i>HORNET</i> Control Channel	46
2.6.1	Control Channel Frame Length	46
2.6.2	Control Channel Transmission	50
2.7	Control Channel Frame Synchronization	53
2.7.1	Dispersion Management for Control Frame Alignment	53
2.7.2	Control Channel Frame Synchronization Protocol	57
2.8	Network Reconfigurability in <i>HORNET</i>	65
2.8.1	Dynamic Traffic in the Metro Area	65
2.8.2	Necessary Technologies for Dynamic Networks	68
2.9	Quality of Service on <i>HORNET</i>	71
2.9.1	Motivation for Constant Bit Rate	71
2.9.2	Circuits over <i>HORNET</i> (CoHo)	72
2.9.3	Potential Reservation and Scheduling Mechanisms for <i>HORNET</i>	74
2.9.4	A Source-based Reservation Mechanism Using Broadcast	77
2.10	Quality of Service Summary	106
3	<i>HORNET</i> Network Simulations	107
3.1	Introduction	107
3.2	<i>HORNET</i> Simulator Design	108
3.2.1	Basic Concepts of the Simulator	108
3.2.2	Variable Packet Sizes	111
3.2.3	Segmentation and Re-assembly on Demand	113
3.2.4	<i>HORNET</i> Overhead	114

3.3	Optimal Control Channel Frame Size	116
3.4	Segmentation and Reassembly On Demand (SAR-OD)	118
3.5	DQBR Performance Simulations	120
3.5.1	DQBR Measured Fairness Performance	120
3.5.2	DQBR Performance Penalty	125
3.5.3	DQBR with Variable-Sized Packets	126
3.6	<i>HORNET</i> versus <i>RPR-over-WDM</i>	129
3.6.1	<i>RPR-Over-WDM</i> Simulator	129
3.6.2	Simulation Results	131
3.6.3	Equipment per Node Comparison	132
3.7	Summary	134
4	<i>HORNET</i> Subsystems	136
4.1	Introduction	136
4.2	Fast-Tunable Packet Transmitter	136
4.2.1	Tunable Semiconductor Laser	137
4.2.2	Laser-Tuning Controller	140
4.3	Asynchronous Packet Receiver	143
4.3.1	<i>HORNET</i> Research on Asynchronous Packet Receivers	144
4.3.2	Research on Digital Asynchronous Packet Receivers	146
4.4	Linear Optical Amplifier	147
4.4.1	EDFA Dynamics	149
4.4.2	Gain-Clamped EDFAs	153
4.4.3	Gain-Clamped Semiconductor Optical Amplifiers	155
4.4.4	Transient-Controlled EDFAs	157
4.5	Summary	157

5	<i>HORNET</i> Testbed: Experimental Demonstrations	160
5.1	Testbed Description	160
5.2	Protocol Demonstrations	161
5.2.1	<i>HORNET</i> Media Access Control Protocol and Survivability	161
5.2.2	Control Channel Frame Synchronization Protocol	167
5.2.3	Data and Control Processing Module: single-PCB implementation	174
6	Deployment Issues	179
6.1	Introduction	179
6.2	<i>HORNET</i> Power Budget Analysis	179
7	Conclusions	191
7.1	Accomplishments	191
7.2	Key Lessons Learned	194
7.3	Future Work	197
7.4	Final Thoughts	198
	Appendix - Included Papers	199
	Bibliography	295

List of Tables

1.1	SONET data rate hierarchy.	4
2.1	E[TTL] for different applications and dependence of MST on E[TTL]	90
5.1	HW implementation data paths	176
6.1	Estimated loss for components in a HORNET node	184

List of Figures

1.1	SONET uses a time-division-multiplexing (TDM) hierarchy to connect source and destination over a high capacity optical link.	5
1.2	SONET ring with add-drop multiplexers, which add and drop three STS-3 streams from the OC-48 optical ring.	6
1.3	RPR uses a bi-directional ring network with packet switches in all nodes.	10
1.4	Packet add/drop multiplexer in the RPR node. O/E = optical to electrical converter; E/O = electrical to optical converter.	10
1.5	When a cut occurs on an RPR ring, the nodes switch the traffic away from the cut. E/O = electrical to optical converter. O/E = optical to electrical converter.	11
1.6	<i>RPR-over-WDM</i> node when 2 wavelengths are used (only one of the two directions is shown).	12
1.7	In addition to receiving and transmitting their own traffic, RPR nodes must receive, switch, and re-transmit the traffic coming from upstream nodes and going to downstream nodes. E/O = electrical to optical converter. O/E = optical to electrical converter.	13
2.1	The <i>HORNET</i> architecture is a bi-directional wavelength routing ring network with tunable transmitters in each node.	16

2.2	(a) The tunable transmitter in Node n sends packets on the wavelength received by the destination node. (b) The packets pass through all intermediate nodes without being processed. (c) Only the destination node processes the packets because wavelength routing is used.	18
2.3	A collision occurs when a transmitter inserts a packet on a wavelength that is currently carrying a packet through the node.	22
2.4	The control channel conveys the availability of the wavelengths during a framed time period.	26
2.5	This cumulative distribution function (CDF) of IP packet sizes on a particular link measured by NLANR shows that packets range from 40 bytes to 1500 bytes.	28
2.6	After receiving the packet segments, the node queues them in separate queues sorted according to the source node. After the entire packet is received, it is passed onto the packet switch.	30
2.7	The <i>HORNET</i> ring unwrapped while focusing on the wavelength received by Node $N-1$. Nodes closer to Node $N-1$ have more difficulty sending packets to Node $N-1$ than the nodes further upstream.	31
2.8	Since a node has a difficult time sending packets to nodes near it, the VOQs associated with those destinations are likely to have a large backlog.	32

2.9	DQBR operation: (a) A node monitors the requests on the upstream control channel coming from the downstream nodes. (b) The node increments the RC counters for any requests it sees. (c) When a packet arrives in a VOQ, the value in the corresponding RC counter is stamped onto the packet as the WC. The packet cannot be transmitted during the availability on wavelength k because the WC value is nonzero. (d) The WC counter is decremented for every availability that passes by on the corresponding wavelength. (e) The packet can now be transmitted. (f) When a packet arrives at VOQ m , the value from RC_m is moved into the WC and stamped onto the packet. The packet will have to allow three empty frames on Wavelength m to pass before it can be transmitted.	37
2.10	The architecture of a 2-fiber unidirectional path-switched ring network. This architecture is commonly deployed in metropolitan area SONET networks.	39
2.11	The architecture of a 4-fiber bi-directional line-switching ring network.	40
2.12	(a) Under normal operating conditions, the protection fibers and equipment are unused. (b) When a fiber cut occurs, the two optical switches surrounding the cut are activated in order to switch the traffic away from the cut.	41
2.13	The protection equipment can also be activated using a span switch to restore an interrupted connection.	41
2.14	(a) Under normal operating conditions, a node attempts to load-balance its traffic while using all available bandwidth in both directions. (b) When Node 32 learned of the cut, it determined that to reach Nodes 0 through 24 it must use the counter-clockwise ring.	44
2.15	Information contained within each control channel frame.	47

2.16	This estimated CDF is based on the collection of data for various links as measured by NLANR.	48
2.17	(a) The minimum possible overhead for a <i>HORNET</i> network with a packet size CDF shown above. (b) Figure (a) zoomed in to focus on lengths less than 100 bytes.	51
2.18	(a) Optical packets on a WDM system with 64 payload wavelengths <i>before</i> propagating through single mode fiber. (b) Optical packets on a WDM system <i>after</i> propagating through single mode fiber. The fiber dispersion causes the packets to drift across control frame boundaries. W_c = control channel wavelength.	55
2.19	The Start-of-Frame Indicators and the packets on the payload wavelengths can become misaligned as they pass through the nodes.	58
2.20	Time lapse image on a digital oscilloscope of the random misalignment between the control channel frames and optical packets after the packets have propagated through (a) one node, and (b) two nodes. The instrument is triggered by the detection of the SOF indicator in the receiver.	59
2.21	Calculated probability density function of accumulated jitter after 8, 16, and 32 nodes of propagation.	60
2.22	The control channel and the payload packets passing through the nodes pass through two different paths. S_n denotes splice locations, L_n denotes fiber lengths.	61
2.23	The output phase of the PLL and the delay states are controlled by the node to provide perfect control channel frame synchronization.	62
2.24	The setup for the <i>lab cal</i> procedure.	63

2.25	(a) During the <i>IS-cal</i> , the node cycles its process clock phase through all possible phases. In this example, only four phases are used (0 , $\frac{\pi}{2}$, π , and $\frac{3\pi}{2}$). (b) For the first sampling phase, the node perceives that the calibration packet front edge arrives one clock cycle before the SOF indicator flag. (c) For the third sampling phase, the node perceives that the calibration packet front edge and the SOF indicator flag arrive simultaneously.	66
2.26	Logical schematic of the function of the Reconfigurable Optical Drop needed in <i>HORNET</i> . M is the maximum number of wavelengths the node requires.	69
2.27	Typical R-OADM design proposed in many other research projects. The expensive optical components and the W photonic receivers make the design impractical for a metro network.	70
2.28	Functional block diagram of the <i>HORNET</i> node.	78
2.29	Service rate of VOQ-10 at all nodes: for source-clearing and destination-clearing.	84
2.30	Service rate of VOQs at nodes 2, 13 and 28, for greedy (a) and non-greedy (b) protocols.	86
2.31	Max-Min fairness exhibited by the protocol: one hot-spot destination, multiple sources case	87
2.32	Service rate and utilization of the non-greedy protocol and dependence on $E[\text{TTL}]$	88
2.33	Building blocks of the reservation and scheduling sub-system	92
2.34	State machine diagram of the algorithm used inside the reservation and scheduling sub-system (and implemented in VHDL)	97
2.35	Sample reservation on Ring Network	102
2.36	Corresponding reservation tables for sample reservation	102

3.1	Diagram of the <i>HORNET</i> simulation architecture. The " T_n " represents packets that arrived to the nodes' VOQs during time step n . In the diagram shown here there is no propagation delay between nodes (i.e. the number of columns in the availability array equals the number of nodes).	109
3.2	Simulated performance of <i>HORNET</i> networks with 33 nodes and 33 wavelengths and with 50 nodes and 50 wavelengths.	111
3.3	A cumulative distribution function of packet sizes modelled by the simulator.	114
3.4	This graph shows the penalty incurred for the use of variable-sized packets and 16-byte packet headers.	116
3.5	Impact of packet size distribution on overhead, and thus performance. (a) Distribution 1, which is similar in average packet size to previously shown distributions. (b) Distribution 2, which has a smaller mean packet size. (c) Overhead measured by the simulator for the two distributions. The minimum lines in (c) are the calculated overhead if no packets are segmented.	117
3.6	Simulated performance of <i>HORNET</i> with control frame sizes of 40 bytes, 56 bytes, 64 bytes, and 200 bytes. As predicted, using a 64-byte control channel frame results in the best performance.	118
3.7	This graph shows the advantage of using SAR-OD instead of automatically segmenting all packets into small, fixed-sized cells. The network in the simulation has 33 nodes and 33 wavelengths.	119
3.8	Throughput in the nodes' VOQs that use Wavelength 18 for all nodes on the network.	121

3.9	Throughput divided by load on the nodes' VOQs that use Wavelength 18. Nodes 10 and 11 are sending a large amount of traffic to Node 18, while the other nodes are only sending light amounts of traffic.	121
3.10	Throughput for VOQ number 18 for the 25 nodes on a <i>HORNET</i> network. VOQ number 18 corresponds to Wavelength 18, which is received by Node 18. The total network load for Wavelength 18 is 1.5 times its capacity. There is enough propagation delay between nodes to hold 50 control frames.	123
3.11	<i>Throughput divided by load</i> for VOQ number 18 for several nodes. VOQ number 18 corresponds to Wavelength 18, which is received by Node 18. The graph shows that with DQBR, all nodes have the same ratio of <i>throughput to load</i> , thus proving that DQBR solves the unfairness problem. In this simulation, the load on VOQ 18 in Node 10 is 9.33 Gb/s, and the load on VOQ 18 in Node 11 is 4.67 Gb/s. All other nodes have only a small load.	124
3.12	Average packet latency in each <i>HORNET</i> node for the unbalanced traffic case (Nodes 10 and 11 have a heavy traffic load for Node 18 while all others have light traffic).	125
3.13	Packet loss probability in each <i>HORNET</i> node for the unbalanced traffic case (Nodes 10 and 11 have a heavy traffic load for Node 18 while all others have light traffic).	126
3.14	Throughput for VOQ number 18 for several nodes for the following cases: no fairness control; DQBR without considering SAR-OD (DQBR1); and DQBR while considering overhead due to SAR-OD (DQBR2). VOQ number 18 corresponds to Wavelength 18, which is received by Node 18.	128

3.15	Simulated performance of <i>HORNET</i> and <i>RPR-over-WDM</i> on a 61-node bi-directional ring network. <i>RPR-over-WDM</i> is simulated with both 13 and 14 wavelengths.	131
3.16	Comparison of the number of transmitters and receivers in each node for <i>HORNET</i> and <i>RPR-over-WDM</i> for varying number of network nodes.	133
4.1	The tunable packet transmitter subsystem contains three components: the tunable laser, the laser-tuning controller, and the data modulator.	138
4.2	The Sampled-Grating DBR laser.	140
4.3	The laser-tuning controller for the fast-tunable packet transmitter used in <i>HORNET</i>	141
4.4	The design of the nonlinear clock extraction technique, which re-creates the perfectly synchronized clock tone from the incoming data.	146
4.5	The total optical power at any location in the asynchronous WDM link in <i>HORNET</i> is random. W_0 = Wavelength 0.	148
4.6	(a) Design of a typical EDFA. (b) 3-level energy structure of an EDFA.	150
4.7	The gain of an EDFA changes when the input power changes. In this experiment, the peak power on Wavelength 1 is 9.5 dB higher than the peak power on Wavelength 2.	153
4.8	Design of a gain-clamped EDFA.	154
4.9	The gain-clamped SOA maintains constant gain under dynamic input conditions, whereas the conventional EDFA has dynamic gain.	156
5.1	Generic Testbed Diagram	161
5.2	The <i>HORNET</i> experimental testbed	162
5.3	Photograph of the electronics in a <i>HORNET</i> testbed node.	163

5.4	(a) Packets transmitted to Node 4 from Node 1. After a cut, Node 1 must send packets to Node 4 in the CCW direction. (b) Packets transmitted from Node 3 to Node 1.	164
5.5	(a) Restoration delay for the path from Node 3 to Node 1; (b) Transition of routes in Node 3 after the cut is reported as fixed (delay is only due to differences in fiber length along paths).	166
5.6	Testbed setup for MAC protocol Demonstration	167
5.7	The control channel path and the payload wavelength path. S_n denotes splice locations, L_n denotes fiber lengths.	169
5.8	The output phase of the PLL and the delay states are controlled by the node to provide perfect control channel frame synchronization. . .	169
5.9	The setup for the <i>lab cal</i> procedure.	170
5.10	During the <i>IS-cal</i> , the node measures the time difference between the arrival of the SOF indicator flag and the packet front edge by cycling its process clock phase through all possible phases.	171
5.11	The setup of the <i>IS-cal</i> procedure for a node downstream of a previously calibrated node.	172
5.12	The location of the samples for all phases for the two incoming waveforms in the <i>IS-cal</i> procedure of the second node.	173
5.13	(a) Alignment of retransmitted control channel SOF indicator with a packet passing through the node before the <i>IS-cal</i> ; (b) After the phase adjustment portion of the <i>IS-cal</i> ; (c) After the complete <i>IS-cal</i>	173
5.14	Time-lapse image of the retransmitted control channel and packets after two nodes of propagation. (a) Random misalignment with no frame synchronization protocol. (b) Perfect alignment with the protocol.	173
5.15	HORNET data and control processing module Printed Circuit Board	174

5.16	Photo of HORNET data and control processing module Printed Circuit Board	177
5.17	HORNET Block diagram with data and control processing module Printed Circuit Board	178
6.1	Block diagram of the HORNET node	180
6.2	OSNR vs. Number of nodes propagated for different amplifier parameters (left) $P_{\text{sat}} = 20\text{dBm}$, $\text{NF}=5\text{dB}$ (right) $P_{\text{sat}} = 10\text{ dBm}$, $\text{NF}=8\text{dB}$	187
6.3	Calculated BER versus node number based on Gaussian noise assumption for a 10Gbps system for different amplifier parameters (left) $P_{\text{sat}} = 20\text{dBm}$, $\text{NF}=5\text{dB}$ (right) $P_{\text{sat}} = 10\text{ dBm}$, $\text{NF}=8\text{dB}$	188
6.4	Calculated OSNR and BER versus node number for 10Gbps system if the loss of reconfigurable drops can be reduced to 1dB ($P_{\text{sat}} = 20\text{dBm}$, $\text{NF}=5\text{dB}$)	189
6.5	Calculated OSNR and BER versus node number for 10Gbps system with -0.2dB gain error on each amplifier ($P_{\text{sat}} = 20\text{dBm}$, $\text{NF}=5\text{dB}$) .	190

Chapter 1

Introduction and Motivation

1.1 Next-Generation Metro Area Networks

In the early days of Internet photonics research, the metropolitan area networks did not attract a lot of attention. Most companies and research institutes were focused on pushing the capacity of photonic links into the terabit per second (Tb/s) realm. However, a noticeable shift occurred just before the turn of the century, as it became apparent that the ultra-high capacity backbone links would not necessarily be useful if a bottleneck existed in the metropolitan area between the Internet backbone and the user. The last few years of investment in metropolitan area networking has resulted in a few competing architectures aimed at cost-effective solutions that deliver moderate capacity. However, metropolitan area networks are only at the beginning of a major evolution towards a new age of end users and applications.

A metropolitan area network of the near future will be characterized by the quantity and diversity of its end users, by the high percentage of randomly fluctuating packet-based data traffic, and by the incredible load placed on the network at peak usage times. End users may range from today's typical users, such as home and business users, to futuristic users such as automobiles, appliances, hand-held devices,

and other things not yet imagined. It is no longer unthinkable for over a million users to simultaneously access the same metro network in the near future. With this many users, it is reasonable to believe that metro networks will be forced to support capacities of up to and beyond 1 Tb/s. Additionally, it is safe to assume that a large portion of this traffic will be bursty, packet-based data traffic, as is common with Internet traffic.

Next-generation metro networks will also be largely affected by a new Internet trend. In today's Internet, a large majority of users is getting a large majority of content from only a few providers, such as popular news organizations or dominant Internet Service Providers (ISPs). The result is that the majority of the traffic on the Internet squeezes through the same corner, or bottleneck, of the Internet, which of course results in slow downloads for users. A solution to this, called Web caching, has been proposed in the literature [13, 14], and has begun to appear commercially. With Web caching, commonly accessed content is cached closer to the end users, potentially in the metropolitan area network nodes. It helps to keep the load in the Internet more balanced and reduces download times for end users. Protocols have already been developed that allow networks, such as a metro network, to be aware of the content that all nodes in the network are caching, thus allowing the entire metro network to serve as a distributed cache [14]. With Web caching, when a metro node receives a request for commonly accessed content, it routes the request to a node on the metro network that is caching the content instead of routing the request to the original source of the content.

Web caching can ultimately have a very interesting impact on traffic patterns in metro networks. Currently, metro networks are thought of as collection and distribution networks. This means that they are used to collect traffic from local users and send it to the Internet backbone and to distribute the traffic from the backbone to the users. However, with Web caching, the percentage of intra-network traffic (traffic

from access node to access node) will increase dramatically. Adding to this effect is the new trend of distributed file and processor sharing. This Internet technology is most famous for the controversial exchange of music and video files, but has many other practical extensions as well. It is clear that this peer to peer technology will also increase the amount of intra-network traffic. It is conceivable that the combination of these two technologies, along with other new concepts such as increased wireless traffic within the metro area, will boost the level of intra-network traffic to the point where it is even a majority of the total network traffic.

In summary, next-generation metro networks will likely be as follows. There will be millions of end users simultaneously accessing the network, resulting in more than 1 Tb/s of load on the network. Traffic will be composed primarily of randomly fluctuating, bursty, packet-based data traffic, much of which may be intra-network traffic. Additionally, the market for metro network operators is much more competitive than that of Internet backbone operators, and hence the cost-effectiveness and efficiency of a network are crucial. Thus, a network architecture for next generation metropolitan area networks should *cost-effectively* support more than *1 Tb/s of bursty, packet-based data* traffic with *randomly distributed* source and destination node pairs.

1.2 Current Approach: SONET Ring

The current solution for today's metropolitan area networks is called Synchronous Optical Network (SONET). SONET was developed nearly two decades ago for the high-speed digital transmission of long-distance telephone calls. The operation of SONET is based on time division multiplexing (TDM) of tributary circuits according to a standardized hierarchy. The quantum unit in the hierarchy is the Synchronous Transport Signal-1 (STS-1) circuit, which has a bit rate of 51.84 Mb/s. When an STS-n channel is transported on an optical link, the optical data stream is referred

SONET Channel	Gross Data Rate
STS-1 (OC-1)	51.84Mb/s
STS-3 (OC-3)	155.52 Mb/s
STS-9 (OC-9)	466.56 Mb/s
STS-12 (OC-12)	622.08 Mb/s
(OC-18)	933.12 Mb/s
STS-24 (OC-24)	1244.16 Mb/s
STS-36 (OC-36)	1866.24 Mb/s
STS-48 (OC-48)	2488.32 Mb/s
STS-192 (OC-192)	9953.28 Mb/s

Table 1.1: SONET data rate hierarchy.

to as Optical Channel-n (OC-n). The minimum circuit size may change in the near future, as the industry is working on a new version of SONET that performs virtual concatenation, a technique that allows lower-bandwidth circuits to be provisioned (on the order of 1.5 Mb/s) for finer granularity. The current hierarchy of tributary bit rates in SONET is listed in Table 1.1.

The concept of SONET is illustrated in Figure 1.1. Tributaries are time division multiplexed together using byte interleaving at SONET multiplexers. Also, equipment called SONET add/drop multiplexers (ADMs) can drop and add TDM tributaries from a SONET pipe. It is important to understand the restrictions placed on the network with the use of SONET's TDM operation. In Figure 1.1, element *A* has an STS-3 circuit multiplexed onto link ABCD, and thus bytes from element *A* appear every fourth byte on link ABCD. If element *A* has no payload to send on the link, every fourth byte on link ABCD contains no payload. No other tributary circuit can use that bandwidth, even if they could benefit from it. A network system that uses statistical division multiplexing (SDM) can allow elements to use bandwidth that is not being utilized by the circuits that were originally allotted that bandwidth. This extra degree of flexibility, which can have tremendous benefits, cannot be built into the rigid hierarchy of SONET.

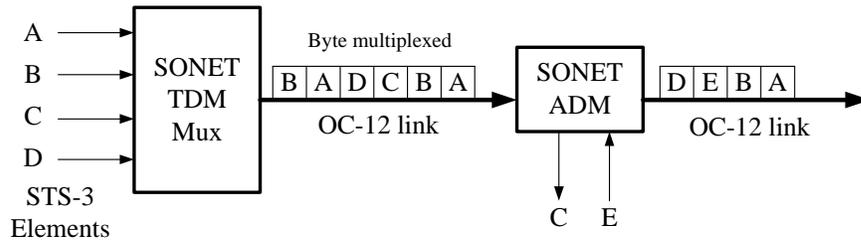


Figure 1.1: SONET uses a time-division-multiplexing (TDM) hierarchy to connect source and destination over a high capacity optical link.

Figure 1.2 illustrates how SONET can be applied to an optical ring network, which is the topology that is typically found in the metropolitan area. In the example in Figure 1.2, the four nodes are connected with each other by multiplexing STS-3 circuits onto the OC-48 optical fiber ring using SONET ADMs. Each ADM adds three STS-3 channels and drops three STS-3 channels for its respective node. Each of the three STS-3 channels represents a connection with one of the other three nodes on the network.

Figure 1.2 shows the logical operation of the SONET ring. In reality, SONET rings use multiple fiber cables. The advantage of multiple-fiber ring architectures is that there are two paths from every node to every other node. A fiber cut or equipment failure removes only one of those two paths, and thus the network can survive under such a circumstance. Typically, nodes on a SONET ring network, such as the one shown in Figure 1.2, transmit their SONET channels in both directions of the ring. Thus, nodes receive two sets of identical data, but they will only regard one of the two. If an event occurs that causes the primary data stream to disappear, the node simply diverts its attention to the secondary stream.

Although SONET rings are by far the most common architecture in today's metropolitan area networks, there are a number of very important disadvantages. First of all, as described earlier, the TDM operation of SONET can waste bandwidth

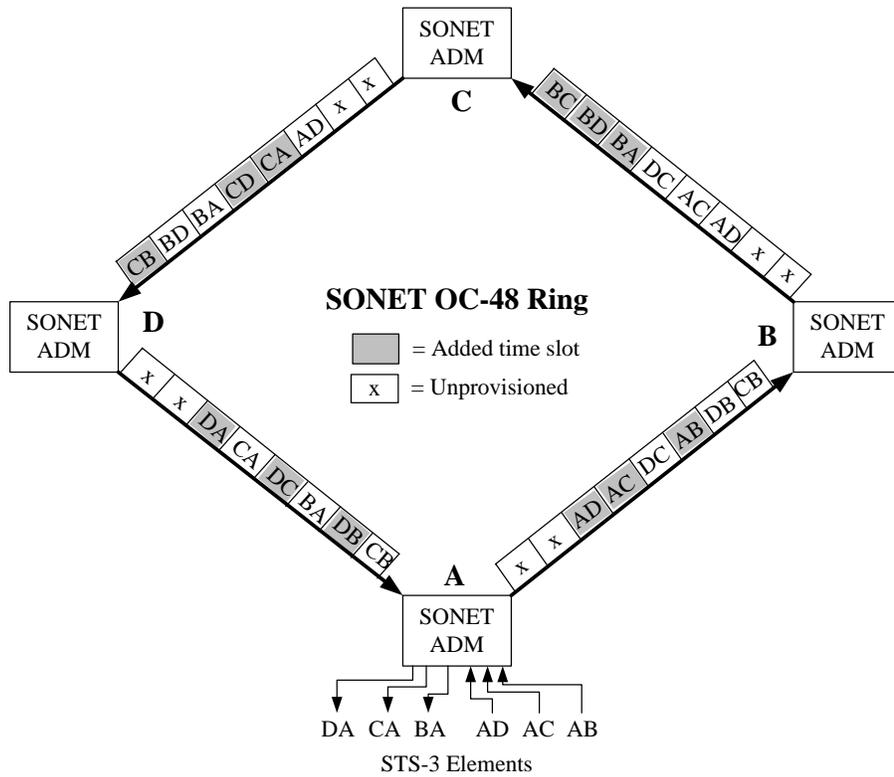


Figure 1.2: SONET ring with add-drop multiplexers, which add and drop three STS-3 streams from the OC-48 optical ring.

because if one node does not have data to send to a particular destination node, the corresponding TDM slots will go unused, even if another could make use of the extra bandwidth. Clearly, in a network environment that features randomly fluctuating, bursty, packet based data traffic, fixed-bandwidth TDM circuits are sub-optimal.

A second disadvantage is the difficulty of provisioning new circuits. To provision a new circuit between a pair of nodes, all of the SONET ADMs and multiplexers between the pair of nodes must be configured. This requires a good deal of planning and network management by technicians. Typically, consumers complain of delay times that range from 6 weeks to 6 months for the provisioning of a new circuit. In today's and tomorrow's quickly changing Internet world, this is unacceptable.

A third disadvantage is the wasted bandwidth and equipment that is used to maintain survivability. As described above, a SONET ring network transmits the data channels in both directions on the ring. Thus, twice the equipment is used, as well as twice the bandwidth (which implies twice the wavelengths in a WDM network). In other words, only half of the bandwidth and equipment in the network is utilized for working traffic. The other half is used for protection, which includes the transport of redundant copies of best-effort traffic and idle traffic. A better scheme is to protect only the traffic that must be protected (which is certainly a minority of the traffic), allowing the network to use all other bandwidth for best-effort traffic. This would result in more utilized bandwidth per dollar spent on equipment. However, this scheme requires far more flexibility than SONET is designed to provide.

A fourth flaw in SONET is the high price of SONET ADMs. The metropolitan area is a very competitive market, so new entrants are searching for inexpensive equipment. Compared to the new generation of high-speed data equipment (such as routers and Ethernet switches), SONET equipment is quite expensive. Thus, if a new solution emerged that used equipment similar to the less expensive data switching and routing equipment, it would certainly be a more attractive option than SONET.

1.3 Alternate Approach: Ethernet in the Metro Area

SONET's fundamental flaw is that it was designed for circuit switched telephony applications, and thus is sub-optimal for bursty, packet-based data, which is now the dominant traffic. Thus, it seems quite logical to replace the SONET networks with an architecture and protocols developed specifically for Internet data traffic. The most popular such network architecture and suite of protocols belong to Ethernet. Although Ethernet has commonly been considered only for local area networks (LANs), the bit rates defined within Ethernet have grown to 10 Gb/s, and may eventually reach 40 Gb/s.

Ethernet has several primary advantages over the SONET architecture presented above. Although it is intended to operate in a hierarchy similar to SONET, the tributaries are not restricted to the fixed-bandwidth TDM channel approach, and thus bandwidth utilization for statistically fluctuating traffic is much better in Ethernet. For this same reason, provisioning new circuits is much simpler. An operator can turn on a new 100 Mb/s Ethernet connection for a customer without re-engineering the entire network. Thus, provisioning time and costs should be much lower. Also, it is commonly accepted that Ethernet switches are much less expensive than SONET multiplexing equipment of the same capacity.

Given these advantages, it appears that Ethernet could be the new solution for the metro area. Thus, it should be no surprise that many in the networking industry are arguing for the deployment of Ethernet instead of SONET. However, a few very important points have been neglected in this argument. Ethernet is not designed to be implemented over a ring, which is the most common topology of currently deployed fiber infrastructures. Thus, a metro area Ethernet network would not take advantage of the ring for survivability purposes. In theory, if a link fails in an Ethernet ring,

the network will eventually discover this and find the alternate path, but it will likely take far too long since Ethernet does not have a survivability protocol optimized for a multi-fiber ring architecture. Also, Ethernet is not designed with the ability to handle quality of service (QoS) or global network fairness issues, which may be very important in metropolitan area networks [15].

1.4 Emerging Solution: Resilient Packet Ring

It is fair to say that Ethernet is close to being an optimal solution for today's metro networks, but falls short because of its inability to take advantage of the ring topology and because of its disregard for fairness and QoS. Thus, a modified version of Ethernet that is designed to compensate for these two shortcomings may be the best solution. It is this line of thinking that is behind the formation of the *IEEE Resilient Packet Ring (RPR) Working Group* and the *Resilient Packet Ring Alliance* [15]. RPR is a new data link layer protocol designed for metro area photonic ring networks. The RPR Working Group is attempting to use the virtues of Ethernet on a metro area *ring* architecture with survivability, fairness, and QoS.

The architectural concepts of RPR are illustrated in Figure 1.3 and Figure 1.4. RPR operates on a bi-directional optical ring network with a packet switch in each node. The packet switch, which functions as a packet add/drop multiplexer (packet ADM), is diagrammed in Figure 1.4. Packets that enter a particular node's input from the ring are either destined for that node or for a node further downstream. If the ADM determines that the packet is destined for its node, it drops the packet into the node. If the packet is not dropped, it is sent into the *transit* queue, which is a first-come-first-serve (FCFS) queue that holds the packets until they can be sent to the transmitter. Between the transit queue and the output transmitter is the add component of the packet ADM. Packets that are to be transmitted onto the

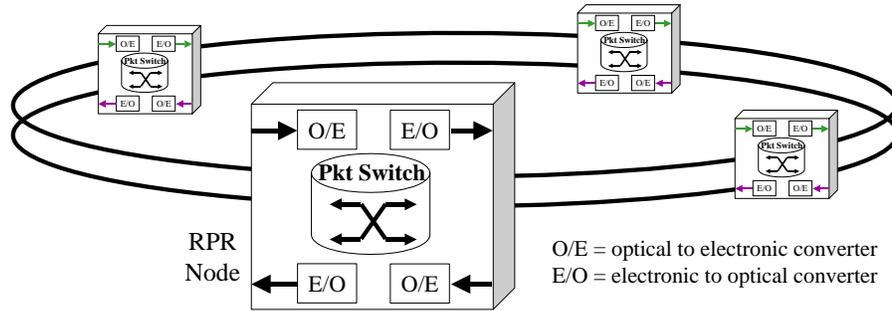


Figure 1.3: RPR uses a bi-directional ring network with packet switches in all nodes.

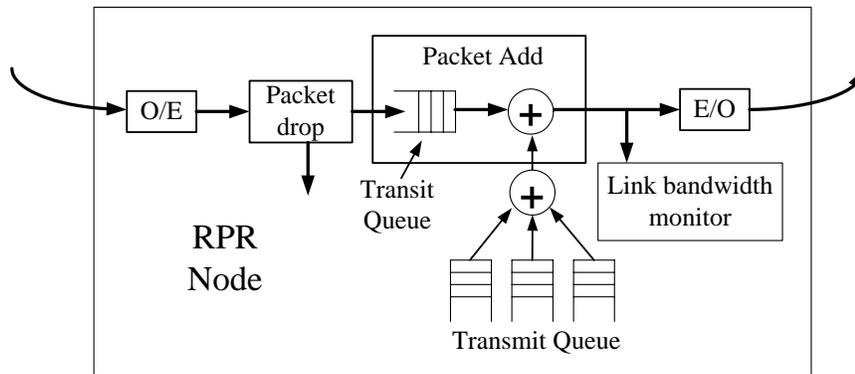


Figure 1.4: Packet add/drop multiplexer in the RPR node. O/E = optical to electrical converter; E/O = electrical to optical converter.

network by a particular node are queued in the *transmit* queue waiting to be sent to the transmitter. An arbitrator uses the RPR fairness algorithm to determine when to send packets from the *transit* queue and when to send packets from the *transmit* queue. Note that the packet ADM has an advantage over a traditional Ethernet switch because the Ethernet switch will not perform arbitration between packets passing through the node and packets being inserted onto the network.

Another disadvantage to using traditional Ethernet on a metro ring is that it would not properly utilize the ring architecture for survivability in the event that a fiber is cut or a node fails. RPR, on the other hand, is designed with survivability in

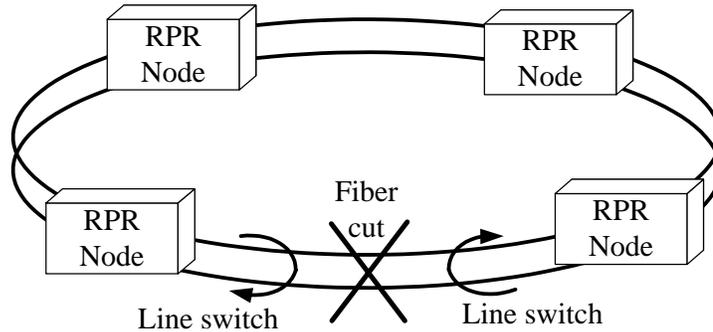


Figure 1.5: When a cut occurs on an RPR ring, the nodes switch the traffic away from the cut. E/O = electrical to optical converter. O/E = optical to electrical converter.

mind. When a cut or failure event occurs, the two nodes surrounding the location of the event will detect its occurrence. Each of the two nodes will then perform what is referred to as a line switch. A line switch wraps the traffic that is headed for the network cut and onto the other ring, which is carrying traffic away from the cut. This is depicted in Figure 1.5. Generally, network architectures have been designed with optical switches to perform the line switch, but it can also be implemented electronically. It is unclear today whether the RPR standard will specify whether the line switch should be optical or electronic.

1.5 RPR-over-WDM

The initial deployment of RPR will likely use only one wavelength in each of the two fiber rings. However, it is clear that to support the quickly increasing demand for bandwidth in the metropolitan area, RPR will be forced to scale its capacity using WDM. Such an architecture is referred to in this work as *RPR-over-WDM*. Although RPR is intended to remain transparent to the use of WDM, the appearance of the node changes, as shown in Figure 1.6. In this simple example, two WDM channels enter a node and are then demultiplexed. The packet ADM is now charged with

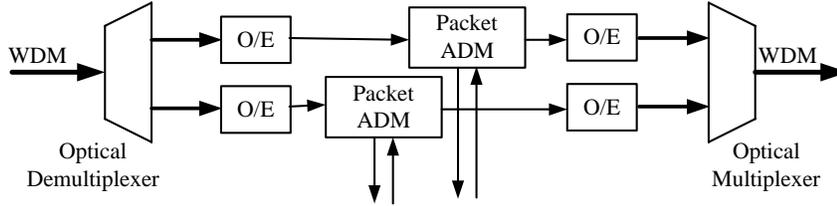


Figure 1.6: *RPR-over-WDM* node when 2 wavelengths are used (only one of the two directions is shown).

dropping packets from both of the two channels. A transit queue is used for each of the two wavelength paths, and the node has the ability to add packets into both of those paths. It is assumed in this work that a single transmit queue has the ability to add packets on any of the wavelength paths, though it is unclear if RPR-over-WDM will actually be implemented in this fashion.

As mentioned in Section 1.1, it is expected that in the near future metro networks will be forced to support capacities of up to or even beyond 1 Tb/s. Obviously, at such high capacities, a large number of wavelengths will be required. This can be troublesome for RPR-over-WDM, because if there are W wavelengths in each of the two rings, then each node will contain $2W$ receivers and $2W$ transmitters. Also, the packet ADMs must be designed to drop packets from W wavelength paths in each of the two directions, while the transmit queue should be designed to add packets on any of the W paths in each of the two directions. This is clearly expensive to design, especially considering that the data path will be operating at 10 Gb/s.

Clearly, the cost of the equipment in a node becomes quite high when the capacity of the network must scale to the capacities of the near future because of the excessive amount of photonic transmitters and receivers, and because of the complexity of the packet ADM. However, when looking at the operation of the node, it becomes apparent that adding more intelligence into the design brings about a much more cost-effective solution. Notice that so much photonic equipment and electronic

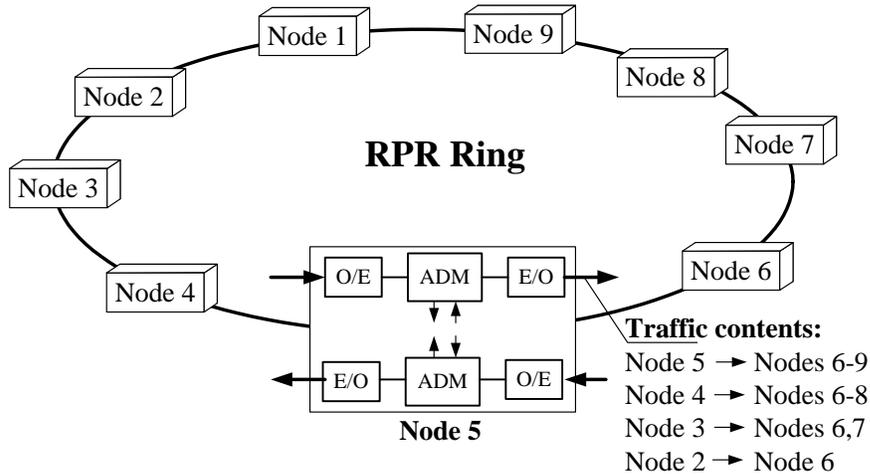


Figure 1.7: In addition to receiving and transmitting their own traffic, RPR nodes must receive, switch, and re-transmit the traffic coming from upstream nodes and going to downstream nodes. E/O = electrical to optical converter. O/E = optical to electrical converter.

complexity is required within each node because the node is receiving, switching, and re-transmitting a lot of traffic that comes from an upstream source and is going to a downstream destination. Consider the example of a 9-node bi-directional ring shown in Figure 1.7. Node 5 transmits traffic to Nodes 6, 7, 8, and 9 in the counter-clockwise direction. However, it also has to transmit traffic *from* Node 4 to Nodes 6, 7, and 8, and *from* Node 3 to Nodes 6 and 7, as well as traffic *from* Node 2 to Node 6. Thus, under uniform traffic conditions, only 40% of the traffic being transmitted by the node's transmitters came from this node. Additionally, only 40% of the packets coming through the packet drop stage were destined for this node. Though this situation is unreasonable, consider how unreasonable the situation becomes when the number of nodes on the ring increases to large values. It can be shown that for 25 nodes with uniformly distributed traffic, only 15% of the traffic transmitted by the node was originated by the node. The other 85% of the traffic is only passing through.

It is network designs such as this that have caused a high level of interest in

wavelength routing. Clearly, the cost of the node could be decreased significantly if traffic that originated upstream and is destined downstream would pass through the node optically. The photonic components would be required to operate on far less traffic, and thus could be reduced. The packet ADM would not process nearly as much traffic, and thus the complexity could be significantly reduced.

Unfortunately, however, RPR is not designed to utilize wavelength routing. A typical wavelength routing implementation would have each node receive a uniquely assigned wavelength. When a node wants to transmit a packet to a particular destination node, the transmitting node inserts the packet using a transmitter that emits on the *wavelength assigned to the destination node*. This implies that the node has transmitters for every wavelength in the network, even though the node is only terminating traffic on one (or maybe a few) wavelengths. However, the RPR media access control (MAC) protocol is only designed for the electronic packet ADM. The wavelength routing design requires a new MAC that controls an *optical packet ADM*. Unfortunately, an optical packet ADM similar to RPR's electronic packet ADM cannot be constructed because there is currently no practical means of queuing packets *optically*, and thus there can be no *transit queue* for the optical signals passing *through* the node. As a result, the new MAC protocol would likely need to be more complex because of the shortcomings of the optical packet ADM. Ultimately, however, if a MAC protocol and an optical packet ADM can be designed that utilize wavelength routing advantageously, it is clear that the cost of the network would decrease tremendously.

Chapter 2

HORNET: a Next-Generation Network

2.1 *HORNET* Architecture

As shown in the previous chapter, a new solution for metro networks that utilizes the advantages of wavelength routing can tremendously *decrease* the cost of a next-generation network. The solution requires a new method of transmitting packets that incorporates an *optical packet ADM*, as opposed to the electronic packet ADM proposed in RPR. A new MAC protocol also needs to be developed to control the optical packet ADM, as it differs significantly from the electronic packet ADM.

These requirements form the basis of the *HORNET* architecture. *HORNET*, which stands for Hybrid Opto-electronic Ring Network, utilizes fast-tunable packet transmitters, wavelength routing, and a novel MAC protocol to form an architecture that is more *cost-effective at high capacities* than any of its commercial predecessors. The generic design of the *HORNET* architecture is shown in Figure 2.1. *HORNET* is a 2-fiber bi-directional ring topology, so it can use the already deployed fiber optic infrastructure of today's SONET networks. Unlike SONET, however, *HORNET* uses

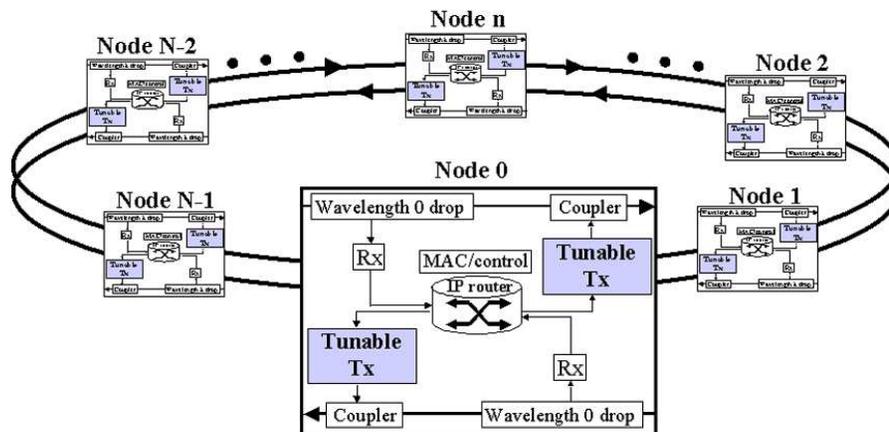


Figure 2.1: The *HORNET* architecture is a bi-directional wavelength routing ring network with tunable transmitters in each node.

all available bandwidth and equipment for working traffic. The nodes use their routing protocol to determine which is the best direction to transmit packets to certain destinations. Generally, the decision is made on the basis of load balancing and the node's position on the ring.

As Figure 2.1 shows, nodes use fast-tunable packet transmitters to insert packets onto the ring. The packets are coupled optically onto the ring using a wideband coupler (currently, a fast-tunable wavelength-selective multiplexer is not commercially available). A packet is transmitted on a wavelength that is received by the packet's destination node. A wavelength drop is used to drop one or more assigned wavelengths into each node. Thus, only the packets destined for a particular node are dropped into the node. All of the packets carried by the other wavelengths will pass through optically, such that the node does not receive or process them.

Consider the example illustrated in Figure 2.2. Node n wants to transmit two packets, one to Node 0 and one to Node 1. Assume Node 0 receives packets on Wavelength 0 and Node 1 receives packets on Wavelength 1. The transmitting node tunes its packet transmitter to Wavelength 0 and inserts the packet for that is destined

for Node 0. It then tunes its transmitter to Wavelength 1 and inserts the packet that is destined for Node 1. The packet for Node 0 does not match the wavelength drop of any of the other nodes, so it traverses the ring in optical form until it reaches Node 0, where it is finally dropped and processed. The same happens for the packet destined for Node 1.

In the *RPR-over-WDM* architecture, those two packets would have been received, converted to an electronic signal, sent through the packet ADM, and then retransmitted by every node between the source node and destination node. This example illustrates why *HORNET* nodes are less expensive than *RPR-over-WDM* nodes. The *RPR-over-WDM* nodes require significantly more equipment because they have to receive, process, and re-transmit all packets that pass through. In *HORNET*, a node only needs enough equipment to process the packets to and from its local users.

HORNET is not the only project investigating next-generation metropolitan area ring networks. Several other projects [16, 17, 18, 19, 20, 21, 22] have also in recent years investigated optical ring architectures for the metro area. Some of these projects even use the same wavelength routing concepts that are used in the *HORNET* architecture. However, the survivability scheme and the MAC protocol developed for *HORNET* are unique. The novel MAC protocol developed for *HORNET*, which is optimized for variable-sized packets and provides fairness control, is described in detail in Sections 2.3 through 2.4.

2.2 *HORNET* Subsystems

In every photonic link, there are three primary subsystems: *transmitters*, *receivers*, and *amplifiers*. The use of the three primary subsystems in *HORNET* must be investigated because *HORNET* is completely different from any previously deployed architecture. The design issues related to the three subsystems as they are used in

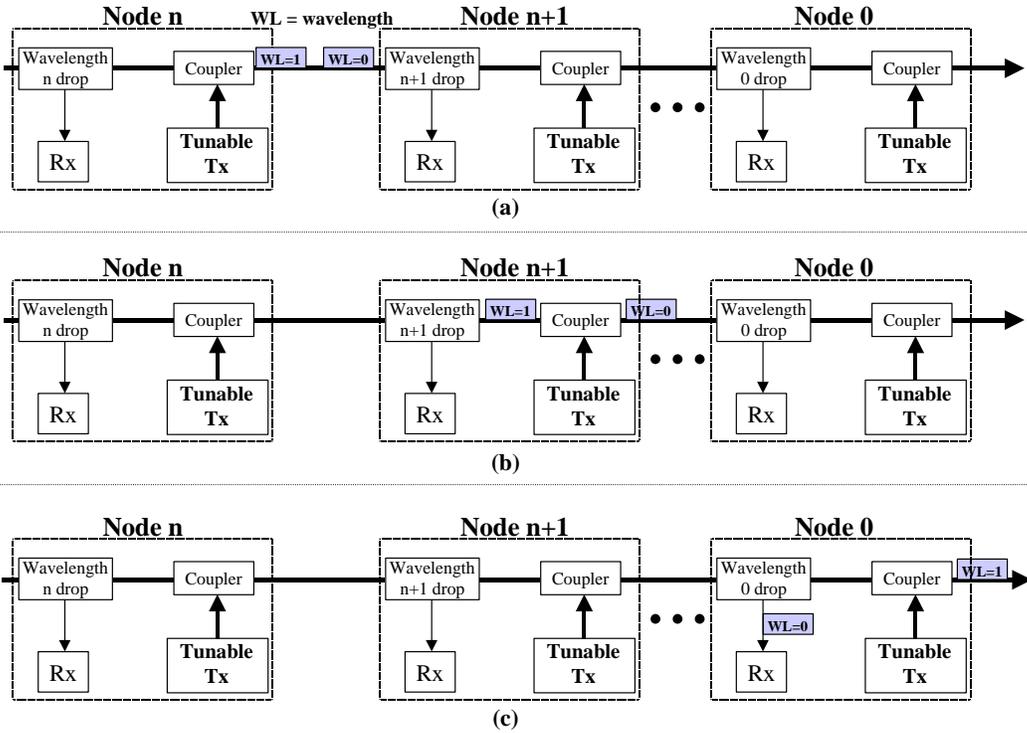


Figure 2.2: (a) The tunable transmitter in Node n sends packets on the wavelength received by the destination node. (b) The packets pass through all intermediate nodes without being processed. (c) Only the destination node processes the packets because wavelength routing is used.

HORNET are discussed in this section, and described in great detail in Section 4. As described, the novel architecture of *HORNET* imposes new constraints on all three subsystems.

2.2.1 Fast-Tunable Packet Transmitter

The transmitter in a *HORNET* node sends each packet on the *wavelength* that is *received* by the packet's *destination node*. Thus, the transmitter must have the ability to emit light on every wavelength in the network. One approach is to use W lasers in each node, where W is the number of wavelengths in the network, and where each laser emits at a unique wavelength. However, since W can be a large number in a typical *HORNET* network, the node would contain an excessive number of lasers, causing the cost of the node to be much higher than desired. The alternative approach, which is used by *HORNET*, is to use a laser with a controllable (i.e. tunable) output wavelength.

The requirements on the tunable transmitter are critical. The transmitter must be able to tune across a broad wavelength range, it must tune precisely enough to hit all network wavelengths, and it must tune incredibly quickly. Commercial tunable semiconductor lasers have been available in recent years that can meet the first two requirements. The third requirement, fast tuning, is the most difficult to achieve. However, as is shown in Section 4, a lot of research has recently been conducted on this subject, both within and outside of the *HORNET* project. The results from this thorough research show that it is possible to implement a fast-tunable packet transmitter. Ultimately, the commercial development of a fast-tunable packet transmitter appears to be inevitable.

2.2.2 Asynchronous Packet Receiver

Consider once again the fact that the transmitter in a *HORNET* node is sending consecutive packets on different wavelengths, and thus to different destination nodes. From the perspective of a receiver in each node, this implies that consecutive packets coming into the receiver are transmitted by different transmitters from different nodes. Therefore, the receiver operates asynchronously, as consecutive packets are spaced apart by unknown time differences and have random bit-phases with respect to each other. Also, the exact baud rate of each of the two consecutive packets may be slightly different (often within 0.001%).

In a synchronous network, the receiver can lock onto the bit phase and the exact baud rate at the link setup. From that point on, it only needs to remain in synchronization, and thus the amount of time required to synchronize is not important. However, in the case of the asynchronous packet receiver in *HORNET*, the receiver must perform the synchronization tasks at the arrival of every packet. Just as tuning time is overhead, the time required to achieve bit-synchronization is overhead because payload data cannot be properly received during those moments. Therefore, the asynchronous packet receiver for *HORNET* must be designed such that the bit phase and frequency are acquired in very little time, preferably in only a few bytes.

The issue of a fast-synchronizing packet receiver is a problem for optical packet switching networks as well as for *HORNET*, so research has been active on the subject in recent years. Analog solutions and digital solutions have been investigated, and are presented in detail in Section 4. It is clear from the results generated in these recent research efforts that the digital solution is the better of the two, especially with the continuing progress in high-speed digital electronics. Also, the research demonstrates that the issue of implementing an asynchronous packet receiver is not a problem. Nonetheless, it is not *yet* commercially available because there is currently no commercial use for such a fast-synchronizing receiver.

2.2.3 Linear Optical Amplifier

The typical optical amplifier subsystem in photonic networks is the EDFA. To provide the necessary output power for today's dense WDM systems, EDFAs must be operated in saturation. When the amplifier is not saturated, the gain is linear (the output power grows linearly with input power). In saturation, the gain is no longer linear, and thus it is dependent upon the input power (or output power) [23]. This is not a major problem for conventional networks because the instantaneous power at the input of the EDFAs in a link is held constant by using techniques such as scrambling, coding, and transmitting idle packets when no data packets are to be sent.

In contrast, in the *HORNET* network the instantaneous power at any point in the link is very dynamic. This is a result of the fact that nodes only transmit packets when they have a packet to transmit. Packet transmissions will thus occur at random. As a result, at any point on the link at any moment, the number of wavelengths carrying packets is random, and thus the optical power is random. The dynamic power on the network will affect the gain of the amplifier. As packets pass through the amplifier, the gain they receive will be dynamic, causing the amplitude of the packets at the output of the amplifier to be distorted. It is very difficult to design a receiver that can properly receive the bits in a packet with highly dynamic amplitude.

As a result, conventional EDFAs cannot be used in the *HORNET* network. The amplifiers for *HORNET* must provide linear performance (i.e. constant gain) when faced with dynamic conditions. Three solutions have recently emerged: gain-clamped semiconductor optical amplifiers [24], gain-clamped EDFAs [25], and transient-control EDFAs [26]. Each of these solutions, including the experimental demonstration of a gain-clamped semiconductor optical amplifier, is presented in Section 4. Ultimately, it is shown that linear optical amplifiers will be available for use in a commercially deployed *HORNET* network.

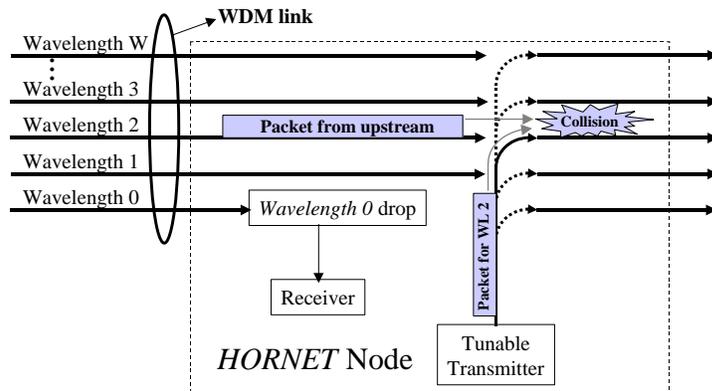


Figure 2.3: A collision occurs when a transmitter inserts a packet on a wavelength that is currently carrying a packet through the node.

2.3 *HORNET* Media Access Control (MAC)

Since the packet ADM process in the *HORNET* architecture is completely different from the ADM process of any preceding commercial network, a new suite of protocols for the data-link layer must be developed, starting with the MAC protocol. The primary function of the MAC protocol in *HORNET* is to prevent collisions at the point in the node where the transmitter inserts packets. Since the transmitter can insert a packet on any wavelength, and since most of the wavelengths are passing through the node without being terminated, a transmitter could insert a packet onto a particular wavelength that collides with another packet that was passing through the node on that wavelength. Figure 2.3 shows the occurrence of a collision. To prevent collisions, the MAC protocol should monitor the WDM traffic passing through the node, locate the wavelengths that are available, and inform the transmitter of which wavelengths it is allowed to use at a particular moment. As a result, the transmitter will not insert a packet on a wavelength that is currently carrying another packet through the node.

2.3.1 Potential Solutions

The most difficult aspect of the implementation of the MAC protocol is the method of obtaining the information about the status of the WDM wavelengths passing through the node (referred to as the *wavelength availability information*). One possible solution is to centralize all of the control and processing to one master node on the network. The scheduler treats the entire ring network like a high-capacity packet switch. The nodes place requests to the scheduler for transmissions, the scheduler determines the best schedule for all of the requests, and then the scheduler informs the nodes of when to transmit their packets. In principle, the scheduler appears to be an ideal solution because it can avoid collisions and it can determine the fairest possible schedule for the packet transmissions. However, in practice, the scheduler is incredibly difficult to implement because the network will be switching more than 1 Tb/s of total traffic. Also, difficulties arise because of the large geographical area across which the network is spread. A 1 Tb/s packet switch receives exactly current information and can convey the schedule to the transmitters immediately. This is certainly not the case in a network that may have a circumference of several tens of kilometers.

A second design that mimics a high-capacity packet switch is one in which the nodes send requests to the destination nodes for a transmission slot and then wait for an acceptance. Only if they receive a positive acceptance will they send a packet. This MAC protocol attempts to copy the operation of a packet switch like the one discussed in [27]. In theory, this ensures that collisions do not occur, and that the network is fair to all users. This is certainly more attractive than the centralized scheduler, but the problem of the geographic size of the network remains. In general, this scheme requires a time-slotted environment in which the slot duration is equal to the propagation time of light around the optical ring (on the order of a *millisecond*). Thus, the time necessary for requests and acceptances adds at least a few milliseconds

to the queuing delay. An excellent example of this competitive approach is thoroughly described in [17].

The remaining options for determining the wavelength availability information are all localized, meaning that the decision about when to transmit is made within the node based on information available within the node. The most obvious solution for this method is for the node to monitor the optical power on each wavelength as the wavelengths pass through the node. If the node measures no power on a wavelength for the duration of a packet, then it concludes that the transmitter can use that wavelength without causing a collision. In this design however, a problem arises because of the difficulty in optically monitoring the power on several tens of WDM wavelengths. One option is to tap a small percentage of power from the ring within the node and to send that WDM stream to a WDM channel monitor, which is composed of a scanning optical filter and a detector. However, because IP packets on the optical ring can be as short as 50 ns, the filter would have to scan the entire WDM transmission bandwidth at a rate of greater than 20 MHz. This is difficult to achieve today. A second option is to send the tapped WDM stream to a WDM demultiplexer, which has a photodetector at each of the outputs of the demultiplexer. This option is far more expensive than desired, however, because of the high cost of WDM demultiplexers and the large number of photodetectors and receiver circuits that are required.

The first design of the MAC protocol for *HORNET*, which is called *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) [28], accomplishes the same result as the above scheme, but with much lower equipment cost. In the CSMA/CA protocol, each network wavelength is assigned a corresponding unique RF frequency that has a higher value than the baud rate of the payload data stream. For example, if the payload data rate is 10 Gb/s, the lowest possible RF frequency must be significantly greater than 10 GHz (e.g. 15 GHz). When a node transmits a packet, it

frequency-multiplexes a subcarrier tone onto the packet, where the subcarrier uses the frequency that corresponds to the wavelength carrying the packet. A node determines which wavelengths are occupied with packets in the WDM traffic passing through the node by tapping a small amount of optical power from the WDM link and receiving it with a photodetector (no optical demultiplexing is used on the WDM signal). The resulting instantaneous power spectrum contains power at the subcarrier frequencies corresponding to the wavelengths carrying packets at the moment. An experimental demonstration is reported in [28]. Clearly, by using only one photodetector and by using RF demultiplexing instead of optical demultiplexing, costs can be significantly reduced as compared to the alternative methods of wavelength monitoring presented above.

Despite the apparent advantages of the CSMA/CA scheme, it was ultimately determined that the scheme was not the best. The main concern is the fact that the subcarrier frequencies lie well beyond the payload data baud rate. This is necessary for proper demultiplexing of the subcarrier tones and the payload data in both the subcarrier receiver and the payload data receiver. Thus, if the data rate is 10 Gb/s, the subcarrier tones may be required to be higher than 15 GHz. Because of the difficulty of building narrow-band filters at such high frequencies, and because of the large number of subcarrier frequencies used in a high capacity network, the band for the subcarriers may stretch over several GHz. As a result, for a bit rate of 10 Gb/s, the network nodes would likely be forced to use transmitters and receivers with a total bandwidth of 20 to 25 GHz, significantly increasing the cost of the network. Additionally, the combination of analog and digital signals significantly increases the cost of the transmitter and receiver. Consequently, it was determined that a different approach is necessary for a network that will scale to the anticipated high capacity of a next-generation metropolitan network.

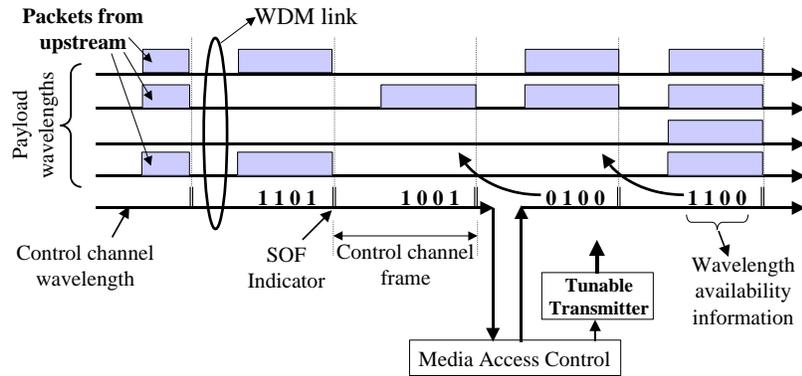


Figure 2.4: The control channel conveys the availability of the wavelengths during a framed time period.

2.3.2 The *HORNET* MAC Protocol

Possible replacements for the CSMA/CA protocol were investigated, some of which are described in [29]. Ultimately, the current approach for the *HORNET* MAC protocol evolved from these designs. *HORNET* uses a control channel similar in principle to the optical supervisory channel (OSC) typically used in conventional photonic networks. The primary function of the control channel is to convey the *wavelength availability information*. The control channel is carried on its own wavelength in the WDM network. That control wavelength is dropped and added in every node so that all nodes can process and modify the control channel. Figure 2.4 illustrates the operation of the control channel for the MAC protocol. The control channel is time-slotted into frames, much like any typical point-to-point high-speed data stream. The frame boundaries are demarcated with a *start-of-frame* (SOF) indicator byte. Within each frame is a bit-stream that conveys the wavelength availability information for the time period during the following frame. This allows the node to see one frame into the future. Potentially, the design could be modified to allow for more look-ahead if it is determined to be beneficial.

The wavelength availability bit-stream is a sequence of bits of length W , where W

is the number of wavelengths in the network. If bit w equals a '1,' then wavelength w is carrying a packet during the time period of the next control channel frame. A '0' bit indicates that the wavelength is available during the next control channel frame. A node sorts its queued packets into virtually separated queues called virtual output queues (VOQs) [30], the classic technique to avoid the head-of-line (HOL) blocking problem [31]. Each VOQ corresponds to a wavelength in the network. When a node reads the bit stream, it determines the *set of VOQs* that have a packet to transmit that overlaps with the *set of available wavelengths*. The node then determines which packet in the overlapping set it will transmit during the next frame. If the node decides to send a packet on wavelength w , it modifies bit w in the wavelength availability bit-stream to a '1.' All nodes clear the wavelength availability bit(s) corresponding to the wavelength(s) that they receive.

The framed format of the control channel makes the MAC protocol ideal for small, fixed-sized packets. However, Internetworking Protocol (IP) packets are inherently variable in size. Figure 2.5 shows a cumulative distribution function (CDF) of packet sizes measured on a typical IP link. This data is measured and reported by the National Laboratory for Applied Network Research (NLANR) [32]. As shown in the figure, IP packets have a very wide range of typical sizes, from 40 bytes to 1500 bytes.

Such a wide range of packet sizes is not at all compatible with a framed control channel with inflexible frame sizes. A simple solution exists for this problem that avoids any changes to the MAC protocol. As is done in IP-over-ATM, the variable-sized IP packets can be segmented into small, fixed-sized cells. The size of the segmented cell and the size of the control channel frame can be designed to match each other. Although the solution is simple, there is a significant drawback to the segmentation. Whenever a packet or a segment of a packet is transmitted, a header must be applied. The *HORNET* header includes information about the source and destination, allowance for transmitter tuning time and clock recovery time, and a

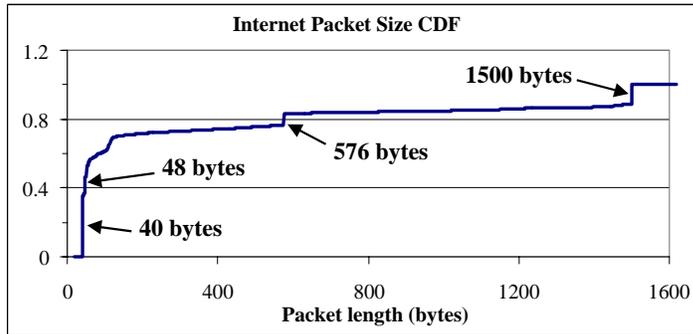


Figure 2.5: This cumulative distribution function (CDF) of IP packet sizes on a particular link measured by NLANR shows that packets range from 40 bytes to 1500 bytes.

few other items as well, as detailed in Section 2.6.1. Thus, a long packet, such as a 1500-byte packet, will have the *HORNET* packet header applied to it a large number of times. This will result in an excessive amount of overhead.

Adding only a small amount of intelligence into the MAC protocol can significantly reduce the overhead. Instead of automatically segmenting the packets such that each packet fits in one frame, the *HORNET* MAC protocol segments packets only when necessary. This modification to the MAC protocol is called *segmentation and re-assembly on demand* (SAR-OD). In this protocol, a node must begin to insert a packet in alignment with the beginning of the control frame. If the packet is longer than the control frame duration, the node *continues to transmit* the packet (without segmenting the packet and re-applying the header) until either the packet is complete or until the MAC protocol informs the transmitter that another packet is coming from upstream on the transmission wavelength. If a packet is coming from upstream while the node is transmitting a packet, the node *ceases the transmission* of its packet at the end of the last available frame (i.e. the one before the frame that is carrying the oncoming packet). At the end of the packet segment, the transmitter applies a byte that indicates that the segment is an incomplete packet. The node is now free to send

packets on different wavelengths while it waits for an opportunity to finish the packet it had begun. At the next opportunity, the node begins transmitting the segmented packet beginning from the location in the packet at which it was segmented. When the final segment of a packet is completely transmitted, the node finishes the packet with a byte that indicates that the packet is complete.

The receiver in a *HORNET* node has a slight amount of extra intelligence built into it to work with the SAR-OD protocol. The receiving process is illustrated in Figure 2.6. The receiver in a node maintains separate virtual queues for each node on the ring. When a packet arrives at the receiver, the receiver reads the *HORNET* packet header to determine the source node and then begins to write the payload of the arriving packet into the virtual queue corresponding to the source node. If the last byte of the segment indicates that the packet is complete, then the packet is transmitted out of the queue to the packet router to be sent to its final destination. If the last byte of the segment indicates that the packet is incomplete, the segment remains in the queue. The next segment arriving at the receiver from the same source node will belong to the same packet, and thus the receiver will store this segment at the queue location immediately following the previously received segment, just like a FCFS queue. When the packet is fully received, it will be sent to the node's packet router with the integrity of the IP packet completely preserved.

In the example shown in Figure 2.6, Node 0 is sending a long packet at Node n . Two of the segments already arrived to Node n and are stored in the queue waiting for the rest of the packet. After beginning the third segment, a packet from Node 1 to Node n passed through Node 0, forcing it to segment the packet again. After the packet from Node 1 has passed, Node 0 can begin the fourth segment of the packet for Node n . When the third and fourth segments arrive to Node n they will be stacked in the queue on top of the first two segments. If the fourth segment is the last, the final byte will indicate so, and Node n will pass the re-assembled packet on to the

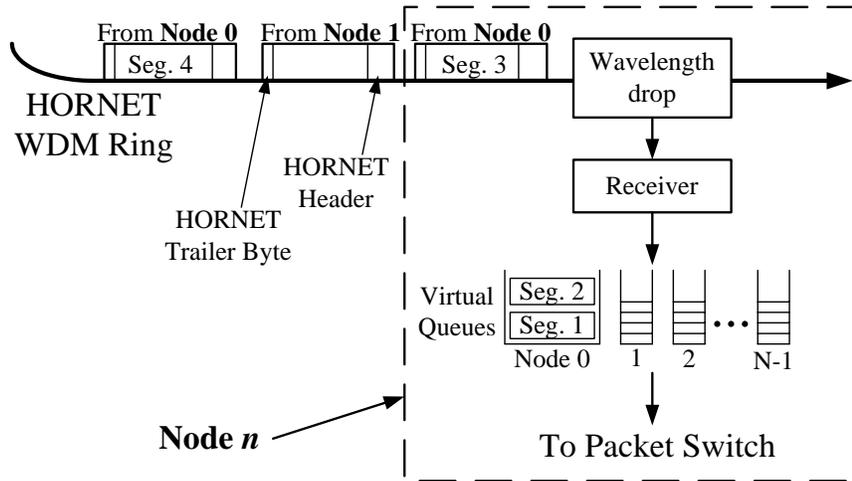


Figure 2.6: After receiving the packet segments, the node queues them in separate queues sorted according to the source node. After the entire packet is received, it is passed onto the packet switch.

packet switch.

2.4 HORNET Fairness Control

2.4.1 Unfairness of the HORNET Architecture

Although there are many advantages to using the bi-directional ring architecture for *HORNET*, there is one problem that arises because of it. Multiple-access ring networks are inherently unfair. The unfairness problem is most easily seen by considering only one of the network wavelengths and then unwrapping the ring, as is done in Figure 2.7. Consider the wavelength that is received by Node $N-1$ in Figure 2.7. When Node 0 wants to send packets to Node $N-1$, it is never blocked on the wavelength received by Node $N-1$. When Node 1 wants to send packets to Node $N-1$, it has to contend with (can occasionally be blocked by) the packets transmitted by Node 0 on the wavelength of Node $N-1$. Node 2 has to contend with Nodes 0 and 1, while

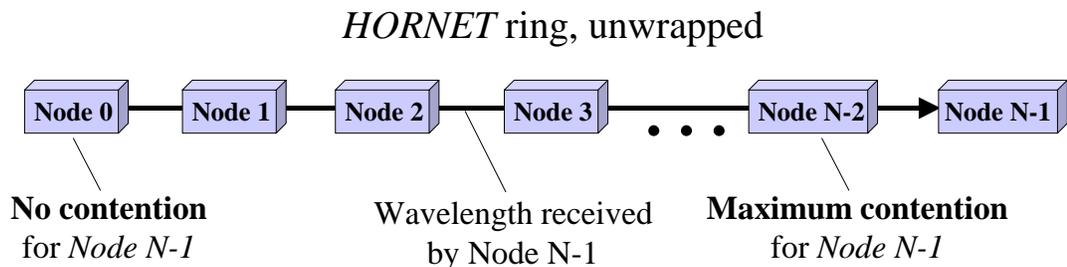


Figure 2.7: The *HORNET* ring unwrapped while focusing on the wavelength received by Node $N-1$. Nodes closer to Node $N-1$ have more difficulty sending packets to Node $N-1$ than the nodes further upstream.

Node 3 has to contend with Nodes 0, 1, and 2. This pattern continues around the ring to Node $N-2$, which has to contend with all of the nodes except Node $N-1$, making it more difficult for Node $N-2$ to transmit packets to Node $N-1$ than for the nodes further upstream. Thus, the network is *biased against* nodes closer to the destination.

To understand the result of the unfairness of the architecture, a closer look is helpful. Figure 2.8 shows what can happen in an unfair architecture. Consider a node on a network that has W wavelengths. The node maintains W VOQs, although the VOQ(s) corresponding to the wavelength(s) it receives is unnecessary. In one of the two directions, the node will transmit to the set of nodes $[n+1, |n+m|_N]$. Node $n+1$ is the closest, while Node $|n+m|_N$ is the furthest. If traffic on the ring is completely uniform, then the packet arrival rate at all of the VOQs will be the same. Since Node $|n+m|_N$ is the furthest away, Node n will typically encounter the least contention on the wavelengths received by Node $|n+m|_N$. Thus, the node has no difficulty inserting its packets destined for Node $|n+m|_N$, and the backlog in the queue remains small. In contrast, the wavelength(s) received by Node $n+1$ will in general be carrying a lot of packets from all of the upstream nodes, and thus may be occupied at nearly every instance. As a result, Node n has a very difficult time inserting packets onto the network for Node $n+1$, and thus the backlog in the VOQ

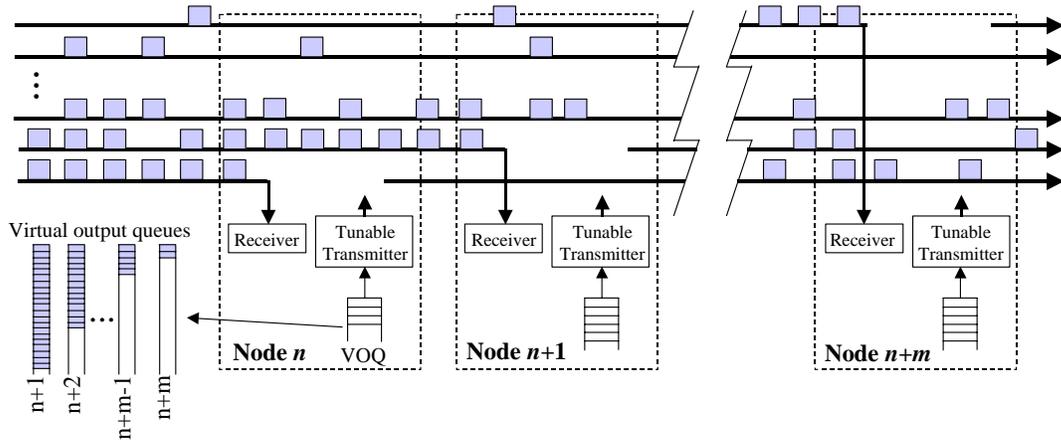


Figure 2.8: Since a node has a difficult time sending packets to nodes near it, the VOQs associated with those destinations are likely to have a large backlog.

can be very large, causing a higher latency and higher probability of packet loss for packets in that VOQ.

Clearly, it is undesirable for certain nodes to have a positional advantage over other nodes. Conventional ring architectures, such as *RPR-over-WDM*, do not have a difficult time dealing with the unfairness problem because the electronic packet ADM can buffer the packets that are passing through the node from an upstream source to a downstream destination. However, the *HORNET* architecture cannot buffer the packets passing through from upstream nodes because they are passing through optically. Thus, a protocol is required to force the upstream nodes to buffer their packets before transmitting them in such a way that allows downstream nodes to have an equal opportunity to insert their own packets.

Before designing a fairness control protocol, it is imperative to determine the goal of the protocol. Often when defining fairness, the network nodes are treated as users, and the fairness scheme is designed to give all nodes the same amount of bandwidth. In such a control scheme, if there are ten nodes on the network, and they all want a percentage of a wavelength's bandwidth arbitrarily greater than 10%, then the

network would allocate 10% of the available bandwidth for each node. Similarly, if three nodes are attempting to access the same wavelength, and one node wants 99% of the wavelength's bandwidth while the other two want 10% each, then the first node receives 80% and the other two receive their 10%. In this case, only one node suffers for the over-subscription of the wavelength. The other two get exactly what they desire.

However, in the *HORNET* ring network, nodes are not end users. In this work, it is argued that basing the fairness control on the principle of allocating bandwidth to nodes does not eliminate positional priority. Consider the following hypothetical example. A Web server attached to a node in the network hosts an Internet contest. Contestants are required to make a Web connection with the server that requires a significant amount of bandwidth. If many more users in one geographic area of the ring are intelligent enough to participate than in another area, then the contestants in the intelligent region are penalized. Consider the case where the wavelength received by the node hosting the contest's Web server can only support enough bandwidth for 1000 connections. Attached to one node in the network are 999 contestants who desire to participate, while two other nodes are hosting only 100 contestants each. In this case, many users on the node with the large number of contestants will be shut out of the competition, simply because of their geographic location.

In this work, fairness is considered on the basis of the end user, not the node. The fairness control protocol designed for this work attempts to transform the ring into one large FCFS queue. If a wavelength becomes oversubscribed, as in the previous example, then all nodes will suffer the same average packet latency, and all connections will have the same probability of being dropped. Thus, a *user's* position on the ring becomes completely irrelevant. There is no disadvantage to being located closer to the destination, and there is no disadvantage to living in an area densely populated with similar users.

2.4.2 *HORNET* Fairness Control Protocol: *DQBR*

The solution for the fairness control protocol developed during this project is a novel protocol established specifically for incorporation into the *HORNET* MAC protocol. It is called *Distributed Queue Bi-directional Ring* (DQBR) because the protocol attempts to transform *HORNET*'s bi-directional ring architecture into a distributed FCFS queue. The protocol is an adaptation of an older protocol called *Distributed Queue Dual Bus* (DQDB) [33, 34, 35, 36], which was created for single channel dual-bus metro networks of the 1980's. DQDB is also known as IEEE 802.6.

In *IEEE 802.6*, when a packet arrives at the front of a transmitter's queue, the node sends a request in the direction opposite to which the packet must be transmitted (upstream). The request consists of setting the *request bit* in the control information field of the current frame. The request passes through all nodes that are upstream of the requesting node with respect to the direction that the packet will travel. The nodes count the requests they see. According to the protocol, a node must allow enough *unused* frames to pass through to satisfy all the requests it has seen *before* a packet came to the front of its queue. To an approximation, this causes the network to emulate a distributed FCFS queue. For example, if packets arrive at Nodes 2 and 3 before a packet arrives at Node 1 (where Node 1 is further upstream), Node 1 must allow two empty frames to pass by before sending its packet so that Nodes 2 and 3 can send their packets first.

DQBR, the *HORNET* fairness control protocol, is adapted from IEEE 802.6 to accommodate *HORNET*'s WDM ring by allowing one request *for each wavelength* in each control channel frame, and by maintaining request counters *for each wavelength*. It is implemented using the *HORNET* control channel. The control channel frame carries two bit streams, each of length W , where W is the number of wavelengths. The first bit stream indicates wavelength availability information (as explained in Section 2.3.2) and the second indicates *requests* (note that if $2W$ bits are used for the requests,

four levels of priority can be requested, but that extension is not covered in this work). When a node receives a packet in its transmitter's VOQ, it sets the bit in the *request bit stream* corresponding to the wavelength the packet will use for transmission. All nodes clear the request bit(s) from the control channel corresponding to the wavelength(s) that they receive.

The original version of IEEE 802.6 contains a well-known unfairness problem, which is thoroughly described in [35, 36]. When an upstream node is saturating the transmission bandwidth of the multiple-access channel, a downstream node can be nearly locked out of the network. This issue occurs because when the downstream node places a request, there is significant propagation time for the request to get to the upstream node and then for the available slot to reach the downstream node. During the time between the two events, the upstream node can fill all available slots with packets. Only after the downstream node transmits its packet can it send another request, because the request is sent when a packet reaches the *front* of the transmission queue. The result is that the upstream node is allowed to use almost the entirety of the channel bandwidth.

A correction was developed for this unfairness problem named *bandwidth balancing* [35, 36]. Bandwidth balancing allocates transmission bandwidth evenly among the nodes. However, this solution is contrary to the definition of fairness presented in Section 2.4.1. Therefore, a different solution was developed for *DQBR*. With DQBR fairness control, the node places the upstream requests as soon as a packet arrives to the *back* of the transmitter's queue. The result is that upstream nodes are made aware of the downstream nodes' need for bandwidth and as a result they allow the nodes the opportunity to transmit. The fairness result is proven later in Section 3.5.

Note that the wavelength availability information corresponds to packets going downstream, while the requests should travel upstream (in the other of the two fiber cables) with respect to the direction of propagation of the corresponding packets.

Thus, if a node has a packet to transmit in the counter-clockwise direction, the node places a request on the control channel propagating in the clockwise direction. When it inserts the corresponding packet it marks the wavelength availability information on the control channel propagating in the counter-clockwise direction.

The request counting system, which is diagrammed in Figure 2.9, works as follows. A node maintains a *request counter* (RC) for each wavelength. Every time the node sees a request bit on the control channel for any particular wavelength, it increments the RC for the corresponding wavelength, as shown in Figure 2.9 (a) and (b). Whenever the transmitter's VOQ in a node receives a packet that desires to use a particular wavelength, the value in the RC for that wavelength is transferred to a *wait counter* (WC), which is *stamped* onto the arriving packet, as shown in Figure 2.9 (c). The RC is then cleared. After the packet makes its way to the front of the VOQ, the node decrements its WC value for each available frame it sees on the desired wavelength, as illustrated by Figure 2.9 (c) and (d) ('available frame' refers to a '0' bit in the downstream control channel wavelength availability bit stream). Only when the WC value has been decremented to zero can the packet be transmitted. If the WC value equals zero, or if there is not a packet in the front of the queue, the RC value is decremented each time an available time frame passes by on the corresponding wavelength.

The DQBR request-counting system attempts to ensure that if two packets arrive at two different nodes and desire the same wavelength, the one that arrived first will be transmitted first, as if the network is one large distributed FCFS queue. According to the definition of fairness presented in Section 2.4.1, the distributed FCFS operation is in fact fair to all users in the network. The concept of the distributed FCFS queue implies that all users are accessing the network through the same *location*, thus eliminating the idea of *positional priority*. The ability of the protocol to guarantee equal opportunity for all users of any location is investigated later in Section 3.5.

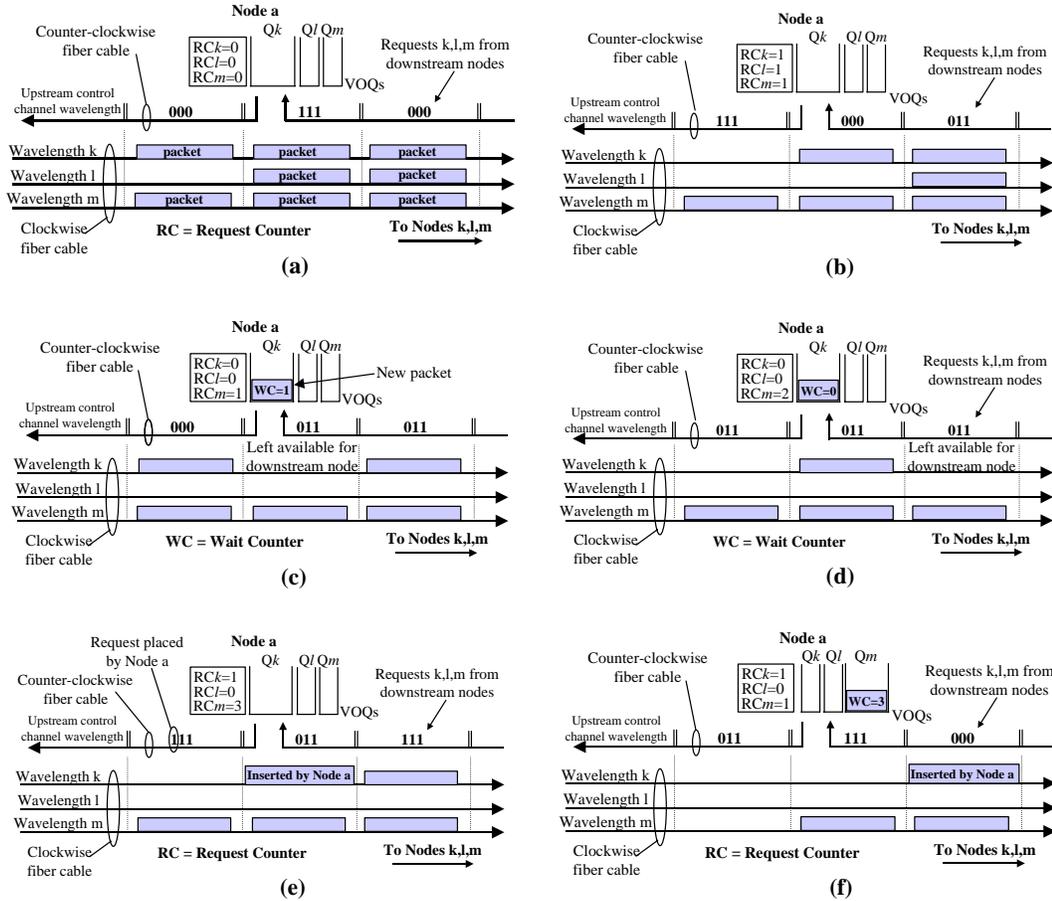


Figure 2.9: DQBR operation: (a) A node monitors the requests on the upstream control channel coming from the downstream nodes. (b) The node increments the RC counters for any requests it sees. (c) When a packet arrives in a VOQ, the value in the corresponding RC counter is stamped onto the packet as the WC. The packet cannot be transmitted during the availability on wavelength k because the WC value is nonzero. (d) The WC counter is decremented for every availability that passes by on the corresponding wavelength. (e) The packet can now be transmitted. (f) When a packet arrives at VOQ m , the value from RC_m is moved into the WC and stamped onto the packet. The packet will have to allow three empty frames on Wavelength m to pass before it can be transmitted.

2.5 *HORNET* Survivability

2.5.1 Conventional Survivable Architectures

SONET ring networks are famous for their ability to provide survivability through the use of protection. When connections to a SONET network are provisioned, they are assigned a particular TDM time slot within the network. Provisioning these TDM circuits is very complex and thus requires a great deal of planning. Conventional networks do not possess the ability to plan these circuits automatically. Thus, if a fiber cut occurs, the network cannot optimally reconfigure its circuits to best utilize the bandwidth available after the cut. Therefore, protection is designed and built into all SONET networks. Protection implies that for every active circuit, the necessary amount of bandwidth and equipment to back up that circuit is strictly reserved for that purpose.

The most basic architecture used for protection in SONET ring networks is the 2-fiber unidirectional path-switched ring (2FUPSR) [37]. The architecture is illustrated in Figure 2.10. The working traffic (the primary traffic streams used under normal conditions) is all transmitted in the same fiber ring in the same direction. The other fiber ring carries the protection traffic (the secondary traffic streams used when a cut or equipment failure occurs) in the opposite direction of the working traffic. Every node has one protection transmitter for every working transmitter, and one protection receiver for every working receiver. The nodes use the protection transmitter to send the exact same data into the protection ring that the working transmitter sends into the working ring.

As a result, the protection receivers in each node receive identical data streams as the working receivers. The SONET receivers are designed to monitor the performance of all the circuits received in the working receiver. If a receiver determines that the performance of any of the circuits is insufficient, then the node ignores the circuit

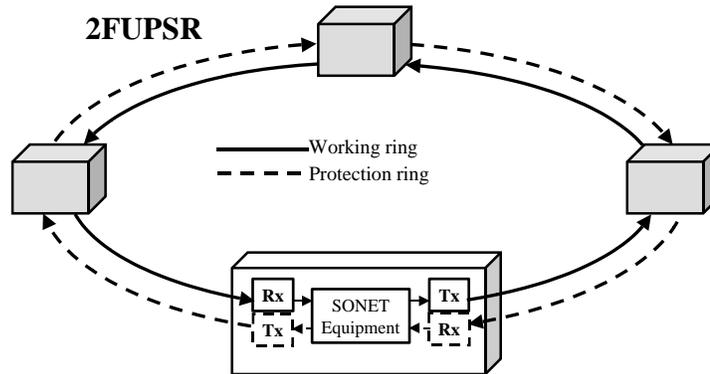


Figure 2.10: The architecture of a 2-fiber unidirectional path-switched ring network. This architecture is commonly deployed in metropolitan area SONET networks.

in the working receiver and uses the copy of the circuit coming into the protection receiver. If a fiber cut occurs, some of the circuits will no longer be received by a node in its working receiver. The node determines which circuits have been cut off, and then switches over to the protection receiver for those circuits. This architecture also protects against equipment failures, such as a transmitter, receiver, or amplifier failure. Note that the protection switch is not protected though.

The drawback of the 2FUPSR architecture is that the node contains $2N$ transmitters and $2N$ receivers, but uses only N at a time. Also, there are two fiber cables available to carry traffic, yet the architecture only delivers the capacity of one fiber cable. Thus, the rigid style of protecting the circuits results in an architecture that can only utilize one half of the available resources.

An alternative to the 2FUPSR is the 4-fiber bi-directional line-switched ring (4FBLSR) [37]. This architecture is illustrated in Figure 2.11. In the 4FBLSR design, two of the fibers are used for working traffic and two are used for protection traffic. The two fibers used for working traffic form a bi-directional ring network. The other two fiber rings are used for protection. If a fiber cut occurs, the two nodes that surround the cut are responsible for performing a line switch. In a line switch, traffic

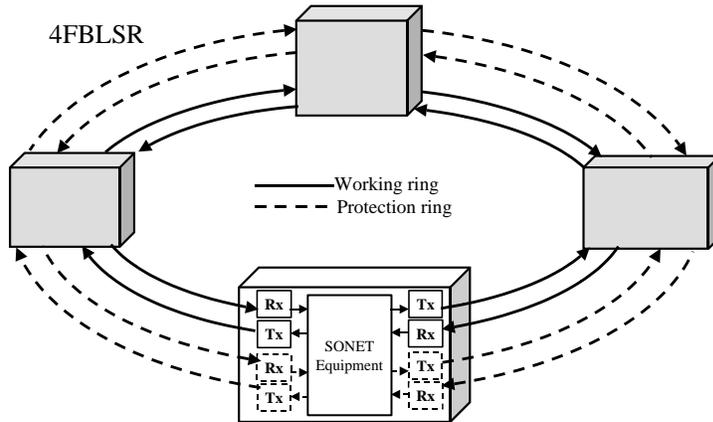


Figure 2.11: The architecture of a 4-fiber bi-directional line-switching ring network.

heading towards the fiber cut is routed into the protection fiber that will carry the traffic away from the cut. This concept is illustrated in Figure 2.12. The line switch can be performed electronically (within the SONET equipment) or optically (by an optical switch). If wavelength routing is used, which is likely to be the case, then the switch must be performed optically, because not all wavelengths will be passing through the nodes' SONET equipment. Figure 2.12 shows the case where the switch is performed optically.

The 4FBLSR architecture also protects against equipment failures [37]. If a transmitter, amplifier, or receiver fails in a link between two nodes, the nodes will switch over to the backup link that is formed with the protection fiber that is carrying traffic in the same direction as the working traffic link. This type of protection switching, called a span switch, is shown in Figure 2.13.

Just as in the 2FUPSR architecture, the 4FBLSR architecture only utilizes one half of the equipment that is deployed. Of the four fiber rings, only two are utilized at a time. Only half of the transmitters and receivers within each node are used. This is a result of the rigid TDM architecture of SONET. All TDM time slots are protected, regardless of whether they are idle, carrying best effort traffic, or even un-provisioned.

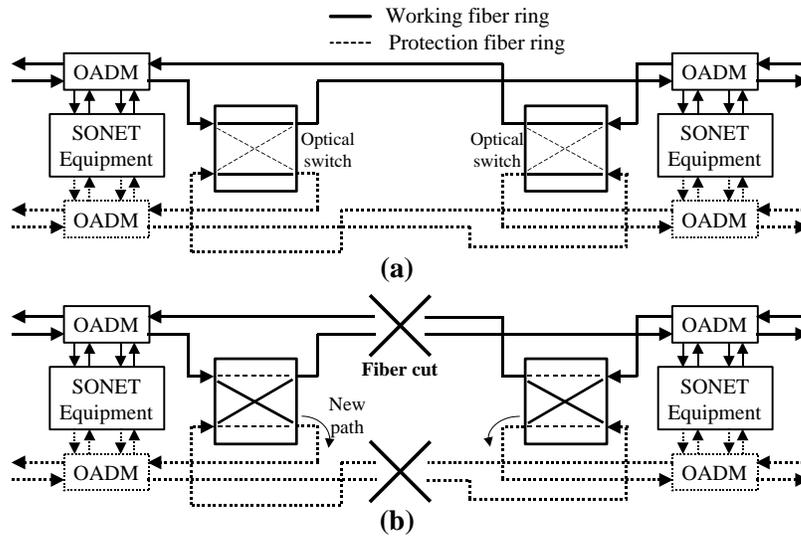


Figure 2.12: (a) Under normal operating conditions, the protection fibers and equipment are unused. (b) When a fiber cut occurs, the two optical switches surrounding the cut are activated in order to switch the traffic away from the cut.

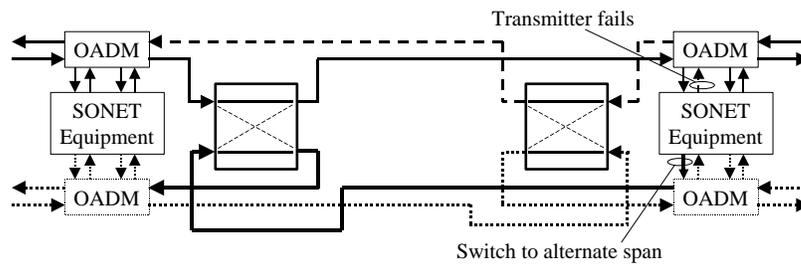


Figure 2.13: The protection equipment can also be activated using a span switch to restore an interrupted connection.

2.5.2 *HORNET* Survivable Architecture: 2FBPSR

Modified versions of both the 2FUPSR and the 4FBLSR architectures could be applied to *HORNET* to make it survivable. In the 2FUPSR implementation, each node would send all of the packets in both directions. The receiving node receives both copies of the packets. Each of the two receivers inspects the source address of the packet and decides whether to keep it or ignore it, based on which receiver is intended to be the working receiver for the particular packet. If for some reason the working path between two nodes is interrupted, then the protection receiver in the destination node is instructed of its new role, and it no longer ignores the packets from the particular source.

The *HORNET* implementation of the 4FBLSR would look similar to the SONET 4FBLSR network. Two of the fiber rings would be used to create a bi-directional ring network. Each node would transmit packets in the optimal direction to the destination. If a fiber cut occurs, the two nodes surrounding the cut would activate optical switches that would switch the traffic that is headed towards the cut into the protection ring, which will send the traffic around the ring in the opposite direction. Span switching is also possible in this architecture.

Just as with SONET, the 2FUPSR and the 4FBLSR architectures require the *HORNET* network only to use one half of the deployed equipment (transmitters, receivers, fibers, amplifiers, etc.) at one time. The other half remains idle. This protection mechanism is necessary for SONET because of the rigid TDM architecture. However, *HORNET* is based on an entirely different premise. *HORNET* is far more flexible than a conventional SONET link, and thus a more efficient survivability mechanism can be implemented.

To see how it is possible, consider a 2-fiber bi-directional *HORNET* ring network and a 2FUPSR SONET network. The two networks look similar, as each has two fiber rings carrying traffic in opposing directions, and each node has at least one

transmitter and receiver for each of the two rings. Under normal circumstances (no cuts, failures) a node on the 2FUPSR SONET ring will send all of its working traffic in one direction around the ring. It would certainly be advantageous if the node could utilize both of the directions of transmission (and thus all of its equipment) for working traffic under normal circumstances. However, consider what would happen if it did utilize both directions. When a cut occurs, at least some of the connections for at least one of the transmission directions will be cut off. There are two paths between all sources and destinations, so the connections that were cut off can be resurrected by using the other direction of transmission. However, performing this step would require a SONET network to quickly reprovision all of its TDM circuits because both directions were carrying working traffic. This is currently not possible for a SONET node to do.

In contrast, *HORNET* does not have rigid circuits that must be reprovisioned in order to change paths between sources and destinations. This advantage over SONET is the basis for the *HORNET* survivability mechanism. The *HORNET* survivable architecture is a 2-fiber bi-directional path-switched ring (2FBPSR). In the *HORNET* 2FBPSR network, *all of a node's transmission capacity* is used for working traffic. None is reserved for protection. In the *HORNET* bi-directional architecture, two paths exist between any two nodes. Under normal conditions, when an access node has a packet to send, it chooses the transmitter that will send the packet along the better of the two paths, as determined by a simple routing algorithm. When a cut occurs, only one of the paths remains to each destination, and thus the node is forced to use that path. The path switch occurs logically inside the node's control and routing electronics. This ensures fast, reliable path switching in the event of a cut. This concept is illustrated in Figure 2.14.

The control channel is used for the detection of the cut and for broadcasting the necessary information about the cut. When a cut occurs between two nodes, both

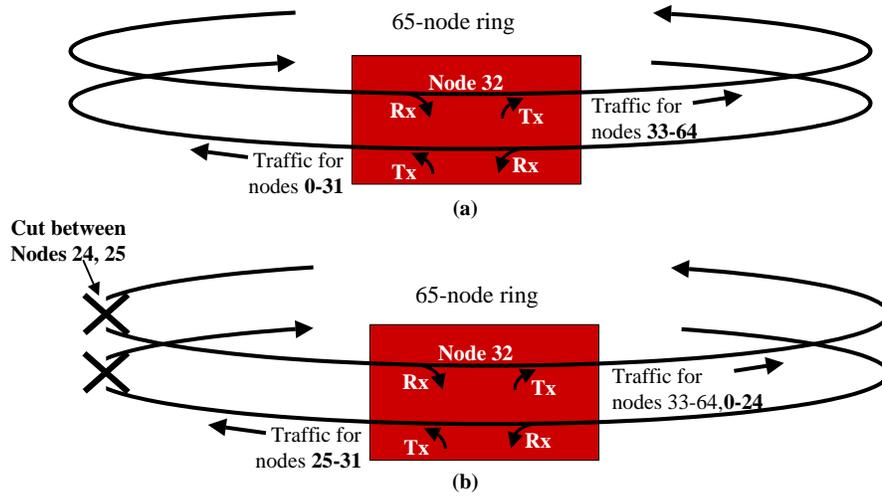


Figure 2.14: (a) Under normal operating conditions, a node attempts to load-balance its traffic while using all available bandwidth in both directions. (b) When Node 32 learned of the cut, it determined that to reach Nodes 0 through 24 it must use the counter-clockwise ring.

of those nodes realize that they are no longer receiving optical power on the control channel or in the payload receiver(s). After the two nodes determine that a cut occurred, they each insert a message into the first control channel frame possible. The message contains two bytes. The first indicates the node number originating the message and the second indicates that a cut has occurred. The message passes through all other nodes on the ring. When a node reads the message, it determines the location of the cut based on the node address in the message. The node then uses what it knows about the topology of the network to determine for which nodes it now must use a different path, as illustrated in Figure 2.14.

In selecting a different path for some of the destinations, the node is performing a *switch logically*, similar to the physical switching that occurs in the SONET architectures. The logical switch takes place within the forwarding engine of the packet router within the *HORNET node*. The packet router's forwarding engine inspects

the IP address of the packet and determines the destination node, similar to any typical IP router. Then, a second stage of the forwarding engine determines which path (clockwise or counter-clockwise) the packet should take to get to the destination. A table is maintained in the forwarding engine to indicate which direction packets should use for certain destination nodes. This table is updated based on network conditions, load balancing, and of course, fiber cuts. When a node learns of a cut, it determines which entries in the table must be toggled (i.e. which paths must change), and then toggles the values in the table. The amount of entries will be small, because the length of the table is equal to the number of nodes on the network. Also, the determination of which entries to modify is a simple operation, especially if the nodes are numbered sensibly. When a cut occurs in a conventional SONET network, the transmission capacity of each node is unaffected because all links are fully protected. In *HORNET's* architecture, the effect the cut has on transmission capacity of a particular node is location-dependent. For nodes far away from the cut, the transmission capacity is generally unaffected. However, nodes closer to the cut are more affected. The extreme case is the node adjacent to the cut, which, as a result of the cut, only has the use of one of the two fiber rings for all of its transmitted data. This in general reduces its available capacity by one half, bringing it down to the same capacity as a node in a conventional network (neglecting *HORNET's* inherent advantage of being able to dynamically adapt to traffic variations). This implies that the *HORNET* architecture can guarantee to its users the maximum capacity of a conventional network, while providing up to 100% more transmission capacity for best-effort traffic, which of course is the most common traffic on the Internet today.

2.6 *HORNET* Control Channel

Recall from the previous sections that the control channel in *HORNET* performs many valuable tasks. It is used to convey wavelength availability information, to relay requests for the DQBR fairness protocol, and to broadcast messages, such as in the case of a fiber cut. There are two important design parameters for the control channel that can significantly impact both the performance and the cost of the network. First, based on the wide range of possible IP packet sizes, an optimal control channel frame size must exist that maximizes the performance of the SAR-OD protocol. Second, the photonic transmitters used for the control channel wavelength must be inexpensive to keep the cost of the node down. Therefore, the type of laser and the wavelength used for the control channel are very important. Both of these control channel design parameters are discussed in this section.

2.6.1 Control Channel Frame Length

As was described in Section 2.3.2, the SAR-OD MAC protocol requires all packets to be inserted to coincide with the beginning of the control channel frame. If a packet that is being transmitted on a particular wavelength is completed somewhere in the middle of the control frame, then the rest of the control channel frame duration on that wavelength must go unused. Another packet cannot be transmitted on that wavelength until the beginning of the next control channel frame. The unused time period on the wavelength is considered overhead and detracts from the performance of the network. Obviously, it is desirable to minimize this overhead.

The minimization occurs at the optimal match between control channel frame length and the distribution of IP packet sizes. The only parameter of these two that the network operator can control is of course the control channel frame size. The control channel has a minimum size that is determined by its functions, but beyond

			W/8 Bytes	W/8 Bytes	4 Bytes	1 Byte	
SOF Indicator	Idle	DQBR requests	Wavelength availability information	Control message	SOF Indicator		

Figure 2.15: Information contained within each control channel frame.

that size it can be extended to any length the network operator desires. A calculation can be made that determines the expected overhead resulting from a certain control channel frame length and an IP packet size distribution. The first step in the process is to determine the minimum control channel frame length.

Based on the functions of the control channel, each control channel frame should contain the following: an SOF indicator byte, a few bytes for carrying messages (four bytes is reasonable), $\frac{W}{8}$ bytes for carrying the wavelength availability information, and $\frac{W}{8}$ bytes for carrying the DQBR requests (W is the number of wavelengths). In general, the value of W should be determined while considering how large the network might eventually scale. If W is 128, then the minimum control channel frame length is 37 bytes (1 for the SOF indicator, 4 for messaging, 16 for the wavelength availability information, and 16 for the DQBR requests). This is summarized in Figure 2.15. The *idle* field can be as long as necessary to optimize the control channel frame length with the distribution of packet sizes.

A distribution of IP packet sizes taken from a particular link [32] was shown in Figure 2.5. As can be learned from the data presented in [32], the exact distribution varies depending on where and when the measurement is taken. An estimated packet size distribution that is based on all of the data measurements in [32] is shown in Figure 2.16. A few popular packet sizes are included in this distribution that did not appear in the data of Figure 2.5. This is the distribution currently expected for the *HORNET* network. However, as new Web applications or transport protocols appear and become popular, the distribution may evolve. Nonetheless, the distribution shown in Figure 2.16 will be used to optimize the control channel frame length in this work.

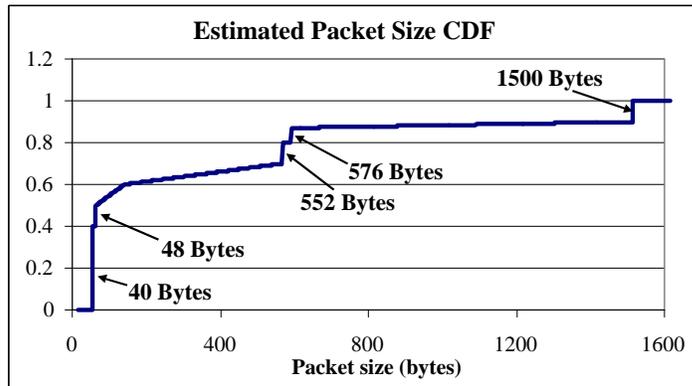


Figure 2.16: This estimated CDF is based on the collection of data for various links as measured by NLANR.

From nearly any measurement, it is apparent that the most dominant feature of the IP packet size distribution is the large percentage of packets at the smallest possible size. For typical measured distributions, at least 50% of the packets are smaller than 48 bytes, with the majority of those packets at 40 bytes. Intuitively, the control channel frame size should be small, because of the fact that if the frame size is longer than a packet, the unused portion of the frame is overhead. The other popular packet sizes, such as 552 bytes, 576 bytes, and 1500 bytes also affect the optimal control channel size, but none have the pronounced impact of the 40-byte packets.

The packet size distributions shown in Figures 2.5 and 2.16 include the payload data and the TCP/IP header. In addition, the *HORNET* transmitter adds another header onto the front of the packet and a trailer to the rear of the packet. The packet size distribution used to determine the optimal control channel frame length must include the payload data, the TCP/IP header, and the *HORNET* header and trailer. The trailer indicates whether the packet is complete or segmented, as discussed in Section 2.3.2. The header has several purposes. It includes a small amount of guard time for packet misalignment and a small amount of guard time for laser tuning (guard

time means that no optical power is transmitted). Additionally, the *HORNET* header is used to provide a bit-sequence that will aid the asynchronous packet receiver in its bit-synchronization process.

Several *information fields* are necessary in the *HORNET* header as well. Most often packets are transmitted to carry payload, but occasionally a packet is sent from one node to another for control purposes. Thus, the header must contain one byte to indicate the function of the packet. The *HORNET* packet header also includes the address of the source node and of the destination node. Typically, networks use the 48-bit hardware address, but twelve bytes is a lot of overhead for short packets, which dominate IP traffic. Thus, in *HORNET* the hardware address is translated into a node number. Since the number of nodes is generally less than 128, only one byte is necessary for each translated address. The final necessary component in the *HORNET* header is a bit sequence for a cyclic redundancy check (CRC). The CRC bit sequence is a common way to check for errors within the payload of the packet. Typically, four bytes are used for the CRC.

It is anticipated that a commercial implementation of *HORNET* would use a *HORNET* header of 20 bytes (1 byte for the trailer, 2 bytes of guard band, 6 bytes for laser tuning, 4 bytes for bit-synchronization, 2 bytes for addresses, 1 byte for a control packet identifier, and 4 bytes for CRC). As technology improves though, the laser tuning time and/or bit-synchronization time will be reduced such that the header only requires 16 bytes. Throughout most of this work, the *HORNET* header is assumed to be 16 bytes in duration.

A calculation can be made to determine the *expected* overhead for a particular frame size, given a packet size distribution. The overhead is defined as the percentage of the transmission that does not contain payload, and is calculated as:

$$\frac{Overhead(bytes)}{Payload(bytes) + Overhead(bytes)} \quad (2.1)$$

where the payload in this analysis includes the TCP/IP header and the data within the packet. The overhead bytes include the *HORNET* header and any *unused* bytes after the packet (before the next control frame). The expected amount of overhead bytes for a particular frame size can be calculated from this equation:

$$EX[OHbytes] = \sum_{i=0}^{1500} PDF(i) \{Header + [\frac{i}{CtrlCh} - \lfloor \frac{i}{CtrlCh} \rfloor] CtrlCh\} \quad (2.2)$$

where i is the packet size in bytes, Header is the number of bytes in the *HORNET* header, PDF(i) is the packet size probability density function evaluated at i , CtrlCh is the control channel frame length in bytes, and $\lfloor \dots \rfloor$ is the *floor* operator.

The results of the calculation for varying frame sizes is shown in Figure 2.17. As expected, the overhead is the lowest at smaller frame sizes. However, the frame size should not be less than 37 bytes because of the amount of information that must be transmitted on the control channel in each frame. Also, for simplicity of the digital implementation, the frame size should be a multiple of four. Thus, although the minimum value of overhead occurs at 57 bytes (neglecting the values for less than 37 bytes), this is not a practical size. The best practical candidates are 60 bytes and 64 bytes, both of which result in an expected overhead of approximately 9%. This subject will be analyzed more thoroughly in Section 3.3. Note that this overhead calculation is the minimum possible overhead because packet segmentation is not taken into account. Segmentation causes additional overhead because the *HORNET* header is applied to a packet multiple times when the packet is segmented.

2.6.2 Control Channel Transmission

As described earlier, the control channel is WDM multiplexed with all of the payload traffic. Thus, a specific wavelength must be reserved to carry only the control channel. Intuitively, it appears necessary to use a typical WDM photonic transmitter that emits

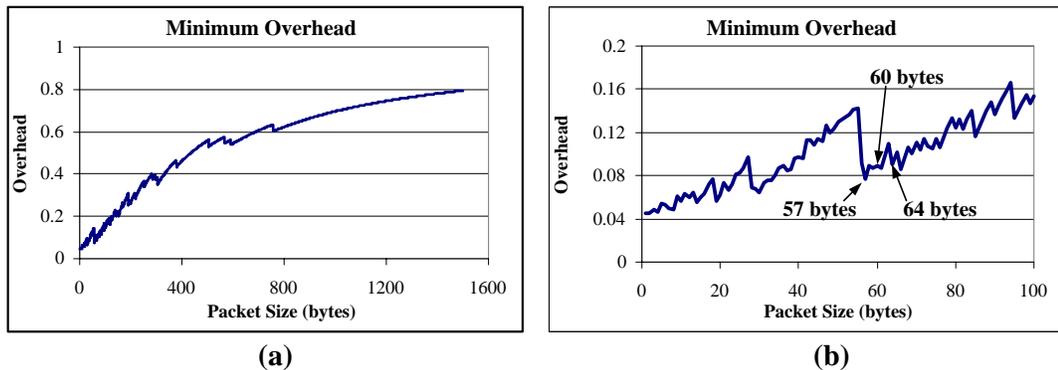


Figure 2.17: (a) The minimum possible overhead for a *HORNET* network with a packet size CDF shown above. (b) Figure (a) zoomed in to focus on lengths less than 100 bytes.

on a specific wavelength within the conventional WDM band. Unfortunately, WDM transmitters are relatively expensive components, and as a result would noticeably increase the node cost. If an *inexpensive photonic transmitter* can be used that still allows the control channel data to be transmitted on a unique wavelength that can be optically dropped and added in every node, it would result in significant cost savings.

Although all of the payload wavelengths in the *HORNET* network are located within the conventional transmission band (centered around 1550 nm), it is possible to use a control channel wavelength that is away from this band. In fact, it is well known that typical fiber optic cable has low attenuation in the wavelength region of 1310 nm. Thus, a laser that emits in the 1310 nm region can be used for the control channel photonic transmitter. Such lasers exist in the form of 1310 nm Fabry-Perot (FP) lasers, which are often used in non-WDM or coarse WDM transmission systems. Transceivers using this laser are commonly sold for less than 10% of the cost of a WDM transmitter. The laser does not emit on only a single wavelength, as WDM transmitters do, but because the payload wavelengths are so far away (in the wavelength domain), this is not an issue. One issue of concern is the difference

in group velocity between 1310 nm light and 1550 nm light. This dispersion causes the control channel and the payload wavelengths to lose synchronization. This issue is addressed in Section 2.7.1.

Although the 1310 nm control channel wavelength solution appears to be a viable one, some may prefer a solution with the control channel in the conventional 1550 nm band. As it turns out, it is possible to use an FP laser to emit on a single wavelength in the 1550 nm wavelength band without increasing its complexity. It is well known that by injecting a single wavelength into the FP laser's cavity, the laser will become a single wavelength laser emitting on the same wavelength as the injected signal (assuming the injected wavelength is within the gain spectrum of the FP laser). It has been shown recently [38] that injection-locking an FP laser with a data-carrying wavelength locks the FP laser to the desired wavelength and allows the FP laser to act as a single-wavelength photonic transmitter.

The design would work as follows. One node on the network uses a typical single-wavelength WDM laser for the control channel photonic transmitter. The node downstream from this node taps some of the optical power from the incoming control channel and injection-locks its FP laser. This causes the FP laser to become a single wavelength laser at the proper wavelength. The downstream node takes this same action, as does the node downstream from it. Thus, only one (or a few) nodes on the network would be required to use an expensive WDM laser for the control channel transmitter. An important factor in the design of this solution is to ensure that the FP laser can be injection-locked by the control wavelength coming from either of the two directions. This is because when a fiber cut occurs, the FP laser may be cut off from its primary injection-locking source. In such a case it needs to use the control channel wavelength coming into the node from the other direction for wavelength locking.

2.7 Control Channel Frame Synchronization

As described in Section 2.3.2, a *HORNET* node is required to begin its packet transmission to coincide with the control channel SOF indicator. Once a packet has been inserted, it is crucial for it to retain nearly perfect alignment with the SOF indicator as it traverses the ring. If a packet becomes misaligned with the SOF indicators, then another packet may collide with it when inserted into the ring. Thus, any misalignment must be absorbed with guard band around the packets, which results in large overhead if the mis-alignment is more than a few bytes.

There are two causes for control channel frame misalignment in the *HORNET* network architecture. First, dispersion within the fiber optic cable causes the different wavelengths of light to travel at different speeds. Thus, the information on the control channel is travelling through the fiber at a different speed than the packets referenced by the control channel information. Secondly, the control channel and the optical packets on the payload wavelengths travel through different paths at each node. Differences in the propagation times through the two paths cause misalignment. Solutions to these two control channel frame misalignment scenarios are discussed in the following subsections.

2.7.1 Dispersion Management for Control Frame Alignment

The SOF indicator on the control channel wavelength marks the start of the control channel frame so that the nodes on the ring know exactly when to begin their packet transmission. It is crucial for the SOF indicator to propagate around the ring at exactly the same speed as the packets that have been transmitted. Otherwise, a packet placed on the network could *drift* across the frame boundaries into a frame that is marked 'empty' by the wavelength availability information. Figure 2.18 shows what happens to the alignment when different wavelengths propagate around the network

at different speeds. This misalignment can cause a node to transmit a packet that collides with the beginning or end of a packet that drifted across a frame boundary.

Frame misalignment occurs in *HORNET* because the group velocity dispersion (GVD) of standard single mode fiber (SMF) optic cable causes optical signals on different wavelengths to travel at different speeds. Therefore, it is necessary to insert dispersion compensating fiber (DCF) optic cable throughout the network to reverse the effect of the GVD of SMF. Optimized lengths of DCF can be concatenated with the transmission fiber at each node. Thorough analysis is necessary to determine the optimal length of the DCF in each node, as well as how effectively the DCF counters the effects of dispersion in SMF.

DCF is typically used in long-haul photonic transmission systems to counter the effect of dispersion on each *individual* optical signal. Dispersion in SMF causes the 'mark' or '1' bit to spread because the pulse contains a finite spectrum. The spectral components of the pulse traverse the link at slightly different speeds, causing the pulse to spread. DCF reverses this effect. The use of DCF in *HORNET* has a much different motivation. In *HORNET*, the DCF is used to maintain perfect alignment between the control channel and the packets on the payload wavelengths. This is a novel use of DCF, as conventional networks are indifferent to the alignment of packets on different wavelengths. However, other future networks, such as optical packet switching networks, may also need to be concerned with optical packet alignment because it is undesirable for packets to arrive misaligned at an all-optical switch fabric.

A mathematical model was generated [39] to calculate the relative time drift of wavelengths in the WDM spectrum and to optimize the design of the dispersion management. The model considers first, second, and third order GVD. Calculations show that two optical signals on wavelengths spaced apart by 50 nm within the 1550 nm transmission window of SMF drift in time relative to each other at a rate of

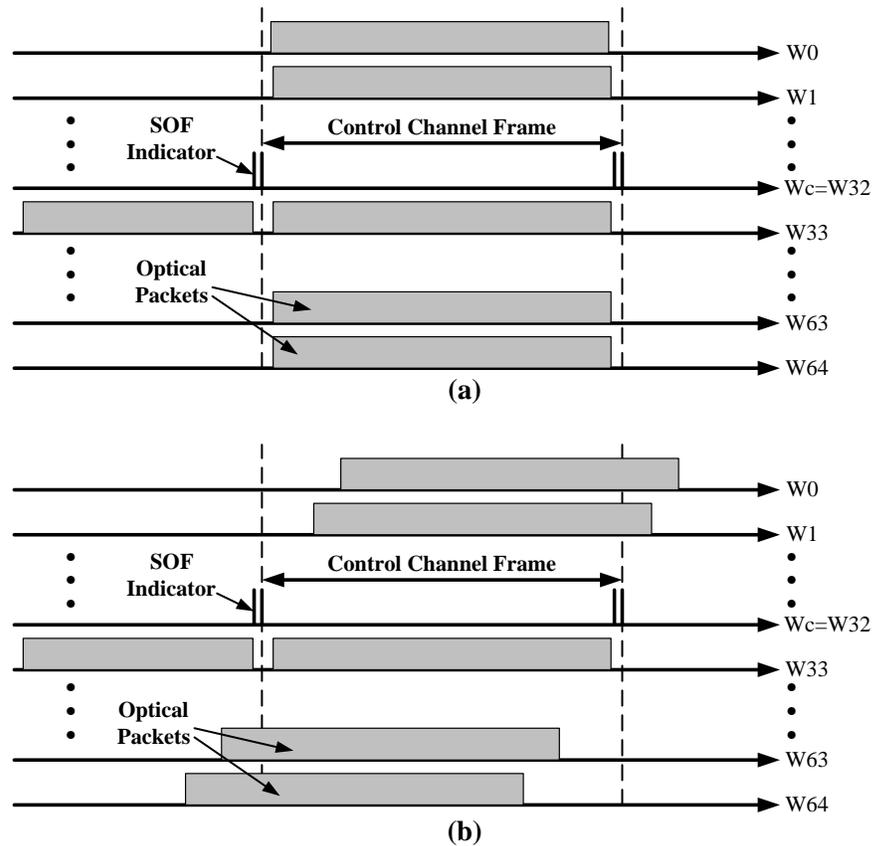


Figure 2.18: (a) Optical packets on a WDM system with 64 payload wavelengths *before* propagating through single mode fiber. (b) Optical packets on a WDM system *after* propagating through single mode fiber. The fiber dispersion causes the packets to drift across control frame boundaries. $W_c =$ control channel wavelength.

1 ns/km. This is unacceptable, considering that the time duration of 1 byte at 10 Gb/s is 0.8 ns, and that nodes in a metro network can be spaced apart by several kilometers. An optimized length of DCF must be used in every node to bring this value down to an acceptable value, preferably less than 0.8 ns for any pair of nodes. Since a ring network may have a circumference on the order of 100 km, this means that the drift should be less than 8 ps/km, which is much lower than the 1 ns/km for SMF mentioned above.

The optimization process for the DCF length is complex because dispersion is nonlinear in SMF and DCF. The DCF does not perfectly reverse the effect of SMF across the entire transmission band. In fact, the length of the DCF can be optimized to perfectly compensate for the time drift in SMF for one wavelength, but generally all other wavelengths will have a nonzero time drift for such a length. Thus, the optimal length of DCF is the length that minimizes the *maximum net time drift* for the wavelengths, where the maximum net time drift is the time misalignment for the *worst* wavelength after propagating through the SMF and optimized DCF.

The results generated using the model were calculated using the dispersion parameters for seven different samples of DCF available in the OCRL. Each sample had slightly different dispersion parameters, as is common for commercial fiber cable. The sample that performs the poorest is able to compensate the time drift to within 10 ps/(km of SMF). This means that after 1 km of SMF and the optimized length of DCF, all wavelengths are aligned with the control channel to within 10 ps. Thus, if the packet on the worst wavelength travelled across 100 km of SMF during its journey around the network, it would only drift 1 ns out of alignment with the control channel. The other samples evaluated compensate the drift far better than this, including one sample that is capable of keeping the maximum drift to below 1 ps/(km of SMF). Clearly, the dispersion-induced misalignment across the WDM payload wavelengths can be compensated using commercially available DCF.

The above analysis assumes that the control channel wavelength is located within the 1550 nm transmission band along with the payload wavelengths. However, as described in Section 2.6.2, it may be desirable to locate the control wavelength at approximately 1310 nm. A slightly different approach to dispersion management is necessary if the control channel wavelength is located at 1310 nm. Since the dispersion of SMF at 1310 nm is nearly zero, DCF is not characterized at 1310 nm and is not designed for operation at 1310 nm. Thus, a network operator should not expect to find DCF to properly compensate the relative misalignment between wavelengths in the 1550 nm band and the 1310 nm band. Instead, the dispersion management design should continue to use the optimization process outlined above to keep the payload wavelengths synchronized to a particular reference point. The control channel is then aligned with a fiber delay after being demultiplexed from the network. Thus, while the payload wavelengths are passing through the DCF, the control wavelength is passing through an optimized length of SMF, just before being received. The length of the SMF does not have to be optimized as well as the DCF because a control channel frame synchronization protocol (described next in Section 2.7.2) can accommodate a few bytes of misalignment. If the target is plus or minus two bytes, then that allows the fiber delay line to have a tolerance of plus or minus 1.6 ns, or 32 cm (for 10 Gb/s data transmission).

2.7.2 Control Channel Frame Synchronization Protocol

The control channel is regenerated by every node on the ring. As the control channel SOF indicator is being regenerated, the packets on the payload wavelengths are passing through a different path, which is all-optical. It is intended that the packets and the SOF indicator will exit the node and re-enter the ring in perfect synchronization, just as it was when they arrived. However, as discussed in the following sub-sections, there are reasons why the SOF indicator would not be inserted at the exact moment

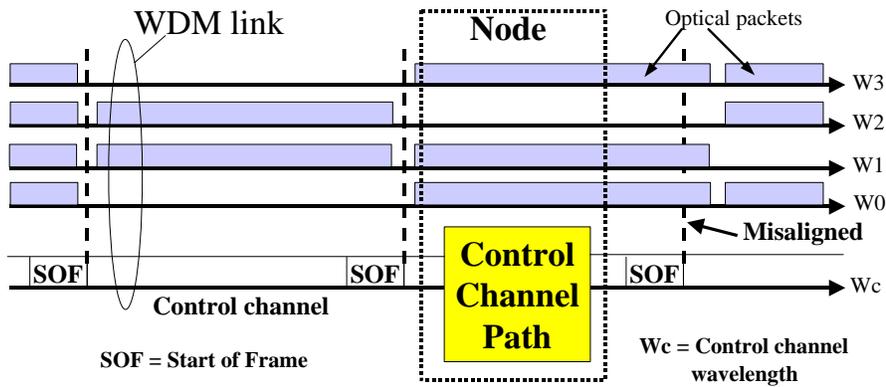


Figure 2.19: The Start-of-Frame Indicators and the packets on the payload wavelengths can become misaligned as they pass through the nodes.

that was intended, and thus it will be misaligned with the packets already on the network. An example of the frame misalignment is shown in Figure 2.19.

In the *HORNET* node, the digital process that is responsible for the processing and retransmission of the control channel is driven by the arrival of the SOF indicator byte (a comparator in the control channel receiver sets a flag high when it detects the SOF indicator byte). The local clock that drives the control and retransmission process is in general not perfectly synchronized with the incoming bit stream. As a result, the control process' clock samples the SOF indicator at a random location along the indicator, and thus the process begins at a random time with respect to the arrival of the SOF indicator. Thus, the elapsed time between the arrival of the SOF indicator and the retransmission of the SOF indicator is random. The elapsed time is uniformly distributed across a 1-byte time duration, and accumulates stochastically as the SOF indicator is retransmitted by each node. In this work, the random deviation from the desired time alignment of the SOF indicator is called *jitter*. The jitter between a control frame and a packet's front edge after one node of propagation and after two nodes of propagation is shown in Figure 2.20. The figure shows that jitter is accumulated stochastically at each node. In the image for Figure 2.20, the digital

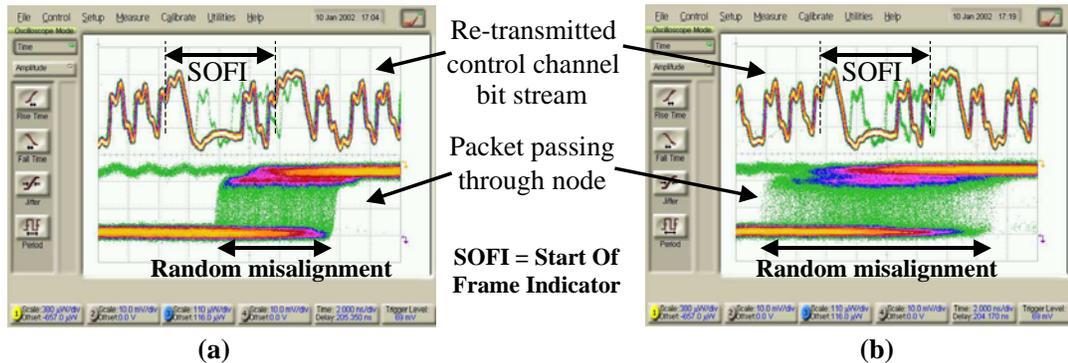


Figure 2.20: Time lapse image on a digital oscilloscope of the random misalignment between the control channel frames and optical packets after the packets have propagated through (a) one node, and (b) two nodes. The instrument is triggered by the detection of the SOF indicator in the receiver.

oscilloscope has sampled the two signals many times using the SOF indicator flag as the sampling trigger. The images of all of the samples are laid on top of each other such that the jitter of the packets with respect to the SOF indicator can be viewed.

Note that if the local clock is at a slightly lower frequency than that of the incoming control channel, there is a finite (though very small) probability that it would miss the arrival of the SOF indicator. It might sample just before the flag goes high and just after the flag goes low. When this happens, the process does not even begin, and an SOF indicator is not retransmitted until the next frame. The faint traces of bits during the SOF indicator in Figure 2.20 are caused by the fact that the receiver occasionally misses the SOF indicator flag and thus the node does not begin its process for the retransmission of the control channel frame.

The jitter that is accumulated in each node adds stochastically as the packet traverses the ring. Since the jitter distribution after one node is uniform, it has a mean $\mu = 0$ and a variance $\sigma^2 = \frac{1}{12}$ of a byte. According to the Central Limit Theorem (CLT), after n nodes of propagation the jitter of a packet can be approximated as a

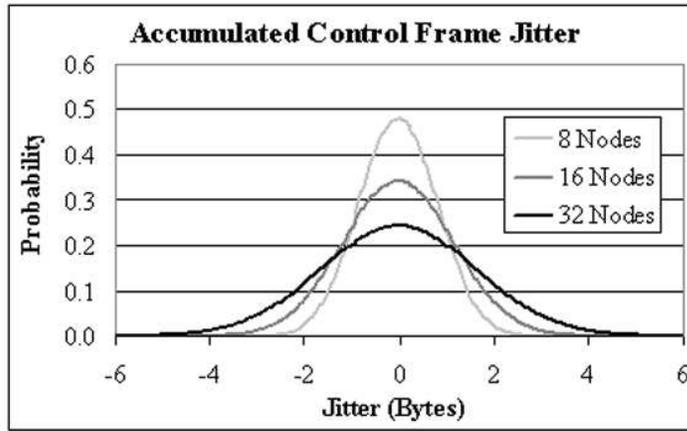


Figure 2.21: Calculated probability density function of accumulated jitter after 8, 16, and 32 nodes of propagation.

normal distribution with $\mu = 0$ and $\sigma^2 = \frac{n}{12}$. The distribution can also be calculated exactly by convolving the original uniform distribution with itself $n-1$ times. Figure 2.21 shows the calculated probability density after a packet propagates through several nodes. According to the calculated cumulative distribution function, *11 bytes* of guard time are required to have a probability of less than 0.001 of the occurrence of a packet that is misaligned beyond the protective guard band after 32 nodes of propagation. This event could lead to a collision.

A natural conclusion is that synchronizing all of the nodes' local clocks can solve the problem. To implement the synchronization, a phase-locked-loop (PLL) recovers a clock from the incoming control channel data stream and then uses the recovered clock for the control process. Doing so guarantees that the SOF indicator will be retransmitted at a deterministic time from the moment that it arrived in the node. However, the solution to the problem is not as simple as this, and thus the use of a PLL is only the first step in the control channel frame synchronization protocol.

For the frame synchronization to work properly, the control channel propagation path must nearly exactly match the through-packet propagation path. These paths

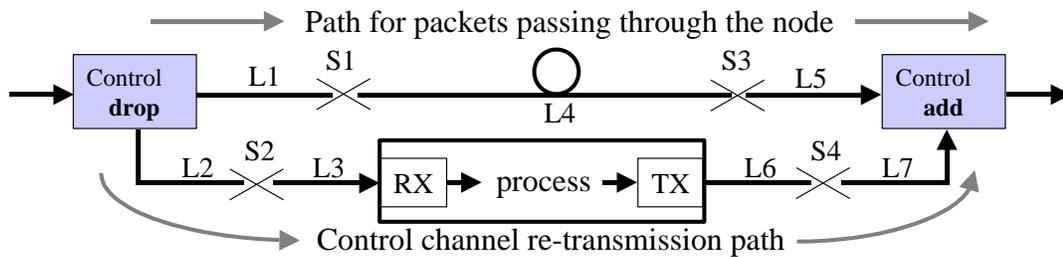


Figure 2.22: The control channel and the payload packets passing through the nodes pass through two different paths. S_n denotes splice locations, L_n denotes fiber lengths.

are shown in Figure 2.22. A fiber delay line in the through-packet propagation path 'buffers' the packets while the control-channel-electronics in the node receive, process, and re-transmit the control channel. Thus, the fiber delay line in the payload wavelengths' path must have the exact propagation delay time as the fiber delay time plus the control channel electronic processing delay time in the control channel path. Because one byte of propagation distance in fiber is approximately 16 cm for 10 Gb/s data, the fiber length must be controlled very tightly. Figure 2.22 shows splice locations and fiber lengths that must be tightly controlled in the manufacturing process. This is not typically done in manufacturing processes for optical networking equipment. Other equally important issues such as microcode upgrades during the design process and after a product release can complicate the situation further. Note that any errors in the design of the equipment (including the microcode processing delay) will be magnified N times, where N is the number of nodes through which the packets may propagate in the network. Errors due to manufacturing issues (mainly fiber cable length errors) will add stochastically at each node.

To avoid the difficulties that can arise with attempting to manufacture a perfect match between propagation paths in the node, a *calibration technique* was developed as part of the frame synchronization protocol. The calibration gives a node the ability

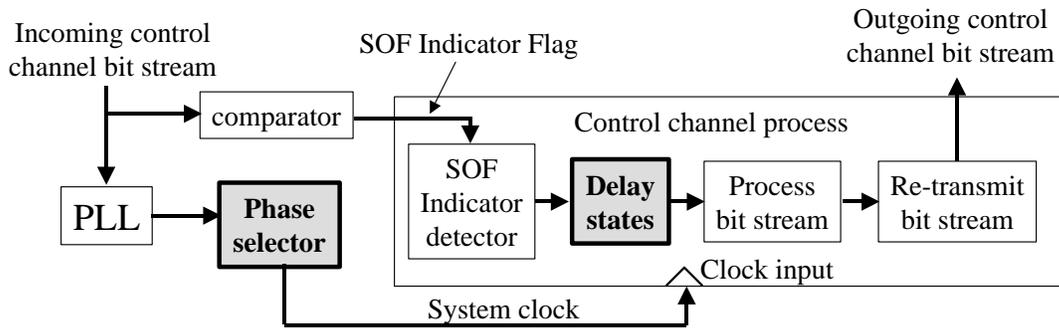


Figure 2.23: The output phase of the PLL and the delay states are controlled by the node to provide perfect control channel frame synchronization.

to *auto-align* the retransmission of the SOF indicator while the node is deployed in the network. A node can use its *microcode* to alter the propagation delay of the control channel path by controlling the phase of the process' clock and by using a variable number of delay states in the digital logic process.

Figure 2.23 shows the important components involved in the calibration. The two highlighted components, the PLL phase selector and the delay states, are used to adjust the propagation delay through the control channel path. Delay states are digital logic process states in the control channel processing algorithm that allow for a programmable processing delay. The number of the delay states can be dynamically increased or decreased to increase or decrease the propagation delay of the control channel path. Since this only gives granularity of 1-byte-durations, something more precise is necessary. For this, the node can control the output phase of the PLL. Generally, PLLs are programmable so that the output phase can be controlled to within $\frac{\pi}{4}$ ($\frac{1}{8}$ of a clock cycle). The output phase of the PLL dictates the moment at which the incoming SOF indicator comparator output flag is sampled. Sampling the SOF comparator output flag near the beginning of its duration will shorten the propagation delay of the control channel path, just as sampling the flag near its end will lengthen the propagation delay.

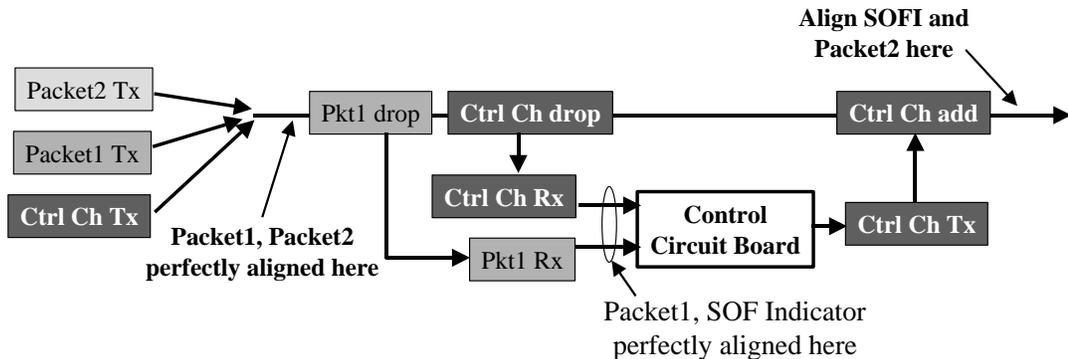


Figure 2.24: The setup for the *lab cal* procedure.

The calibration requires two steps to achieve nearly perfect SOF indicator alignment. The first step is a laboratory calibration (*lab cal*) to put the node in a position to perform its auto alignment when in the system. Essentially, it provides a reference condition for the node. This is a manual step performed by an operator either just before the node is sent out for installment or with remote measurement equipment at the installation point.

To perform the *lab cal* step, the node is placed in a 3-wavelength system. One wavelength is used for the control channel and the other two carry packets that are perfectly aligned in time with each other. The system setup is shown in Figure 2.24. One of the wavelengths carrying packets is the node's drop wavelength. The operator arranges the *lab cal* system such that the SOF indicator flag and the front edge of the dropped packet arrive to the processor at exactly the same instant. The operator then adjusts the node's logical delay states and clock phase until the retransmitted SOF indicator and the through-packet are perfectly aligned at the node output as they are intended to be in the network. This provides a reference state for the node.

Once the node is placed in the network and is turned on, one of the first things it must do is to perform the in-system calibration (IS-cal). The network contains at least one master node, which is notified of the new node on the network. The master

node sends a long stream of short packets to the network's new node. The node measures the time duration between the arrival of the front edge of the packets and the arrival of the SOF indicator. The time is measured to within the phase adjustment granularity of the PLL (most likely $\frac{1}{8}$ of a clock cycle). Since the retransmission of the SOF indicator is currently set (by the *lab cal*) for the condition where the SOF indicator and the packet arrive simultaneously, the node knows that it should adjust the control channel propagation delay by the time difference that it measures between the incoming packet and SOF indicator.

To nearly exactly measure the time difference between the arrival of the SOF indicator and the calibration packet from the *master node*, the node cycles its PLL output clock through all phases, taking several samples between each PLL phase adjustment. As shown in Figure 2.25, the adjustments alter the relationship between the clock phase and the incoming SOF indicators and packets. In some phase settings, the node measures n samples between the SOF indicator arrival and the calibration packet arrival, while in the rest of the phase settings the node measures $n+1$ samples between the two arrivals. It can be shown that the actual time difference between the SOF indicator arrivals and the calibration packet arrivals is the average over all phases of the number of samples between the two.

For example, in Figure 2.25, the node originally has its PLL output phase in the *phase 1* condition. It measures one clock cycle, or one sample between the two arrivals. It then changes to *phase 2*. The result of the measurement is the same as for *phase 1*. After changing to *phase 3*, the node measures zero samples between the arrival of the calibration packet and the SOF indicator. The same is true for *phase 4*. Assuming that these four phase settings complete the cycle of possible PLL output phases, the node can determine that the actual time difference between the arrivals of the SOF indicators and calibration packets is $\frac{1+1+0+0}{4} = \frac{1}{2}$ of a clock cycle.

Once the node has determined the time difference between the arrivals of the SOF

indicators and calibration packets to within the granularity of the phase adjustments, it adjusts the number of delay states and it reprograms its PLL output phase in order to adjust the propagation delay of the control channel path. An experimental demonstration of the protocol is presented later in Section 5.2.2.

2.8 Network Reconfigurability in *HORNET*

2.8.1 Dynamic Traffic in the Metro Area

In the base model for *HORNET*, each node contains a tunable transmitter and a wavelength drop for each of the two directions of transmission. Each node receives only one wavelength, and that wavelength never changes. In such a case, Node 0 receives Wavelength 0, Node 1 receives Wavelength 1, and Node n receives Wavelength n . However, this design is only sensible for a network with static, easily predictable traffic conditions. Naturally, this is not the case for metropolitan area networks. Traffic is and always will be *dynamic* in metro networks. In some cases the dynamics are predictable, such as for the case of diurnal traffic patterns, and in some cases the traffic load fluctuations are random. Regardless of whether the dynamics are predictable or not, it is necessary to design *HORNET* such that it intelligently handles the dynamic traffic inherent to metropolitan networks.

The diurnal traffic patterns of a network provide a clear, simple example for the impact that dynamic traffic has on a network. Consider a node that is geographically located in an area full of corporate establishments. During the daytime hours the node is heavily utilized as the users connected to the node download large quantities of data traffic. During the nighttime hours, the corporate area is practically devoid of users, and only a small amount of network traffic will be flowing into the node. Meanwhile, while the nodes located in corporate areas are quiet, the nodes located in

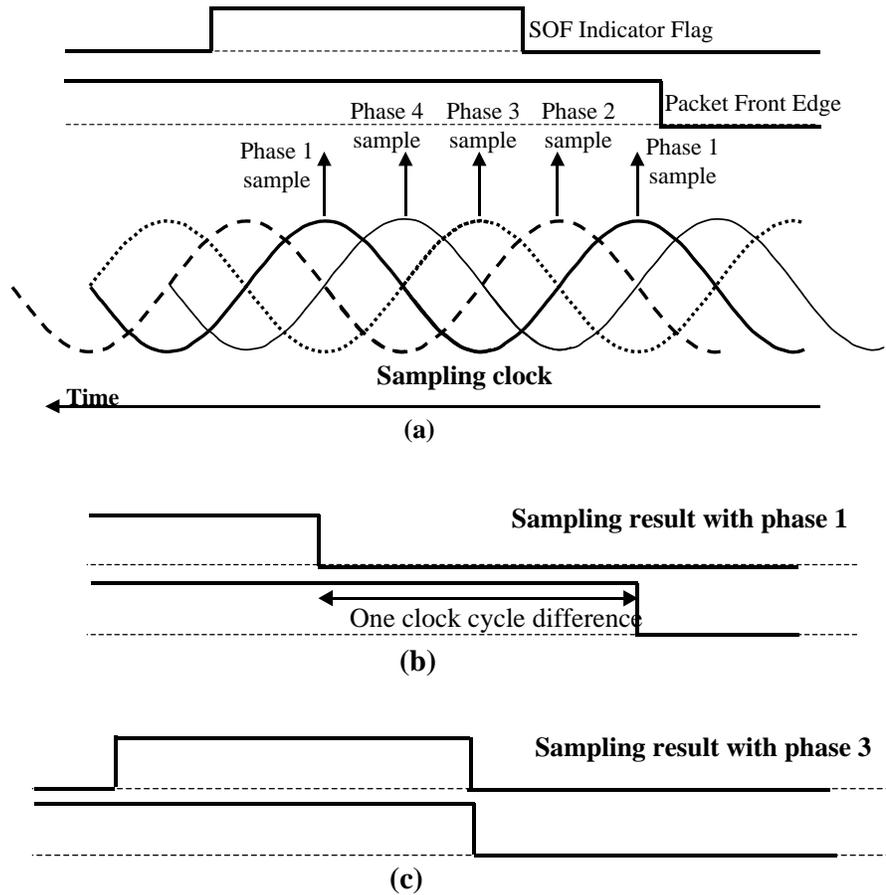


Figure 2.25: (a) During the *IS-cal*, the node cycles its process clock phase through all possible phases. In this example, only four phases are used (0 , $\frac{\pi}{2}$, π , and $\frac{3\pi}{2}$). (b) For the first sampling phase, the node perceives that the calibration packet front edge arrives one clock cycle before the SOF indicator flag. (c) For the third sampling phase, the node perceives that the calibration packet front edge and the SOF indicator flag arrive simultaneously.

residential areas are receiving large quantities of network traffic. The design goal is to ensure that nodes in a corporate area receive enough wavelengths to support the incoming traffic at the peak usage during the daytime, and that nodes in a residential area receive enough wavelengths to support peak usage in the evening and night hours.

The *brute force approach* is to permanently assign each node enough wavelengths such that at peak usage, the node's incoming traffic can be fully supported. However, this solution is wasteful of the available resources and as a result is not a cost-effective design. Consider a network with 20 nodes that has 10 nodes in corporate areas and 10 in residential areas. Imagine that the nodes receive five times as much content during the peak hours than during the off-peak hours. At the peak, each node requires 5 wavelengths to support the quantity of incoming traffic. By the brute force approach, each node is permanently assigned 5 wavelengths (i.e. each node has 5 wavelength drops), resulting in a total of 100 wavelengths. However, *only 60 wavelengths* are being utilized (in the daytime the 10 corporate nodes are using 5 wavelengths each and the 10 residential nodes are using 1 wavelength each).

Unfortunately, optical networks still suffer from bandwidth limitations. The primary limiting factors in *HORNET* are the gain bandwidth of the optical amplifiers and the number of wavelengths supported by the tunable transmitters. As a result, it is necessary to be careful with the use of wavelengths. The brute force approach does not at all do this. The network supports 100 wavelengths, but the bandwidth of only 60 wavelengths is being utilized. If the bandwidth of the amplifiers and the tuning abilities of the transmitters limit the network to only 60 wavelengths, then the network proposed in this example cannot be built. An efficient method for provisioning bandwidth where it is needed must be developed instead.

A more logical solution is to use dynamic wavelength provisioning to place the

bandwidth where it is needed when it is needed. Consider again the 20-node network. If the nodes support dynamic wavelength provisioning, then during the day 5 wavelengths can be provisioned to each of the corporate nodes while only one is provisioned to each of the residential nodes. During the evening hours the situation is reversed. As a result, the network only requires 60 wavelengths, and can thus be constructed, whereas the network described above could not.

The example of the diurnal traffic patterns of metro networks is a clear testimony for the use of dynamic wavelength provisioning in *HORNET*. The virtues can be extended to random, unpredictable fluctuations in network traffic as well. As the network detects that traffic is heavier for certain nodes for a substantial amount of time, an extra wavelength can be provisioned for that node. When a node is not receiving nearly enough traffic to justify the number of wavelengths currently provisioned for it, then the network can reprovision one of the node's wavelengths elsewhere. As a result of the network's ability to place the bandwidth where it is needed, bandwidth is better utilized, and fewer wavelengths are needed.

2.8.2 Necessary Technologies for Dynamic Networks

Two technologies are required to make this proposed solution a reality. The first requirement is the reconfigurable optical add/drop multiplexer (R-OADM). Such a device has the ability to drop a reprogrammable quantity of wavelengths into the node. The second technological requirement is a protocol that automatically provisions the wavelengths to the nodes as necessary. These two technologies are described below.

As expressed in the example in Section 2.8.1, the goal of the R-OADM is to have the ability to drop a reprogrammable number of wavelengths. If it is determined that at peak usage the node must receive M wavelengths, then the R-OADM should have the ability to drop any number of wavelengths m , where m is between 1 and M . Ideally, the m wavelengths dropped by the R-OADM can be any of the W wavelengths

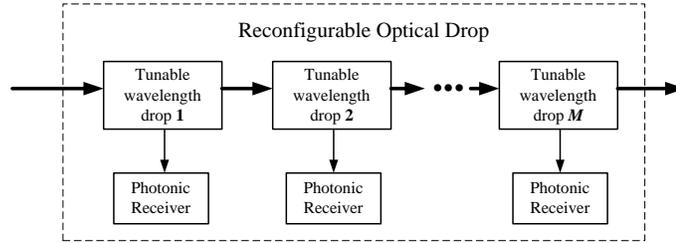


Figure 2.26: Logical schematic of the function of the Reconfigurable Optical Drop needed in *HORNET*. M is the maximum number of wavelengths the node requires.

carried in the network. However, it is possible that available technology would restrict the m wavelengths to a particular subset w of the W wavelengths in the network. The smaller w is, the more restricted the wavelength provisioning becomes. If $w < W$, then there is a nonzero probability that a node would need to provision an additional wavelength and that there would be available wavelengths for provisioning, but that the available wavelengths are not in the set w . Thus, it is desirable to make w as large as possible.

A logical schematic of the R-OADM necessary for *HORNET* is depicted in Figure 2.26. In the logical representation, there are M tunable optical drops. A tunable optical drop can select any (or none) of the W wavelengths in the network and drop it into the node while allowing all other wavelengths to pass through. Though this may ultimately be the actual design, the R-OADM does not necessarily need to be composed of individual tunable drops. Other technologies, such as MEMS or acousto-optics may be used to develop one component that is capable of performing the same function. At the time of the preparation of this report, such R-OADM are not available for commercial purchase. However, researchers have made significant progress, and commercial products are expected in the near future.

Other research projects in recent years have also proposed the use of R-OADM for dynamic wavelength provisioning, such as the work in [16]. However, the R-OADM

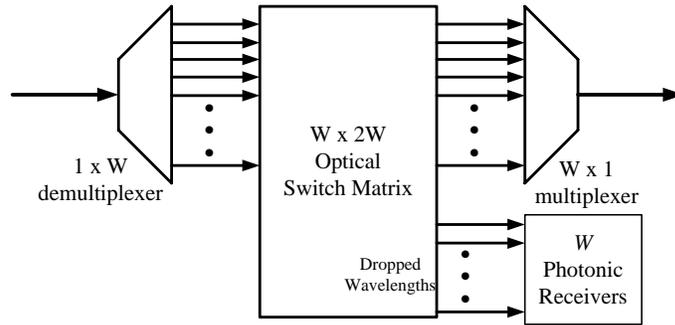


Figure 2.27: Typical R-OADM design proposed in many other research projects. The expensive optical components and the W photonic receivers make the design impractical for a metro network.

design put forth in such projects is not cost-effective. The general design of the R-OADM is shown in Figure 2.27. All wavelengths are demultiplexed in order to give the node access to any of the wavelengths. An optical switch matrix directs each individual wavelength either into the node or back onto the transmission fiber. The design is not cost-effective because it requires the node to have W receivers, where W is the number of wavelengths in the network. Also, the multiplexer/demultiplexer combination is expensive, as is the switch matrix. As mentioned previously, the R-OADM should only contain M receivers, where M is the maximum number of wavelengths the node will require. Generally, M will be *significantly* less than W .

To properly implement a network that uses R-OADMs, the network requires a protocol for provisioning wavelengths for different nodes. The wavelength provisioning protocol is an interesting item for study. The goal of the protocol is to enable the network to automatically provision wavelengths to the nodes that require them, and to notify all nodes in the network of the new network configuration. All nodes will update the forwarding tables in their packet routers when learning of a newly provisioned wavelength. In general, it is not a difficult problem, and thus is not addressed in detail in this report.

The protocol design could take on either of two forms. It could be a centralized protocol in which a master node determines when to provision wavelengths for nodes on the network and which wavelengths to provision for them. The master node can use the information contained on the control channel (i.e. wavelength availability information and DQBR requests) to accurately monitor the demand for any particular node on the network. After determining which wavelength to provision for a particular node, the master node notifies all nodes on the network of the new configuration, and the reprovisioning node reconfigures its R-OADM.

Alternatively, the protocol could take on a distributed form. In one likely implementation of a distributed protocol, a node determines when it needs to receive more wavelengths. It then chooses a wavelength from the set of currently available wavelengths and reconfigures its R-OADM to receive that wavelength. It then uses the control channel to notify all other nodes of the new configuration so that they can update their forwarding tables. Because the protocol is not very complex, and because it does not need to provision new wavelengths very quickly, neither form of the protocol has a distinct advantage over the other. Nonetheless, it is likely that a network operator would prefer the centralized option for easier manageability.

2.9 Quality of Service on *HORNET*

2.9.1 Motivation for Constant Bit Rate

New network architectures such as *HORNET* must support existing services at least as well as their conventional counterparts. One fundamental challenge that stems from *HORNET*'s packet-oriented physical layer is the requirement to support constant bit rate traffic. Although bursty data traffic has long surpassed voice traffic in quantity, the need to support new services such as streaming content delivery, as well

as to support existing voice traffic clearly exists. Unlike SONET networks, which are circuit-based, *HORNET*'s infrastructure must employ special techniques to guarantee constant data rates. Constant bit rate is a subset of the more general characteristic - Quality of Service. When all nodes are allowed access to all wavelengths, as in *HORNET*, each node must self-police in some form to support Quality of Service. In addition, there must be a mechanism by which to inform other nodes of bandwidth requirements - i.e. a reservation protocol.

2.9.2 Circuits over *HORNET* (CoHo)

We have developed protocols to allow fixed bit-rate Circuits over Hornet (CoHo). CoHo allows circuit switched network elements (NEs) such as SONET boxes, point-to-point Ethernet (Gigabit Ethernet for example) switches or Packet over SONET (POS) linecards to communicate over *HORNET*, in a synchronous manner. For a purely packet service, circuits can provide guaranteed bandwidth from customer premises to the carrier POP or CO, rather like current T1 lines.

The applications mentioned above typically require circuits that are high bit-rate (1-100s of Mb/s) and long lived at the same time. In contrast, circuits that can be set up and torn down (reconfigured) quickly may be used to (1) support real-time services efficiently, (2) allow data bursts between nodes, (3) support higher layer circuit based services (TCP Circuit Switching, for example). Typically such circuits would require lower bit-rates (100s kb/s - 1 Mb/s). While there is no immediate application that can make use of circuits that can be setup and torn down rapidly, the ability to do so brings up some very interesting questions in the way networks are currently built and used. Since high bit rate, long-lived circuits can be thought of a subset of rapidly configured (on the order of milliseconds), smaller circuits (100s of kbps), we focused on the latter.

In addition, CoHo has some very nice and unique features, some of which stem

directly from the *HORNET* architecture and packet-based transport: (1) All circuits are independent of each other. Hence service levels, protection mechanisms, etc. can be different for different circuits. (2) CoHo can potentially support reconfigurable circuits with fast setup and teardown times, limited mainly by the transmission delay of signals in an optical fiber. (3) We believe CoHo can provide circuits at a fine granularity (100s of kb/s). (4) Using CoHo, a node can establish circuits on a wavelength without having to ADM (add-drop-multiplex) the wavelength. This is in sharp contrast to SONET or other networks that use circuit-multiplexing methods. This can lead to cheaper interconnected networks. (5) CoHo can be completely distributed with only the source and destination nodes having to maintain state for a circuit.

To allow Circuits over *HORNET*, one of the most important sub-systems and associated protocols is the reservation and scheduling mechanisms. The basic functions of the reservation and scheduling mechanism are as follows: enable nodes to setup fixed bit-rate circuits, schedule transmission over such circuits and teardown the circuits once their usage is over. The following sections cover mechanisms that support these functions and related issues in more detail. Section 2.9.3 will review possible designs for reservation and scheduling schemes as well as suggest modifications to the *HORNET* architecture. Section 2.9.4 describes one such scheme we have examined, discusses network simulations aimed at evaluating two different flavors of reservation schemes in terms of fairness, and provides a hardware design with results obtained from VHDL simulations. Section 2.9.4 goes on to formulate a framework for comparing reservation schemes in context to *HORNET* and tries provide a comparison for the schemes discussed in Section 2.9.3.

2.9.3 Potential Reservation and Scheduling Mechanisms for *HORNET*

The basic function of the reservation mechanism is to reserve a fixed bandwidth circuit on top of *HORNET*'s time slotted physical layer (teardown and scheduling data transmission come later on). One way to provide a fixed pipe is to allow nodes to use slots in a periodic manner. Here we will introduce the first modification to *HORNET*. We will transmit a slot number (referred to as slot # from hereon) on the control channel, in each slot. A possible slot numbering scheme is as follows: there is a master node on the ring (any node can be the master). The master will start generating slots and number them. When Slot 1 loops back to the master node (after 1 trip around the ring = RTT), the master will stop transmitting new slots and re-send slot 1 instead. This creates an 'incomplete' slot on the ring, let's call it slot(N+1). When the incomplete slot, slot(N+1) loops back to the master, the master has two options: it can simply remove it by extending the previous slot (slot N) or mark slot(N+1) as incomplete (for example, a bit in the control channel can be used to indicate the validity of a slot). We will choose the first option above. Hence in this slot numbering scheme, there are N slots in the ring. We would like to stress that there can be other ways of maintain slot numbering. Section 5 discusses one such option. For the purpose of simplicity, let us assume that N will be the number of slots.

As expected, N depends on the length of the ring, the bit-rate carried by the wavelengths and the size of each slot. A 100km ring with a 64 Byte slot-size will contain approximately 8000 slots at 10Gb/s and 2000 slots at 2.5Gb/s. Let us call this one frame. Thus if a node uses one slot every frame, it can achieve a fixed bandwidth pipe, having a bit-rate of 1Mb/s. 1Mb/s is also the granularity of such a system. A reservation mechanism is needed to guarantee transmission in one or more

slot(s) every frame.

To get an intuitive feel as to how one might proceed to design slot reservation mechanism for *HORNET*, we will refer to *HORNET*'s tunable transmitter fixed receiver design. In a time slot, the transmitter can transmit on only one wavelength and hence communicate with only one receiver. Similarly, the receiver can receive only one packet (from some source) in one slot. Thus *HORNET* behaves just like a crossbar or a time multiplexed space switch, the difference being that *HORNET* is distributed over 100km of fiber in a ring topology. Hence a reservation mechanism should find a time slot that is free at both the source node and the destination node.

This is a challenging job, especially if the mechanism is to be handled in a distributed manner. We have explored a number of solutions that can potentially solve this problem. We will discuss these solutions briefly before discussing one of them in greater detail in Section 2.9.4:

1. Reservation mechanism using broadcast of a node's reservation status: Either the source's or the destinations' reservation status can be broadcast in each slot. This can be done quite easily using the control channel; one bit in the control channel is enough. If the bit is = 1, the node has been reserved. Thus in general, there will be a reservation vector in each slot. When a node wants to reserve a slot, it simply sets the corresponding reservation bit to 1. Similarly if a node wants to clear a slot, the correct bit in that slot should be cleared.

Thus in a source-based using broadcast, the receiver's reservation status will be carried in a slot. Source nodes can reserve slots, while slots can be cleared either at the source or at the receiver. In a receiver-based scheme using broadcast, the source's reservation status will be carried in a slot. Source nodes send circuit requests to the receiver. The receiver queues requests from all source nodes. It can then reserve slots and send the slot # information to the source node. Receivers can also take care of clearing the slots. In both cases, the source node needs to be equipped to schedule

transmission of data on reserved slots. By scheduling, we mean the following: when a slot j that has been already reserved arrives at a source node, the node should be able to transmit the correct data, belonging to the circuit using the slot j .

2. Reservation mechanism based on a request-exchange iterative exchange: This scheme is similar to iterative scheduling algorithms used in input queued switches. A typical slot reservation procedure will be as follows: the source node sends a request to the destination, for a time slot. The receiver returns a random time slot that is unreserved for itself, say slot j . If slot j is unreserved at the source node, the slot reservation is over. If not, the source sends a teardown or clear message to the receiver for slot j . The receiver clears slot j at its end. The source sends another request to the receiver and this process continues until either a slot is found or the source node gives up. Once a slot is reserved, the source is responsible for scheduling data transmission.

3. Reservation mechanism using a deterministic (but reconfigurable) time slot allocation: Each source and destination node start-off with a fixed pre-allocation of slots. Let's assume that each source-destination pair gets $1/N$ of the destination bandwidth where N are the number of nodes. Initially, the reservations are deterministic - i.e. the source can choose any pre-allocated slot per destination. It is hence the owner of a quota of slots, per receiver. Once the source exceeds its quota, it can seek permission from other owners to use their time slots. A possible way to handle this is as follows: the source simply sets a bit in the control channel asking for permission, let's call this the permission bit. The owner sees the bit and can set another bit, let's call it the ack bit, giving the source permission to use the slot. If the owner needs a slot it has loaned out back, it again sets a bit, the return bit to inform the source node. The source node then returns the slot, either after its circuit is complete or explicitly tears the circuit down.

It should be noted that the schemes described above do the basic functions of reserving, clearing and scheduling slots. There will be a need for higher layer control

mechanisms to enforce fairness, admission control, etc. This will be discussed in Section 2.9.4, in more detail.

We chose one scheme, the source-based scheme using broadcast, to design in detail and implement using VHDL. The main goal was to get an idea of what is the complexity of such a design, what are the hardware requirements, etc. A recurring theme behind this work was to evaluate how fast can a slot be reserved and cleared and how it affects circuit setup and teardown speed. In addition to evaluating the hardware implementation, we focused on two different reservation protocols within this framework, and performed network simulations to evaluate performance and fairness metrics.

2.9.4 A Source-based Reservation Mechanism Using Broadcast

This section will describe this scheme in great detail. We will start by laying out the basic assumptions. Then a possible procedure for circuit setup and teardown will be discussed with the help of a functional block schematic of a node. We will then discuss the basic functions performed by the slot reservation and scheduling sub-system and the building blocks of such a sub-system. From here, we evaluate two specific reservation protocols using computer simulations. After examining both greedy and non-greedy circuit reservation techniques, we describe the logical flow of an algorithm that performs the reservation functions. Finally, we present simulation results obtained from a VHDL implementation of the above and discuss our findings.

Assumptions

The basic *HORNET* architecture and node design stays the same. The control channel will carry the slot number in each slot. The number of slots in the system is assumed

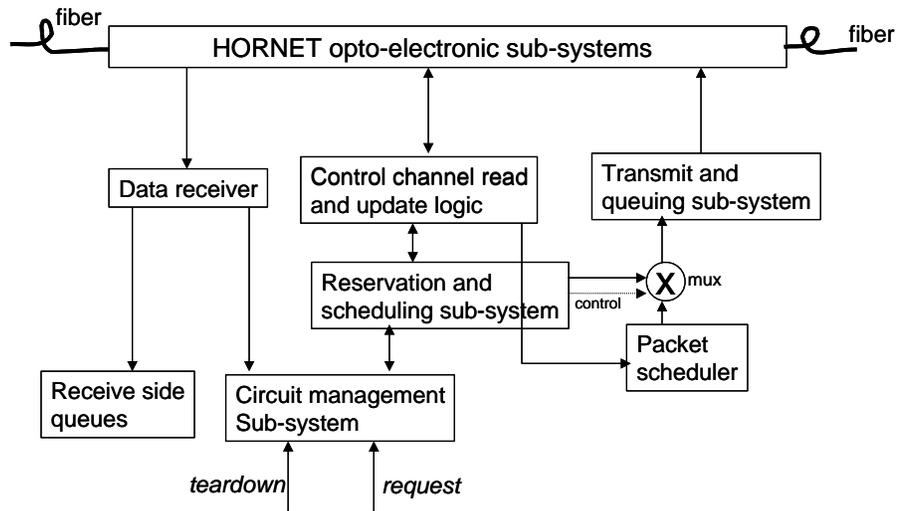


Figure 2.28: Functional block diagram of the *HORNET* node.

to be N . N is determined by the ring size, bit-rate and slot size: $N=2000$ at 2.5Gb/s and $= 8000$ slots at 10Gb/s, for a 100km ring. The granularity of the above system is 1Mb/s.

The control channel carries a k -bit reservation vector where k is the number of wavelengths in the system. The source node sets the bit. The bit can be cleared either by the source or by the receiver, as will be evident from the following sub-section. It is important to note that once the bit is set, no node other than the source node can use the same wavelength in that slot (otherwise the purpose of the reservation is defeated).

Circuit Setup and Teardown Procedure

We will explain one method to setup and teardown a circuit using the source-based scheme. The main purpose of this is to provide context to the hardware design and implementation described later. Figure 2.28 shows the *HORNET* node structure that will be used for the description.

A. Setup:

1. Circuit request arrives at a source node. It is first processed by the circuit management sub-system. The circuit management sub-system figures out the destination (*HORNET*) node of the circuit and the bit-rate requested and maps them onto a *HORNET* wavelength and the number of slots (#slots) required to achieve the requested bit-rate, respectively. It can also take care of fairness and admission control locally amongst all the incoming requests.

2. The circuit management sub-system then generates a unique circuit ID (cctID) for the request and stores the combination (cctID; wavelength; #slots). This is the minimum state it needs to maintain. It then forwards (cctID; wavelength; #slots) to the circuit reservation and scheduling block. This is stored inside the reservation and scheduling sub-system in a specific manner, described later.

3. Assume that the source node is able to reserve a slot for the request, say slot *j*. Let's also assume that slot *j* is currently carrying a best-effort *HORNET* packet.

4. When slot *j* loops back to the node in 1 RTT (0.5ms for a 100km ring), the node sends a special circuit setup packet to the destination node. The setup packet carries the source's ID (this is a 7-bit ID for the *HORNET* node) and optionally the cctID. This packet could also encapsulate a circuit setup packet at a higher layer, for example a 'syn' packet in TCP circuit switching [3].

5. The destination node receives this setup packet on its receive wavelength and forwards it to the circuit management sub-system. The circuit management sub-system makes an entry for the new circuit and allocates resources such as queues. At this point a 1-way circuit has been setup.

The source node can now insert real data the next time slot *j* arrives at its input. Let's assume the source transmits data successfully. At the destination node the data in slot *j* will be switched into the correct queue if the data carries the source ID and the cctID. Another approach can be to maintain a control memory at the receiver

side of the node as well, that is indexed by slot j and carries routing information for slot j (for example, which queue to store the incoming data in etc.). Hence the worst-case setup time, measured from the time slot j is reserved, to the time real data is transmitted, is 2 RTTs (1ms for a 100km ring).

B. Teardown:

1. A teardown message arrives at the source node's circuit management sub-system

2. The circuit management sub-system forwards a teardown request ($cctID$; $\#slots$) to the circuit reservation and scheduling block. Optionally, the management sub-system can generate a teardown signal ($cctID$; 1) that indicates that only 1 slot needs to be cleared. It can generate this signal as many time as it needs.

3. At this point, there are multiple options: a. The source node clears the bit on the control channel in slot j and also uses slot j to send a teardown packet to the receiver. The receiver can then free up its resources. In this case the worst teardown time measured from the time the source node sends a teardown packet to the time the slot and the receiver resources are cleared is 1 RTT. b. Optionally, the source does not clear the bit but sends the teardown packet to the receiver and the receiver clears the bit. This seems difficult to perform in the same instance of the slot j , in which case the receiver can clear slot j after 1 RTT. In this case, the worse teardown time is 2 RTTs. c. Yet, another solution is to add another bit to the control channel, the teardown bit (k -bit vector for k wavelengths). When the source needs to teardown a circuit, it sets this bit to 1. The receiver can simply read the bit from the control channel and clear the reservation and teardown bit in the same slot. In this case, the worse teardown time is 1 RTT.

Network Simulations and Protocol Design

Goals and assumptions: The reservation protocol provides a slot reservation service at layers 1 and 2 that higher layers can use for setup of circuits. The goal of our simulator is hence to test the slot reservation protocol within the *HORNET* network. The performance results from the simulation are also used to finalize the design of the source-based scheme. Before, we start describing the simulations, we introduce the assumptions we make about circuit-setup models that could use our reservation protocol.

We assume that either the source or destination of the circuit is an end-user, for example, the circuit could be between a web-server and a PC. The source and destination of the circuit are connected to two *HORNET* nodes. The *HORNET* node connected to the circuit-source is called the source node while the *HORNET* node connected to the circuit-destination is the destination node.

In the first model, we can assume that a handshake has taken place at a higher layer in order to set up a logical circuit between the source and destination of the circuit. Once a logical circuit has been set up, the source forwards a request to (its) *HORNET* node in order to reserve slots for the circuit. From here on, the slot reservation protocol takes over. In a different model, the *HORNET* node receives and reads the higher layer request and uses it during the slot reservation process. This model has the advantage of speeding up the circuit setup process since the setup is happening simultaneously at all layers.

Network model: We have simulated a 32-node, unidirectional, *HORNET* network with one wavelength per node. Each wavelength has 320 slots. Each node has 31 VOQs (Virtual Output Queues); a VOQ stores incoming requests for a particular destination node/wavelength. We assume that a node does not transmit to itself; hence there are 31 VOQs not 32. Each VOQ is a FIFO queue. Nodes use a round robin match algorithm to decide which VOQ to service.

Traffic model: All nodes receive reservation requests from their local area side. The reservation request consists of the following fields: destination node, number of requested slots (proportional to requested bit-rate) and TTL (time-to-live). We use a Bernoulli iid process for the incoming requests, at each node. Hence the inter-arrival times are geometrically distributed, a discrete form of the Poisson process. The Poisson process agrees well with circuit requests processes, even though it is not representative of packet arrivals in the Internet. For each arrival, the destination can be chosen randomly or by using some weight, depending on the measured parameter.

TTL is the duration for which the slot reservation is active. After the TTL expires the reserved slot is cleared. TTL is modeled as a geometric random variable. $E[\text{TTL}]$ and $\text{var}[\text{TTL}]$ are set based on the parameters tested. For example: if the $E[\text{TTL}]$ of the requests is small, on an average more slots get freed per unit time and hence more new reservations can be made per unit time. Thus $E[\text{TTL}]$ will be an important variable in measuring throughput. Usually, we will express TTL in multiples of round trip times (RTTs).

For this set of simulations, we will assume that requests need to reserve 1 slot each, i.e. minimum granularity circuits.

Performance metrics and measurement parameters:

1. Maximum sustainable throughput (MST): For a slot reservation protocol, throughput can be defined as the number of slots reserved per unit time. As the arrival rate increases, the throughput increases until the network saturates. Further increase in the arrival rate only results in a monotonous build-up of the VOQs. The throughput measured at this saturation point is called the MST. It should be noted that MST depends on the size of network (number of wavelengths and ring circumference) as well as the $E[\text{TTL}]$.

2. Utilization: There are a finite number of slots circulating in the ring: equal to the number of slots per wavelength, times the number of wavelengths. Utilization is

defined as the ratio of the number of reserved slots to the total number of slots. As arrival rate increases the utilization should increase and saturate. Ideally, a protocol will be able to achieve 100% network utilization.

3. Service rate: This is the ratio of the number of incoming to outgoing requests, measured at the VOQs of the nodes. Sometimes the service rate of the entire network is measured. This metric can be used to determine the saturation point of the network. At the saturation point, the service rate falls below 1 and the queues build-up. Service rate is also a good metric for determining fairness between VOQs within a node and between nodes.

Results and Protocol Design

While designing the source-based scheme, there were a few design choices that needed to be made. The source node is responsible for reserving the slots; that portion of the protocol is fixed. But it is unclear as to who should clear the slots. The source gets a teardown request after the TTL has expired. The source transmits this teardown message to the destination so that the destination can clear its resources (memory) associated with the circuit. Hence, both source and destination are in a position to clear the slot. Assume the source clears the slot. In that case, the protocol can be greedy or non-greedy. Greedy implies that a (source) node can reserve a slot that it has just cleared. Hence greedy requires a node to perform 2 functions: slot clearing and reservation in the same time slot. Non-greedy implies that a source node cannot clear and reserve the same slot. The design choices are subtle, but have a big effect on performance.

We choose the source to clear the slot, for the following reason: if the destination, say node-K clears the slot, the node immediately downstream, node-K+1 (or K-1, depending on transmission direction), always gets the first chance to reserve the

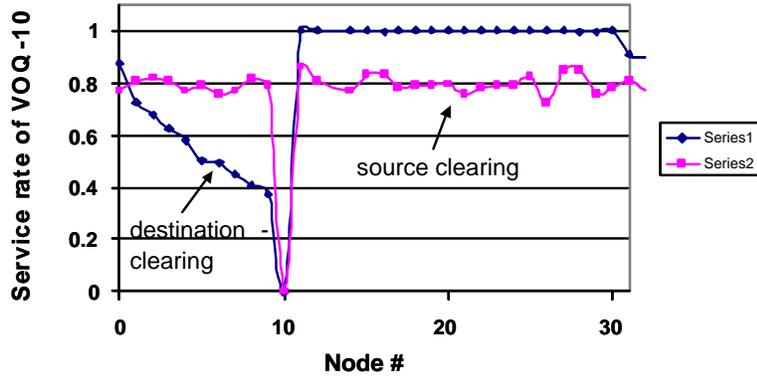


Figure 2.29: Service rate of VOQ-10 at all nodes: for source-clearing and destination-clearing.

cleared slot. In a congested network (under high utilization), this leads to the classic positional priority fairness problem seen in multiple-access ring networks. This is depicted in Figure 2.29 that plots the service rate of VOQ-10 for all nodes, for destination and source clearing. Note that this simulation assumes a unidirectional ring. The direction of transmission is such that node 2's transmission first passes by node 3, and so on. Observe that VOQ-10's service rate for the destination clearing protocol is 100% at nodes 11-30 and it decreases from node 31-9, with node 9 suffering the most. This is the classic problem created by destination clearing. We hence let the source clear the slot. The idea is that since slots are cleared by different sources, they will be cleared at different points in the ring, getting rid of positional priority. This is evident from Figure 2.29, in which the service rate curve of VOQ-10, for the source-clearing protocol, does not show any dependence on the node's position in the ring.

While source-clearing clearly gets rid of positional priority, destination-clearing protocols benefit from the so-called spatial reuse efficiency, resulting in higher throughput. We argue that spatial reuse is important in packet transport, but as the TTL of the reservation increases to a few RTTs, this efficiency becomes negligibly small. We

observed no difference in throughput for TTLs larger than 4-5 RTTs. Later on, we will show that file transfers for most applications will last ≥ 100 RTTs for a 1Mb/s circuit, on *HORNET*. Hence spatial reuse is not an issue in CoHO.

Source-clearing can still suffer from unfairness. If the source-based protocol is greedy, a hog source node can obviously lead to catastrophic unfairness issues. For that reason, we simply restrict nodes from reserving slots that they have just cleared, i.e. non-greedy. To test this, we simulate a 32-node network with uniform traffic to all destinations. The arrival rates at all nodes are equal and are kept high to stress-test the network's fairness. It is important to keep the stress level high because at low arrival rates fairness is not an issue. Figure Fig. 2(a, b) plots the service rate observed at all VOQs for nodes 2, 13 and 28 in a 32-node network, for the greedy and non-greedy versions. Note that each line hits zero at one point on the x-axis: node 2's service rate hits zero at destination 2 on the x-axis and so on. There are two conclusions to be made from Fig. 2. First, the VOQ service rate variation within a single node is very small for the non-greedy protocol: all VOQs inside a node are equally serviced. Second, the VOQ service rate variation across multiple nodes is also very small for non-greedy: VOQ-i in node 2 gets the same service rate as VOQ-i in node 12 and node 18. This discrepancy between greedy and non-greedy can be explained: once a node using the greedy-protocol clears a slot that it had reserved for a particular wavelength-f, the (same) node gets a chance to reserve it again. Since the stress on the network is high, a good fraction of the wavelengths in that slot are already booked, except wavelength-f that has just been freed. Hence the node ends up reserving the slot for the same wavelength-f. This results in the wild variation of service rates, within a node and across nodes, as can be imagined. The non-greedy protocol forces a node to let a slot it has just cleared to go away unreserved. This allows the next node to use it. Similarly, when that node clears the slot, the next downstream node gets to use it. This leads to a round-robin use of a particular slot. Hence no node

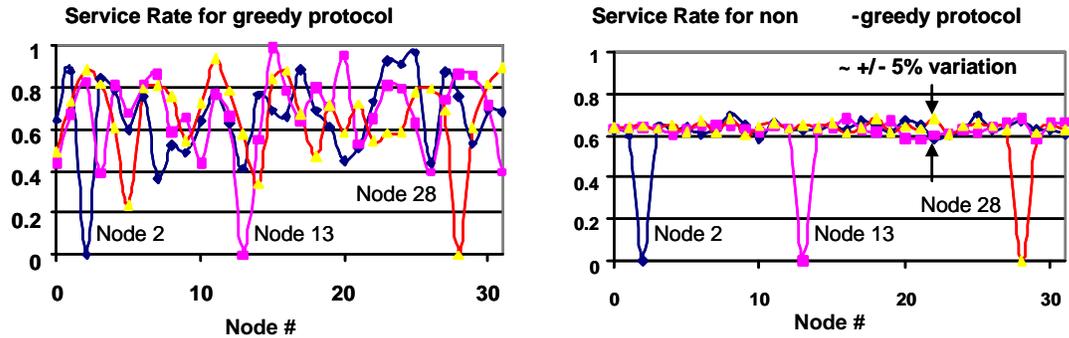


Figure 2.30: Service rate of VOQs at nodes 2, 13 and 28, for greedy (a) and non-greedy (b) protocols.

can hold onto a slot forever and in other words no node can be indefinitely starved.

In fact, the round-robin effect of the non-greedy protocol leads to a max-min fair share access. We ran a simulation that depicts this behavior in the case of a hot-spot destination node. Source nodes are given different arrival rate values with different starting times. Thus nodes become active one by one. We measure the % of the destination's bandwidth used by a particular source node (share of the bandwidth) and plot it in Figure 2.28. Node 1 is activated first with an arrival rate that is equivalent to 18% of the destination throughput. Figure 2.31(a) shows that node 1 gets all 18% that is requested since it is the only active source. The remainder of the destination bandwidth stays unused. Then node 2 gets activated. Its arrival rate is three times that of node 1. Figure 2.31(b) shows that node 1 keeps its 18% share, while node 2 gets its requested share of 54%. Node 3 is now active. Its arrival rate is twice that of node 2. Clearly all nodes cannot be satisfied. Node 1 gets its 18%, nodes 2 and 3 get an equal share of 41%, Figure 2.31(c). The destination is now completely booked. Node 4 then gets active. Its arrival rate is very small, half of node 1s. Figure 2.31(d) shows that indeed node 4 gets 9% share, half of node 1's 18%, while the two hogs nodes 2 and 3 get equal share. Note that although nodes 2 and 3 get equal share, node 2 the smaller hog, gets a share closer to its requested share (54%

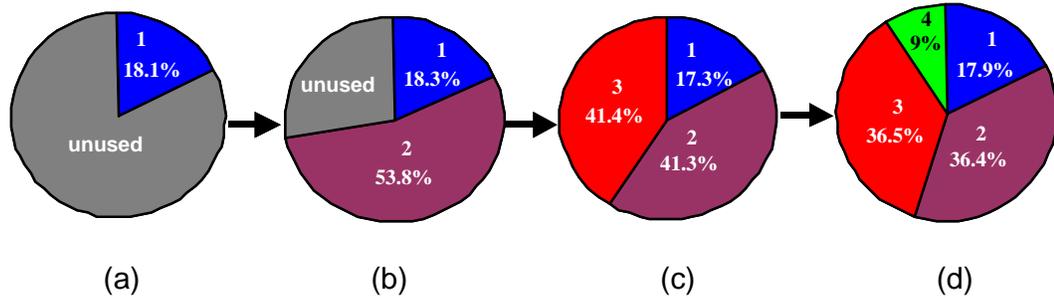


Figure 2.31: Max-Min fairness exhibited by the protocol: one hot-spot destination, multiple sources case

in Figure 2.31(b)), than does node 3. This is characteristic of max-min fair share protocols, in which the demands are satisfied in order of increasing magnitude. Max-min fairness is widely considered a good fairness mechanism for LANs and MANs. Since the non-greedy protocol prevents nodes from clearing and reserving the same slot, one expects its throughput and utilization to suffer. By contrast, the greedy protocol can achieve 100% utilization and high MST, at the expense of fairness. Greedy is thus used as a benchmark, only for this test. We measured and plotted the service rate and utilization of the entire network, for greedy and non-greedy, as a function of the arrival rate (Figure 2.32 (a, b)). Figure 2.32(a) plots this for $E[\text{TTL}] = 10$ RTTs. The absolute values are not of much interest presently, as much as the comparison of greedy vs. non-greedy. We can see that in both cases the service rate is initially $= 1$, it then starts falling as arrival rate increases. The arrival rate where the knee-point occurs is equal to the MST of the network. Beyond the MST, the network is saturated and VOQs back up monotonously. In general the network should be operated at an arrival rate $< \text{MST}$. Similarly, the utilization saturates for arrival rates $> \text{MST}$. It can be seen that the greedy protocol saturates at 100% utilization and has a higher MST than the non-greedy protocol. In Figure 2.32(b), $E[\text{TTL}] = 25$ RTTs. We can observe that the gap between greedy and non-greedy is smaller and

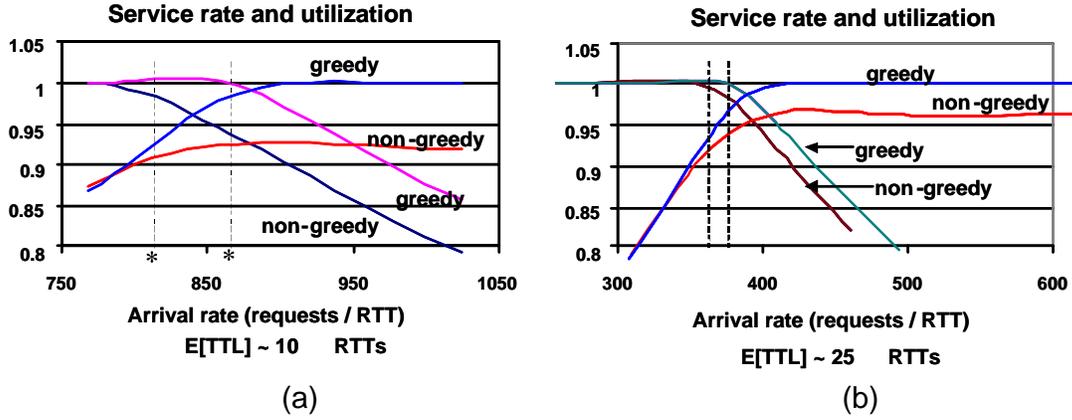


Figure 2.32: Service rate and utilization of the non-greedy protocol and dependence on $E[TTL]$

this gap continues to decrease for larger $E[TTL]$. We have measured 99% utilization for non-greedy and a MST equivalent to greedy, when $E[TTL] \geq 100$ RTTs.

We cannot be certain that $E[TTL] \geq 100$ RTTs unless we consider applications that might use a circuit service. But before that, we state a simple formula for the theoretical MST of the network. The formula can be derived using a basic fact: new reservations can be made only as fast as old reservations can be cleared. Consider the general case where there are m active source nodes and n active destination nodes. Number of slots per wavelength = $numSlots$. Then,

$$\text{MST per RTT for entire network} = (numSlots / E[TTL]) * m \quad (2.3)$$

If $m = n$ or $m/n < E[TTL] / (E[TTL]-1)$

$$\text{MST per RTT for entire network} = (numSlots/E[TTL]) * n$$

If $m/n \geq E[TTL] / (E[TTL]-1)$

$$(2.4)$$

It is important to note that the theoretical MST is dependent only on the $E[TTL]$

and the size of the network. It should not be confused with the performance of the protocol. The theoretical MST gives us an upper bound: the number of reservations made per RTT cannot be larger than the theoretical MST. The MST of the greedy protocol is equal to the theoretical upper bound, at the expense of fairness. For the non-greedy protocol, eqn. (1) is multiplied by a factor: $(E[\text{TTL}] - 1) / E[\text{TTL}]$, eqn. (2) stays unchanged. Hence in eqn. (1), the non-greedy protocol reaches the theoretical upper bound, as $E[\text{TTL}]$ becomes large. For $E[\text{TTL}] = 100$ RTTs, the MST is 99% of the upper bound. We have confirmed that the theoretical formula matches simulation results.

To calculate typical $E[\text{TTL}]$ values, let's consider three basic applications that may use circuit switching for data transfer. Currently they use packet switching like all applications using the Internet.

1. Storage to a file server: End users set up circuits to a file server that is located at a certain point in the MAN (storage node) and upload data. This can be the case of a hot-spot destination. Assume that average file size is 100KB.
2. Downloads from a web-server: End users send requests to the web-server. The web-server sets up a circuit back to the end user and transmits the web-content. This is the opposite of application 1, here one source sets up circuits to multiple end-users. Average file size for HTML transfers is much smaller = 6.4KB.
3. End-user to end-user file transfers: End-user A requests end-user B to setup a circuit from B to A. File is directly transferred from B to A. Currently, P2P applications use the packet switched Internet for transfers that mainly consist of MP3 files. MP3 files are big, average size = 5MB. One can imagine such transfers using circuits.

In all of the above applications, at least one end of the circuit is an end-user. Hence, the bandwidth of the requested circuit is limited by the end-user bandwidth. Let us assume that the end-user has a fast (1Mb/s) DSL connection. The reader may recall that a 1Mb/s circuit in a 100km, 2.5Gb/s *HORNET* network amounts to using

Average File Size	E[TTL] in RTTs	Source / Destination distribution	MST/RTT	MST/sec
5MB (MP3)	78,125	m=n=100 (full mesh)	4	8000
6.4KB (web page)	100	m=1 n=100 (one source)	25	50,000
100KB (file)	1563	m=100 n=1 (one destination)	2	4000

Table 2.1: E[TTL] for different applications and dependence of MST on E[TTL]

1 slot per RTT. Hence,

$$E[\text{TTL}] = \text{File Size} / \text{HORNET slot size} = \text{File Size} / 64\text{B} \quad (2.5)$$

We can thus calculate the E[TTL] and the theoretical MST of the three applications considered above. Table 2.1 below shows the results.

It should be noted that MST depends only on E[TTL] (file size), network bit-rate, network size and slot size. We can conclude three things from Table 2.1:

1. It can be seen that E[TTL] is ≥ 100 RTTs for the applications considered. Hence the MST of the non-greedy protocol will be equal to the theoretical upper bound. One may argue that if the bit-rate of the circuit is higher (e.g. 10Mb/s), the TTL will be lower. But today, most end users use 64kbps modems. 1Mb/s DSL will likely be the future bandwidth available to most end users. By the time 10Mb/s to the home is here, file sizes will be larger and hence average TTL will not change appreciably. Today, larger bandwidth circuits such as 10-100Mb/s will likely be used for aggregated (large) data transfers, once again requiring larger TTL values.

2. The web-server application has largest MST: 25 requests per RTT on average. Thus in a scenario where all three applications exist simultaneously, web downloads can dominate. Also note that the MST is the same as the maximum sustainable rate of circuit requests exchanged by higher layers. The protocol and network used to

carry these higher layer circuit requests should hence have a throughput \geq MST of the dominant application; in this case web-downloads.

3. Any lower layer circuit requests that are generated and transferred across the network in addition to the higher layer circuit requests amounts to circuit switching overhead. For example, in the web download dominated system, 25 slots per RTT (out of 2440) may be used to carry lower layer circuit requests, a 1% overhead. While 1% overhead is small, it is a point worth noting.

Hence from the simulations, we conclude that the source-based reservation protocol should be non-greedy and the source should clear the slots. We have seen that such a protocol is fair. It can achieve a utilization and MST that is 99% of the theoretical maximum for the three (reasonable) applications we consider. These results are tested for Bernoulli iid arrivals with geometric TTL. Work is underway to test the protocol under different conditions.

Design and implementation of the reservation and scheduling sub-system

From the above description it is clear that the one of the main components required to implement the source-based reservation in *HORNET* is the reservation and scheduling sub-system. For the purpose of the implementation, we assume that the source node reserves the slots for a new circuit and also clears a slot when a teardown signal is received (1 slot for each request). We do not consider the many variations discussed earlier, such as the use of a control memory at the receiver, the case where the receiver clears the reservation bit or the use of a teardown bit. Hence the sub-system performs one of the following three generic functions in each time slot: 1. Reserve a slot on a wavelength 2. Clear a slot for tearing down or reducing bandwidth of an existing circuit 3. Schedule data transmission on an already reserved slot Not mentioned above is the ability of the node to do none of the above.

To understand how the above functions are implemented, we will first look at the

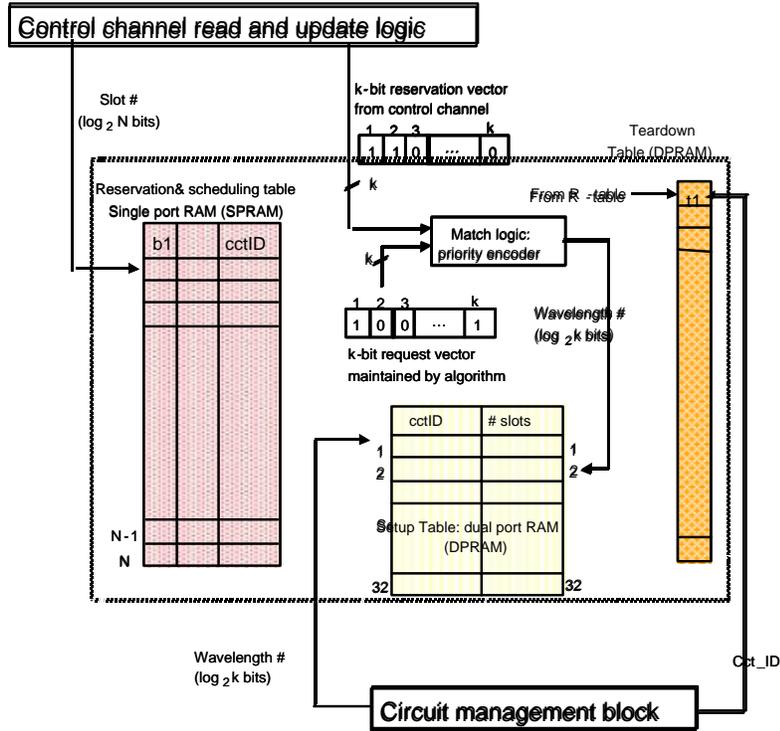


Figure 2.33: Building blocks of the reservation and scheduling sub-system

main building blocks of the reservation and scheduling sub-system. These are shown in Figure 2.33 inside the dotted block.

A. Reservation Table: The reservation table is indexed by the slot #. It stores the following information:

1. The 'b1' bit. If the b1 bit is = 1, the node has a reservation in that slot. If b1 = 0, the node is free to make reservation in that slot.
2. The wavelength #. This is a $\log_2 k$ bit vector, where k is the number of wavelengths in the system. This is the output wavelength that the tunable transmitter needs to transmit on in that time slot.
3. The circuit ID (cctID): The circuit ID is the unique identifier given by the circuit management sub-system, when the request is serviced and entered into the

setup table. The size of this vector should be long enough to represent all possible circuits maintained by the node. For example, in a system with N slots, a node can have a maximum of N simultaneous circuits. Hence, the $cctID$ should be $\log_2 N$.

For each incoming slot, the slot # increments, pointing to the next location in the reservation table. By reading the data at this location, the algorithm learns the reservation status of the node. This helps in either reserving a slot or scheduling transmission on a slot already reserved by the node.

B. Setup Table: The setup table is indexed by the wavelength #. It stores the following:

1. The circuit ID ($cctID$) of requests selected by the circuit management block. Hence it is $\log_2 N$ wide.

2. The number of slots that need to reserved per circuit request. This can be more than one, in general. The maximum size of this vector depends on the maximum size circuit allowed.

The setup table is implemented using dual port RAM. This is important, since there are two algorithms (processes) that need to access the setup table. One process is the reservation mechanism and other process is the circuit management sub-system that replenishes the table with circuit ID and #slots for new circuit requests. It should be noted that the circuit management sub-system can take care of admission control, fairness control between requests arriving at the node and any other policy decisions before updating the setup table.

C. Tear down table: The teardown table is indexed by $cctID$. It needs to hold a single bit. This bit is referred to as the $t1$ bit from here on. If the $t1$ bit = 1, the circuit ($cctID$) needs to be torn down. The teardown table is also implemented using a dual port RAM for the same reasons as the setup table: it needs to be modified by both the reservation mechanism and the circuit management sub-system.

D. Reservation vector: The reservation vector is read from the control channel,

in each incoming slot. It is a k -bit vector, where k is the number of wavelengths in the *HORNET* system. If the $(k-2)$ nd bit is set to 1, for example, it implies that the $(k-2)$ nd wavelength has already been reserved in that slot.

E. Request vector: The request vector is maintained inside the reservation algorithm logic. It is also a k -bit vector. If the $(k-2)$ nd bit is set to 1, for example, it implies that there is a circuit request sitting in the setup table for the $(k-2)$ nd wavelength.

F. Match logic: The match logic uses a priority encoder. The input to the match logic is a k -bit vector = (NOT (reservation vector) AND (request vector)). Hence a bit 1 at the $(k-2)$ nd location in such a vector, for example, implies that the $(k-2)$ nd wavelength is not reserved (on the control channel) and the node wants to reserve a slot on the $(k-2)$ nd wavelength. The output of the match logic is a $\log_2 k$ vector that points to the first instance of the bit 1, in the input vector. This is referred to as the matching wavelength #.

In the future, we might incorporate fairness by implementing a round robin priority encoder. In that case the wavelength # with the highest priority cycles through k different values. The match logic will change depending upon higher layer control such as network wide fairness. For example, it is possible that in a particular slot, a fairness algorithm (like DQDB) running at the node will force the node to reserve a particular wavelength.

We will now proceed to explain the algorithm that links the different blocks described above. Figure 2.34 shows a detailed state-machine diagram of the algorithm, the explanation below follows the figure quite closely.

The algorithm is in the Idle state until it receives a "go" signal from the control channel update logic. The slot number and the k -bit reservation vector are read-in by the control channel logic, at the beginning of an incoming slot. These are then presented to our algorithm and a 'start of slot indicator' (SOSI) signal is turned high.

When SOSI goes high, the algorithm latches in the slot # and the reservation vector and jumps out of the Idle state.

The slot # indexes into the reservation table. The contents of the reservation table are read and latched into internal signal vectors, regardless of their values. These are used throughout the algorithm. If the b1-bit in the reservation table is = 0, it means that the node does not have a reservation in that slot and can reserve on a free wavelength. Let us assume that the b1 bit = 0, for now. The matching logic will either generate a signal that says "match not possible" (this happens when $(\text{NOT}(\text{reservation vector}) \text{ AND } (\text{request vector})) = 0$) or it generates the matching wavelength #. In the first case, the algorithm goes back into the Idle state to wait for the next SOSI.

In the second case, the match is successful and the reservation process should go ahead. The matched wavelength # is first latched in. It is then sent to the control channel update logic to generate the new reservation vector to be transmitted on the control channel. The wavelength # is also used to index into the setup table. The contents of the setup table are read and latched. The circuit ID (read from the setup table above) and the match wavelength # are written into the reservation table. Note that the address pointer of the reservation table is already pointing to the right place, since the slot # has been latched in. The reservation table is thus updated. The algorithm then decrements the #slots counter (already) read from the setup table and writes the new #slots counter value into the setup table. Hence the setup table is updated. If the #slots counter reaches zero, it means that all the slots requested by the circuit have been reserved. In that case, the bit corresponding to the matched wavelength # in the request vector is cleared (zeroed out). The algorithm returns to an Idle state, waiting for the SOSI to go high again. If the #slots counter does not decrement to zero, the algorithm does not update the request vector and can return to the Idle state immediately (hence Function 1)

The request vector and the setup table need to be updated by the circuit management sub-system as well. For the purpose of this project, we assumed that the request queues inside the circuit management block are always full. Hence, when the #slots counter inside the setup table decrements to zero, the algorithm immediately brings in a new request and updates the setup table with it. Hence in effect the request vector is always = 1 (all k bits are 1). This is an interim solution, until the interface with the circuit management sub-system is fixed.

Going back to the start of the algorithm: the contents of the reservation table have been latched in. If the b1 bit = 1, then the algorithm follows a different thread. b1 = 1 implies that the node has already reserved that particular slot for itself, at some earlier time. Hence, it has only two options: tear down the circuit or transmit data.

The cctID (already read from the reservation table) is used to index into the teardown table. If the t1 bit = 1, it means that the circuit needs to be torn down. In that case, the algorithm will clear the b1 bit in the reservation table (b1 = 0). It will also inform the control channel update logic and give it the wavelength # from the reservation table. The control channel update logic in turn will clear the bit in the reservation vector before transmitting the vector on the control channel (hence Function 2)

If the t1 bit in the table is = 0 the node should transmit data. The cctID and wavelength # (read from the reservation table at the start) will be used to select a VOQ to transmit data from. Also the wavelength # will be given to the tunable transmitter as the target wavelength for tuning (hence Function 3).

Implementation Specifications and Results

A) Specifications The simulations use a clock frequency of 80MHz. This was chosen for two reasons: (a) the *HORNET* data transmission sub-system uses 32-bit logic

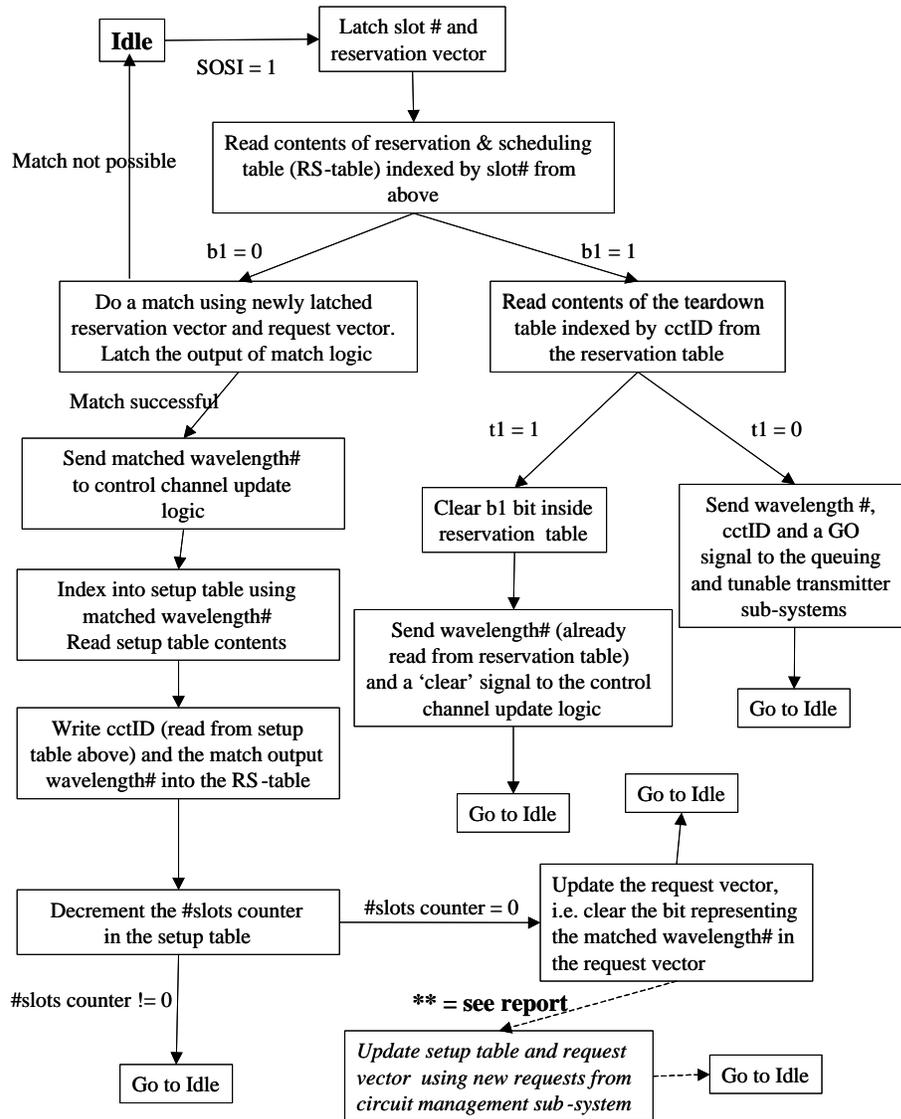


Figure 2.34: State machine diagram of the algorithm used inside the reservation and scheduling sub-system (and implemented in VHDL)

with a 77.76MHz system clock ($77.76 * 32 = 2.488\text{Gb/s}$) and (b) the simulator cannot handle fractional clock rates, the clock period at 80MHz is 12.5ns, a simple number to work with while measuring delays on the timing diagrams. The slot size is 200ns: this corresponds to a 64Byte *HORNET* slot at a bit-rate of 2.5Gb/s and a circuit granularity of 1Mb/s for a 100km ring.

B) Results: Hardware requirements:

The VHDL compiler/synthesizer used for this work is a software program purchased from Cypress Semiconductor, specially designed to work with their PLD devices. The program synthesizes the VHDL code and fits it to a device specified by the user. After the fitting is over, it generates a report file that includes the amount of memory bits, macrocells and I/O ports used. The basic logic block inside a PLD consists of a programmable AND array followed by OR gates with fixed number of inputs (AND-OR logic). The outputs of the OR logic are latched using flip-flops in the macrocell and can be fed back to the programmable AND array as inputs or passed on to other logic blocks or to the I/O pins. Hence each macrocell has 1 flip-flop. The report file thus indirectly indicates the number of flip-flops used by the logic. The report file does not give the number of gates. It is difficult to find the number of gates actually used since the PLD fitting tends to waste a lot of combinatorial logic elements (unless the code is mainly combinatorial). For example, if a constant is stored in a flip-flop, then the AND-OR logic connected to that macrocell is wasted. Our VHDL implementation used 50kbits of memory (on-chip SRAM) and 350 flip-flops. It seems like a very modest set of requirements, especially considering the excellent performance observed in the timing diagrams. The reader should be reminded that this sub-system is only a part of the entire reservation mechanism, albeit a very important one.

A framework for comparing reservation mechanisms on *HORNET*

It is important to develop a framework in order to compare reservation mechanisms. It is equally important that the framework be consistent and applicable to the *HORNET* architecture. In this section, we try to develop a framework that includes both lower layer performance metrics and hooks provided to higher-layer control mechanisms.

1. Lower layer performance metrics

- A) Hardware requirements:

Most reservation schemes will either be implemented using PLDs, FPGAs or custom ASICs. In such a case, the specific requirements can include the number of flip-flops, gates, I/O pins. Most schemes will make use of memory to store tables (e.g. reservation table in Section 4). Important parameters for memory requirements include number of memory bits used, speed of memory, whether the memory is on-chip SRAM or off-chip SRAM, or maybe DRAM. The importance of SRAM versus DRAM is obvious: DRAM costs much less than SRAM and has higher density, although access speeds are much slower than SRAM. On-chip versus off-chip SRAM is important because most medium size and medium priced programmable devices, such as the Cypress CPLD CY39K series, the Altera APEX20K series FPGAs, etc. have (roughly) $< 500\text{kbits}$ of on-chip SRAM.

- B) Scalability with bit-rate (and system clock rate)

This metric seems obvious at first glance, but it has special significance when it comes to choosing the right reservation mechanism. The significance of this metric is best explained with an example. The *HORNET* system uses a system clock of 77.76MHz for a bit-rate of 2.5Gb/s (32-bit logic). The goal of *HORNET* is to scale to 10Gb/s per node. If we assume that today's (absolute time does not matter in this calculation) metro area networks need 2.5Gb/s per node, the system will need to scale to 10Gb/s within 2 years (24 months). If we assume that processing speed scales with Moore's law (doubles every 18 months), we can expect that a 10Gb/s *HORNET*

node will use a system clock of 200MHz (5ns period). Minimum access times for SRAMs are very close to 5ns, a potential problem. Also, while the *HORNET* slot size reduces by a factor of 4 (from 200ns to 51.2ns) the system clock increases only by a factor of 2.65. Referring to the results in Section 4: it takes 10 clock cycles for the worse case slot reservation. 10 clock cycles at 200MHz is 50ns, barely less than the slot-size at 10Gb/s. Hence careful design of the interface with the circuit management sub-system is required to reduce the time further.

C) Granularity

We have seen in the source-based scheme that the circuit granularity was strongly dependent on the size of the ring: a 100km ring with a 64Byte slot size resulted in a 1Mb/s granularity. 100km is a typical circumference (size) for a metro network. If we assume that there is a +/- 50km variation in the ring size (a reasonably large variation), the granularities obtained are 2Mb/s for a 50km ring and 650kb/s for a 150km one. This may or may not be a problem, although it is an annoyance for a system designer. If a service provider considers this to be a problem, modifications need to be made to the source-based scheme, or else a new scheme should be chosen.

Intuitively, it seems possible to de-couple the granularity and the size of the ring. Here is one possible solution used in conjunction with the request-grant iterative scheme. Assume that a maximum of 2000 slots can physically live in the ring, but we want to have 6000 logical slots. Consider the following numbering scheme. The master node in *HORNET* starts generating slots as usual. When it receives slot-1 after it loops back, instead of simply copying it to its output, it rennumbers slot-1 as slot-2001. Also slot-2 becomes slot 2002, Slot-2000 becomes Slot-4000, which eventually becomes Slot 6000. Slot 4001 becomes Slot1 etc. Figure 2.35 shows a ring network with 1 receiver, R1 and 3 sources S1, S2 and S3. There is a master M. For each receiver, there is a space X1 and a space Y1, as shown. Figure 2.36 shows the tables maintained at R1, S1, S2 and S3.

It can be seen that R1 maintains a table with 2 logical parts - one for the nodes that lie in the X1 portion of the ring, and the other for the Y1 nodes. Now we will follow the request that go from sources to R1 and grants from R1 to the sources and check how the tables are updated. Consider request r from S1 to R1. When R1 gets this request, it checks whether S1 lies in the X1 part of the ring, or the Y1. In this case S1 is a Y1 node. Thus R1 will give S1 a free slot from its table using the Y1 side of the table. So R1 returns slot# 1. S1 mark slot 1 as busy, as usual; but R1 marks slot 1 busy in the Y1 part of its table and marks slot 2001 as busy in the X1 part of its table as well. This is because slot 1 becomes slot 2001 after it goes through the master node. Now, S2 sends a request. R1 looks into the Y1 portion of its table (again) and gives slot 2 in the grant. S2 marks slot2 busy, as usual; but R1 marks slot 2 busy in the Y1 part of its table and marks slot 2002 busy in the X1 part of its table as well (same reason as before). Now S3 sends a request. R1 now checks the X1 part of the table and returns slot 1. R1 now marks slot1 busy in the X1 part and marks slot 4001 busy in the Y1. The reason for this is that none of the Y1 nodes should use slot 4001, because after it passes through the master, the same slot becomes slot 1. Hence in this way, by marking one slot in the X1 side and the adjusted slot in the Y1 and vice versa, we can avoid collisions and yet handle 6000 slots. This scheme seems extensible to larger or smaller number of slots (8000 or 4000 foe example). We will analyze this technique further, in our future work. It seems very promising to decouple the size of ring and the circuit granularity.

D) Circuit setup and teardown times The importance of circuit setup and teardown times will change with different applications. For example, if the typical usage of circuits over *HORNET* would be for supporting slowly reconfiguring (with the time of day etc.) circuits, the question of speed of setup/teardown becomes moot. On the other hand, if a circuit will be used by an end user for traditional applications such as http:// (web downloads) or ftp, where typical activity times can be of the order

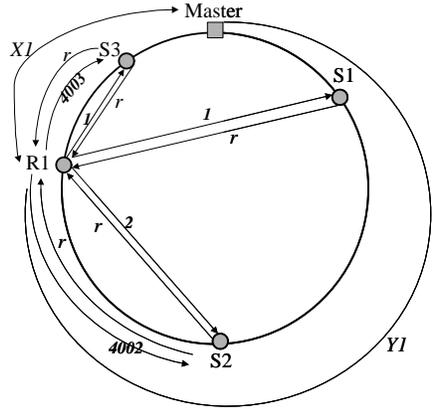


Figure 2.35: Sample reservation on Ring Network

Reservation Table @R1		Scheduling Table @ S1	Scheduling Table @ S2	Scheduling Table @ S3
X1	Y1	R1	R1	R1
1	S1			
2	S2		R1	
3				
	⋮			
2000				
2001	S1			
2002	S2			
2003				S3
	⋮			
4000				
4001				S3
4002				S2
4003	S3			
	⋮			
6000				

Figure 2.36: Corresponding reservation tables for sample reservation

of a few seconds, speed of setup/teardown is important. For this work, our focus has been on supporting circuits that last a few seconds rather than hours, TCP circuits for example.

We have seen that the setup/teardown times for the source based scheme are 1 or 2 RTTs, depending upon how we measure these times and whether variations to the base scheme are used. In addition once the setup is done, the rest of the packets will go through the network with almost fixed delays. That seems very reasonable. The setup time will change based on the load seen by the source and the destination node and it will also depend on the fairness scheme used. We cannot quantify this dependence without further analysis and/or simulation and this is left as future work.

On the other hand, for the request-grant iterative scheme, for low load at source and receiver, the probability of success (matching slot is found) in one iteration will be very high. So in 1 RTT a match is found, say slot j . When the grant is received at the source, the worse case scenario is that slot j has just gone past the source. In this case it will take 1 more RTT for slot j to arrive at the input of the source node, the source uses that slot to transmit the setup-packet and hence the total setup time will be 3 RTTs. This is again a reasonable number. But the probability of success of the iterative scheme reduces as the source get congested and each iteration results in 1 RTT delay. It is our conjecture that with a large number of slots (1000s) the probability of success should be fairly high even under high load. Our conjecture will be quantified in our future work.

E) Throughput Throughput is a very important metric as well. *HORNET* can be represented as a re-arrangeably non-blocking network. Hence the goal of a good reservation mechanism should be to make connections in such a way, so as to maximize throughput. This is a difficult problem, especially when connections cannot be rearranged. If a non-blocking network is desirable, we would like to think of ways

to make *HORNET* strictly non-blocking. In a 3-stage network, such as a TST network, by making the input TSI module and expander stage ($N/2N$) and the output side TSI a concentrator ($2N/N$), the TST switch is made non-blocking. Similarly, if we double the bandwidth of the *HORNET* center stage (the wavelength switched ring), it might be possible to achieve non-blocking properties. Since *HORNET* uses bi-directional architecture, this doubling of capacity might be available without an increase in bit-rate. We would like to explore these options in the future, in terms of need and feasibility.

Higher layer control mechanisms

Service order and fairness

The reservation mechanisms described so far lack discipline in the way they function. For example, in the source-based scheme, when a circuit request arrives at a node and is entered into the setup table, the node grabs the first available slot. While this is acceptable in a network with low utilization, there is no way to predict the behavior of such a mechanism at high loads. Hence we want to impose some form of service order on top of the reservation mechanism. What it boils down to is the following: if a slot gets free on a completely congested wavelength, which node gets it next? This is exactly the same problem solved by output queue scheduling algorithms such as first-come-first-serve (FCFS), round robin (RR), weighted fair queuing (WFQ) etc. But from section 3 it is evident that *HORNET* is like a crossbar with input and output constraints. So, the service order that can be imposed using FCFS, RR etc. is a modified one. For example the modified FCFS algorithm can be summarized as follows: if a slot gets freed, it is used by the first 'free' source node, in the FCFS order. We plan to quantify the performance degradation or unfairness due to this modification using simulations. In general all reservation schemes will allow a hook to establish service order, leading to a form of fairness. In this work, a FCFS

service order will be considered fair. The reasoning behind it is that although FCFS is unfair from a network node's point of view (a greedy node sees a smaller average delay than a non-greedy one), it is fair from an end users point of view. This can be explained using a simple example: imagine an end user's request that arrives at time t_0 to a nearly full queue. Another end user's request arrives at a different node, at an empty queue, at time $t_1 \gg t_0$. If a max-min fair scheme such as RR, WFQ (the more traditional fair schemes) is used it will allow the request that arrived at time t_1 to be served before the request at time t_0 . Hence from an end user's point of view, this is unfair. We conjecture that the source-based scheme can use an extension of the DQDB (Dual Queue Dual Bus) protocol. In DQDB the node maintains a wait counter and a request counter. When a node gets a circuit request, it sends a request upstream. All nodes upstream increment their request counter when they see requests from downstream nodes. Hence when a free slot comes along (freed at the receiver for example), the nodes let as many slots go by as the value in their request counter. If in the meantime, a request arrives at an upstream node, it will copy its request counter into the wait counter. It will then proceed to let as many empty slots go by as the value of its wait counter, before using an empty slot for itself. For *HORNET*, we would have to maintain counters for all wavelengths. Like mentioned before, a FCFS scheme such as this would be 'modified' due to the I/O constraints of *HORNET*. Since in both the receiver-based scheme using broadcast and the request-exchange scheme, source nodes send request to the receiver, it seems better suited to handle fairness. Fairness can be established by serving requests from queues based on an order such as FCFS, RR, WRR etc. The deterministic scheme by nature allows nodes to get equal share under high congestion. Hence a fairness control is not required in this case.

2.10 Quality of Service Summary

In this Section, the architecture for *HORNET* was presented. With the use of fast-tunable packet transmitters and wavelength routing, the architecture requires significantly less equipment than conventional networks. The architecture can survive a fiber cut or node failure. The survivability scheme of the *HORNET* architecture utilizes all bandwidth and equipment for working traffic, which again makes the architecture less expensive than conventional architectures, which only utilize half of the bandwidth and equipment.

A novel suite of protocols was developed that make the *HORNET* architecture practical. First, a MAC protocol that is optimized for variable-sized packets was designed. Second, a fairness control protocol was developed that gives all *users* equal opportunity to access the network. Both protocols use a control channel for carrying information around the network. Design options to keep the cost of the control channel link down were presented in this chapter. Third, a protocol was developed to maintain precise synchronization between the information on the control channel wavelength and the packets on the payload wavelengths. In addition, a mechanism by which to support fixed bit rate traffic over *HORNET* was developed. In summary, the combination of the *HORNET* architecture and the protocols results in a practical, cost effective solution for high-capacity next-generation metro networks.

Chapter 3

HORNET Network Simulations

3.1 Introduction

In Section 2, the virtues of the *HORNET* architecture are discussed using intuitive arguments. Though the arguments are convincing, it is nonetheless necessary to quantitatively verify the validity of the intuitive arguments. Additionally, the performance of the *HORNET* protocols and any penalties associated with them must be determined quantitatively. To do this, computer simulations have been developed specifically for *HORNET*. The simulator precisely models the operation of the *HORNET* network while performing measurements for analysis.

In this section, the simulator is described and the results generated using the simulator are presented. The simulator is used to explore the performance penalties due to the *HORNET* protocols as well as the ability of the DQBR fairness protocol to provide equal opportunity to all users. Also, a quantitative comparison between *HORNET* and *RPR-over-WDM* is presented using the *HORNET* simulator and a similar *RPR-over-WDM* simulator.

3.2 *HORNET* Simulator Design

3.2.1 Basic Concepts of the Simulator

The simulator was created using object oriented programming techniques. A diagram showing the general construction of the simulator is shown in Figure 3.1. The simulation iterates over time steps while iterating over all N nodes during each time step. In the simplest case, the time duration of a time step iteration is equal to the time duration of a control channel frame. While operating on a node, the simulator performs statistical arrivals at the input of each VOQ in the node. The packets arriving at the queues in a particular node are the statistical sum of packets being generated by hundreds of users that are accessing the Internet through that node. If it is determined that a packet arrives at a VOQ within the node, the current time (in time steps) is written to the end of a vector which represents the particular VOQ. The vector length is therefore increased by one.

The simulator then decides which packet to transmit. In the base case, this simply requires determining which wavelengths are available to the node during that time step and then choosing the oldest packet that is at the front of a VOQ corresponding to a wavelength within the set of currently available wavelengths. In practice, other algorithms may be used for selecting which packet to transmit, such as *longest queue first* or *maximal size matching*. In this work, however, packet switching algorithms are not evaluated, so *oldest packet first* is used because it is fair in the FCFS sense. After selecting the packet to transmit, the simulator removes the time stamp from the front of the VOQ that was chosen to transmit the packet and moves all other time stamps forward one spot. The node can only transmit one packet per time step per transmitter.

Two $M \times W$ arrays are used to maintain the availability of each wavelength on each of the two rings, where M is the number of control frames existing in the ring, and

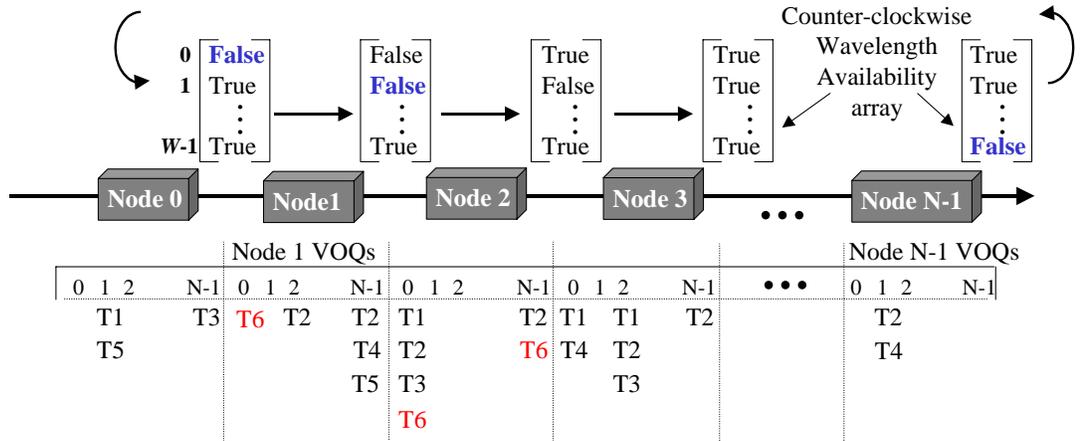


Figure 3.1: Diagram of the *HORNET* simulation architecture. The " T_n " represents packets that arrived to the nodes' VOQs during time step n . In the diagram shown here there is no propagation delay between nodes (i.e. the number of columns in the availability array equals the number of nodes).

W is the number of wavelengths. In the simplest case, the number of control frames is set to be equal to the number of nodes (i.e. there is no propagation distance between nodes). The occupation of a wavelength by a packet during a frame is represented with a Boolean value. Thus, each $M \times W$ array is filled with Boolean values, where 'true' implies that a packet exists on that wavelength of that frame. There are two arrays because one array represents the clockwise (CW) travelling traffic while the other represents the counter-clockwise (CCW) travelling traffic. Each column in the array corresponds to the availability of each wavelength during the frame passing through the node during the current time step (i.e. $\text{array}[n,w]$ is the Boolean value of wavelength w in the frame passing by node n). When a node inserts a packet, the value corresponding to the wavelength that packet uses is changed to 'true' in the wavelength availability array. At the end of the time step iteration, the traffic array is rotated one position around the ring in the appropriate direction of propagation. In each time step, all nodes remove the packets on their drop wavelength(s), which

means that if node n receives wavelength w , $\text{array}[n,w]$ is set to 'false' in every time step.

The expected latency (the average time that a packet spends in a VOQ) is an important statistic measured by the simulator. After a packet is transmitted, the simulator determines the packet's age in time steps (recall that when the packet arrived, the arrival time was recorded). It then adds this age to a cumulative age it stores for each VOQ in each node and increments a counter that is tracking the number of packets that each VOQ transmits. From these two values, the average delay in each VOQ is calculated at the end of the simulation. The maximum capacity of the network can be determined by locating the network load at which the average packet latency asymptotically approaches infinity. From queuing theory, this generally occurs when the rate of packet arrivals into the queue exceeds the rate of packet transmissions from the queue.

Simple intuition can be used to approximate the theoretical maximum performance of the network simulated. If there are N nodes on a *HORNET* network, and each node receives a unique wavelength (i.e. there are N nodes and N wavelengths) and uses only one transmitter per direction, then each transmitter for each direction will transmit to $\frac{N-1}{2}$ destination nodes. If traffic is uniformly distributed, then each transmitter can use at most $\frac{2}{N-1}$ of the bandwidth in each wavelength. Thus, each transmitter will be able to transmit its maximum bit rate in each direction, so that each node contributes a capacity of twice the transmission bit rate to the overall network. For example, for a 50 node ring with 50 wavelengths where all transmitters can transmit at 10 Gb/s, the intuitive maximum capacity of the network is $50 \times 2 \times 10$ Gb/s, or 1.0 Tb/s. Similarly, for a network of 33 nodes and 33 wavelengths, the maximum capacity should be 660 Gb/s. Figure 3.2 shows the simulated average packet latency for the *HORNET* architecture for networks with 33 and with 50 nodes. The results match the intuition.

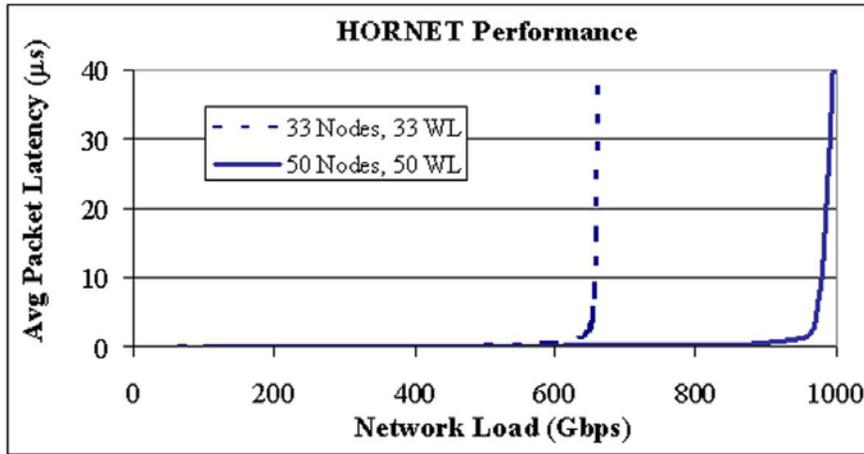


Figure 3.2: Simulated performance of *HORNET* networks with 33 nodes and 33 wavelengths and with 50 nodes and 50 wavelengths.

3.2.2 Variable Packet Sizes

The results shown in Figure 3.2 are generated using a simplified simulation in which all of the packets are of the same duration as the control channel frame (and thus the time step iteration). Naturally, this is not realistic, as IP packets are variable in size. The distribution of IP packet sizes is discussed in Section 2.3.2. Also, in Section 2.6.1 an approximate distribution is presented in Figure 2.16. Estimated distributions such as this one are used in the simulations described in this chapter. Doing so makes the simulator’s packet arrival process more realistic.

Recently in networking simulations research, there has been an emphasis on trying to make packet arrival processes as realistic as possible. Much has been reported about the value of using a *self-similar random process* as opposed to using a Poisson process to generate the packet arrivals at a node’s VOQs [40, 41]. Self-similarity of Internet traffic can be seen at a variety of granularities. Looking at the byte level, a Poisson process generates packets that are all of a small, fixed length that is generally equal to the time step of the simulation, while a self-similar random process provides

a significant probability that packets are longer than one time step. Backing out a step, self-similarity can simulate the fact that one packet is likely to be followed by several other packets as part of a flow (such as a Web site or file download). Backing out another step, *flows* are self-similar because typically a user will generate a series of flows as part of an Internet browsing or downloading session.

In the simulations in this work, self-similarity at the byte level is accurately included by using packet size distributions similar to that described in Section 2.6.1. The self-similarity at other granularities cannot be included in these simulations because the complexity of the simulation limits the tolerable simulation length to about one second of simulated events. Additionally, at coarser granularities, the influence of the networking and transport layer protocols must be considered, making the simulations even more complex. Therefore, to consider burstiness at coarser granularities, a simulator with the complexity at the higher layers should be developed. For practical reasons, such a simulator should probably attempt to simplify the lower layers and thus only model the complexity of the simulator presented in this work. However, the focus of this work is to understand the performance and penalties of the lower layer protocols, such as the MAC and fairness protocols. As such, the current simulator, which features an accurate model for variable-sized packets, is sufficient for this work.

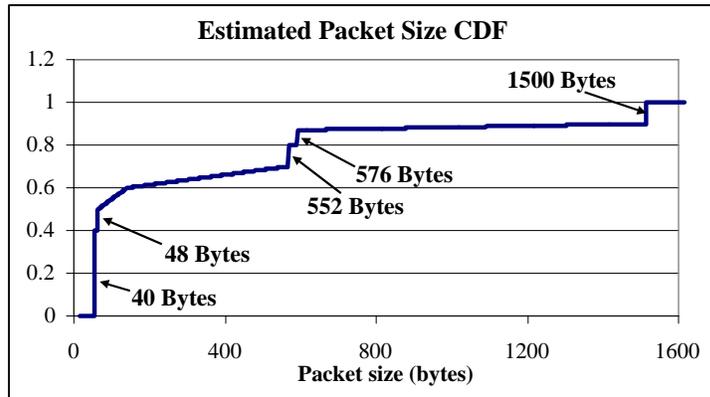
The variable-sized packet arrivals are modeled in these simulations as follows. As described in Section 3.2, in each time step each node iterates over its VOQs and determines whether a packet arrives in the VOQ. If a packet is generated, the simulation determines its length. After it is determined, the length is written to the end of a dynamic array that is parallel to the VOQ that holds the arrival time stamp. The simulation uses a probability density function (PDF) specified by the simulation user to randomly determine the packet length. The parameters for the design of the PDF are based on the data collected in [32]. The PDF of X , where X is the packet length in bytes, is specified as follows:

$$\begin{aligned}
\Pr[X = 40] &= a; \\
\Pr[X = 41] &= \Pr[X = 42] = \dots = \Pr[X = 47] = b; \\
\Pr[X = 48] &= c; \\
\Pr[X = 49] &= \Pr[X = 50] = \dots = \Pr[X = 125] = d; \\
\Pr[X = 126] &= \Pr[X = 127] = \dots = \Pr[X = 551] = e; \\
\Pr[X = 552] &= f; \\
\Pr[X = 553] &= \Pr[X = 554] = \dots = \Pr[X = 575] = g; \\
\Pr[X = 576] &= h; \\
\Pr[X = 577] &= \Pr[X = 578] = \dots = \Pr[X = 1499] = i; \\
\Pr[X = 1500] &= j; \\
a + b(48 - 41) + c + d(126 - 49) + e(552 - 126) + f + g(576 - 553) \\
&+ h + i(1500 - 577) + j = 1.
\end{aligned}$$

The simulation user specifies the values for a through j . The simulator is written such that nodes on the network can have different PDFs. This is intended to model the fact that some nodes may be in residential areas where the users will generate a packet size PDF weighted toward smaller packet sizes, while other nodes may be sending large flows, causing the packet sizes to be larger. Figure 3.3 shows a cumulative distribution of packet sizes with the values of a through j specified.

3.2.3 Segmentation and Re-assembly on Demand

The MAC protocol sometimes requires a node to segment packets during transmission using SAR-OD, as described in Section 2.3.2. The simulation handles SAR-OD as follows. When the node is transmitting a packet longer than a frame, the simulation maintains transmission of that packet in each frame until either the packet is completely transmitted or until the node must segment it. In each time step that the node is able to transmit part of that packet, the appropriate number of bytes is subtracted from the corresponding length value that is stored in the packet length



$$a = 0.4; \quad b = 0; \quad c = 0.1; \quad d = \frac{0.1}{77}; \quad e = \frac{0.1}{426}; \quad f = 0.1; \quad g = 0; \quad h = 0.07; \quad i = \frac{0.07}{923}; \quad j = 0.1$$

Figure 3.3: A cumulative distribution function of packet sizes modelled by the simulator.

array. When the node is forced to break transmission of the packet, it determines a new packet to transmit, and the remaining bytes of the segmented packet are once again a packet (though indeed a smaller one) waiting to be transmitted at the front of the packet length array. When enough bytes have been subtracted (transmitted) from the value at the front of the packet length array that it equals zero, the packet is complete, and the time stamp is removed from the front of the time stamp array. All values in the time stamp array and the packet length array are moved forward one spot (as if in a VOQ).

3.2.4 *HORNET* Overhead

The simulation results presented in Section 3.2.1 assumed that packets are sent without any overhead added onto the IP packet as part of the *HORNET* protocols. In reality, extra overhead must be added to the packet for all of the *HORNET* protocols. The overhead of a *HORNET* packet is discussed in Section 2.6.1. In the *HORNET* simulation, whenever the node begins transmitting a packet, it must first insert the

header. Thus, if the frame length is 64 bytes and the header is 16 bytes, during the first frame the node can only send (and thus subtract) 48 bytes from the packet that is currently being transmitted. If the packet continues into the next frame, then the simulator can subtract 64 bytes from the remaining length, assuming that there are at least 64 bytes remaining in the packet. If the packet is segmented, then the node must insert header bytes again when it resumes transmission of the packet. Thus, segmentation of packets adds extra overhead that detracts from the performance of the network. The simulator tracks the number of overhead bytes transmitted as one of its important statistics.

Figure 3.4 analyzes the performance of *HORNET* when overhead and packet segmentation and re-assembly are included in the simulation. The graph shows the performance penalty due to the overhead in a 17-node network. The curve showing the highest capacity is the result of simulations with small fixed-sized packets and with no headers applied to the packets. The curve to the left of it uses variable-sized packets. The average latency increases because longer packets take longer to transmit and because there is now a small amount of overhead due to packet size mismatch with the control frame size, as discussed in Section 2.6.1. The third curve in Figure 3.4 is the simulation result when 16-byte packet headers are applied to all packets in the network. As the figure shows, a penalty is incurred due to the additional overhead. The penalty resulting from the use of variable-sized packets and packet headers is approximately 15%. The SAR-OD protocol is used in both of the simulations that consider variable-sized packets.

The distribution of packet sizes will affect the performance of the network because of the effect it has on overhead. A packet size distribution that has a smaller average packet size will have higher overhead because the *HORNET* header will on average consume a larger fraction of the transmitted packet. The influence that a packet size distribution has on overhead in *HORNET* is shown in Figure 3.5. Figures 3.5 (a)

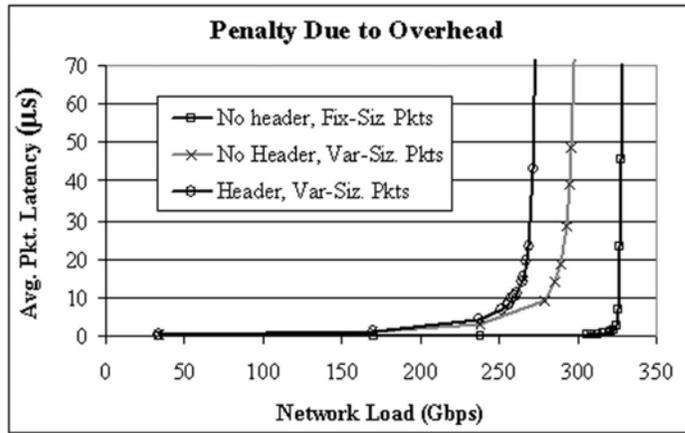


Figure 3.4: This graph shows the penalty incurred for the use of variable-sized packets and 16-byte packet headers.

and (b) show two different packet size distributions that have some of the same characteristics as those discussed earlier. Figure 3.5 (c) shows the overhead percentage measured by the simulator for each of the two distributions. As suggested, when the average packet length is longer, the overhead is lower. Another interesting observation in Figure 3.5 (c) is that when the traffic rate increases the overhead increases. This is a result of the fact that packets are segmented more frequently in the SAR-OD protocol when the traffic rate through a node is higher. The overhead increases because the *HORNET* packet header must be applied to every segment transmitted.

3.3 Optimal Control Channel Frame Size

In Section 2.6.1, it was suggested that 64 bytes is the best selection for the control channel frame size based on the comparison of minimum possible overhead for varying control channel frame sizes, as shown in Figure 2.17. This hypothesis can be verified using the *HORNET* simulator. Figure 3.6 compares the performance of a *HORNET* network with control channel frame sizes of 40, 56, 64, and 200 bytes while using the

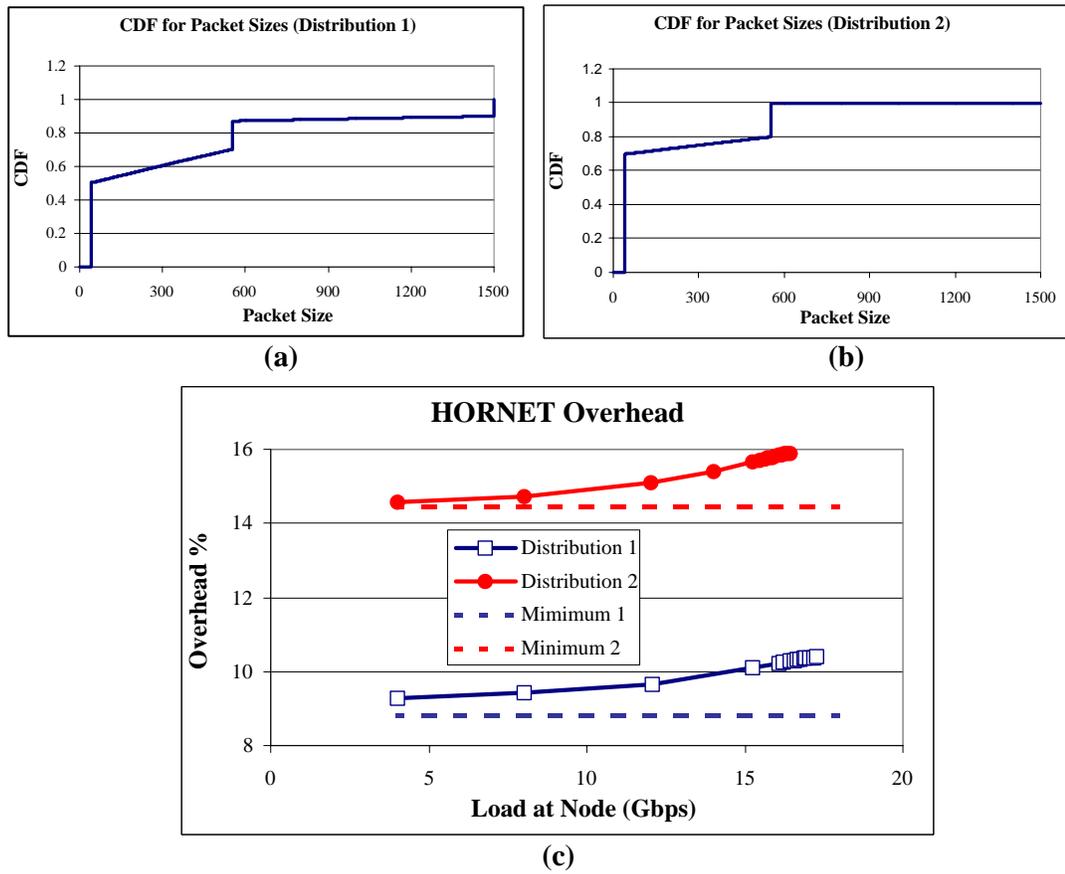


Figure 3.5: Impact of packet size distribution on overhead, and thus performance. (a) Distribution 1, which is similar in average packet size to previously shown distributions. (b) Distribution 2, which has a smaller mean packet size. (c) Overhead measured by the simulator for the two distributions. The minimum lines in (c) are the calculated overhead if no packets are segmented.

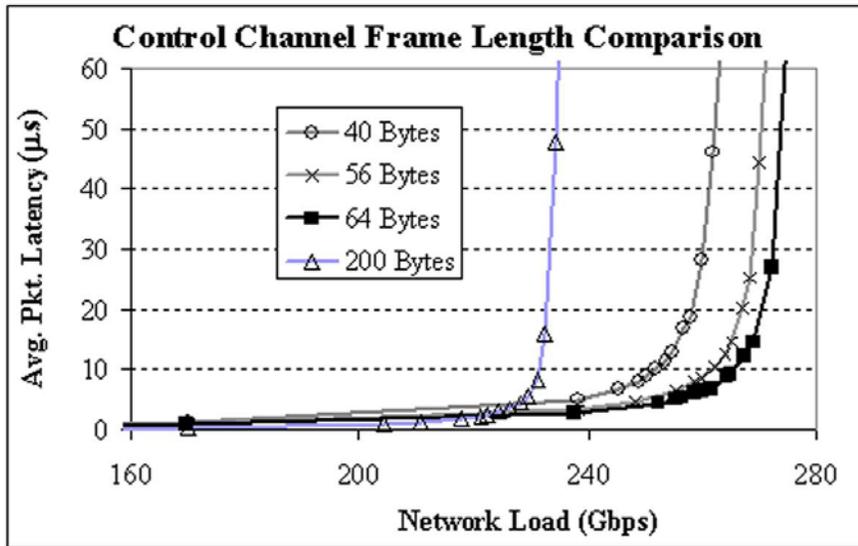


Figure 3.6: Simulated performance of *HORNET* with control frame sizes of 40 bytes, 56 bytes, 64 bytes, and 200 bytes. As predicted, using a 64-byte control channel frame results in the best performance.

variable-sized packet distribution shown in Figure 3.3. As Figure 3.6 shows, with a large control channel frame size (e.g. 200 bytes), performance is seriously degraded because of the amount of overhead incurred when transmitting short packets, which happen to dominate the packet size distribution. Performance is relatively similar for the three short control channel frame sizes, but as expected 64 bytes has the best performance.

3.4 Segmentation and Reassembly On Demand (SAR-OD)

The SAR-OD protocol was developed for *HORNET* to avoid the excessive overhead that can result from segmenting variable-sized packets to fit the transmission frame

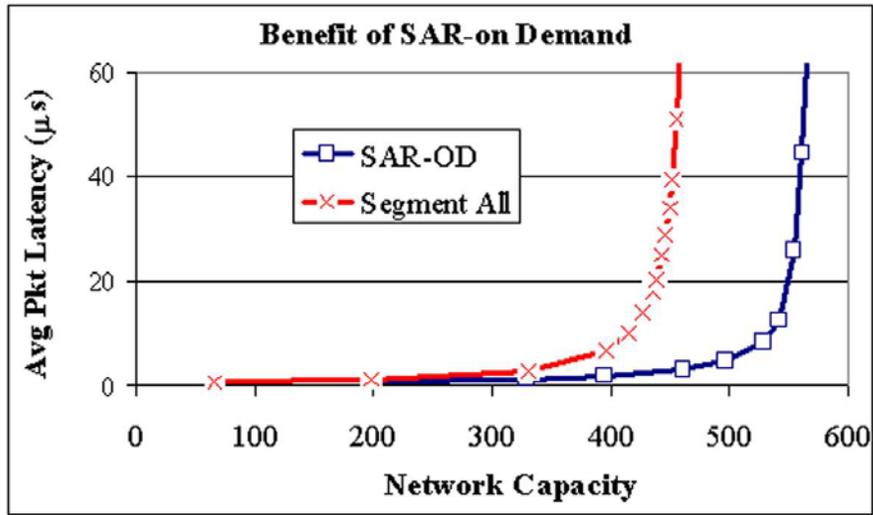


Figure 3.7: This graph shows the advantage of using SAR-OD instead of automatically segmenting all packets into small, fixed-sized cells. The network in the simulation has 33 nodes and 33 wavelengths.

(e.g. IP-over-ATM). However, SAR-OD adds slightly more complexity to the node design than does the alternative. Thus, it is important to measure the performance benefit provided by SAR-OD to determine whether the extra complexity results in a meaningful performance advantage. The performance advantage measured by the simulator is shown in Figure 3.7. The packet size distribution shown in Figure 3.5 (a) is used in this comparison. The graph shows a performance advantage of approximately 15%. Intuitively, this makes sense. The overhead shown in Figure 3.5 (c) is approximately 10.5% for distribution 1. The average overhead for a network that segments all packets can easily be calculated to be more than 25% (16 bytes of overhead in every 64-byte slot, plus unused bytes at the end of the packet). As a result, a performance advantage of at least 15% is expected.

3.5 DQBR Performance Simulations

In Section 2.4.1, the unfairness of the *HORNET* architecture was described. It was suggested that lower throughput occurs for VOQs buffering packets for unfortunate source-destination pairs. The simulator verifies that without fairness control, this result in fact occurs. Figures 3.8 and 3.9 show the throughput in several VOQs in a *HORNET* network when DQBR is *not used*. Figure 3.8 shows VOQ number 18 in each of the 25 nodes on the bi-directional *HORNET* ring, when Wavelength 18 is heavily saturated. VOQ 18 in each node is queuing packets that are destined for Node 18, which receives Wavelength 18. As the figure shows, nodes closer to their destination are unable to transmit to the destination node when the transmission wavelength is saturated.

The simulation that generated the results shown in Figure 3.9 models a network traffic scenario that is very likely to cause unfairness problems in a *HORNET* network. For this plot, Nodes 10 and 11 are sending very heavy amounts of traffic to Node 18, while all other nodes are sending a very light amount. This is saturating the wavelength received by Node 18 (Wavelength 18). The figure shows each node's *throughput* divided by the *load on VOQ 18*. According to the definition of fairness presented in Section 2.4.1, all nodes would have the same ratio of *throughput to load* if the network were fair. However, because of the unfairness of the architecture, the nodes between Nodes 10 and 18 are unable to use Wavelength 18 to send packets to Node 18. Clearly, the simulations verify that there is a need for a fairness control protocol in *HORNET*.

3.5.1 DQBR Measured Fairness Performance

DQBR is modelled in the *HORNET* simulator as follows. The simulator maintains the *RC* and *WC* counters described in Section 2.4.2, as well as the requests propagating

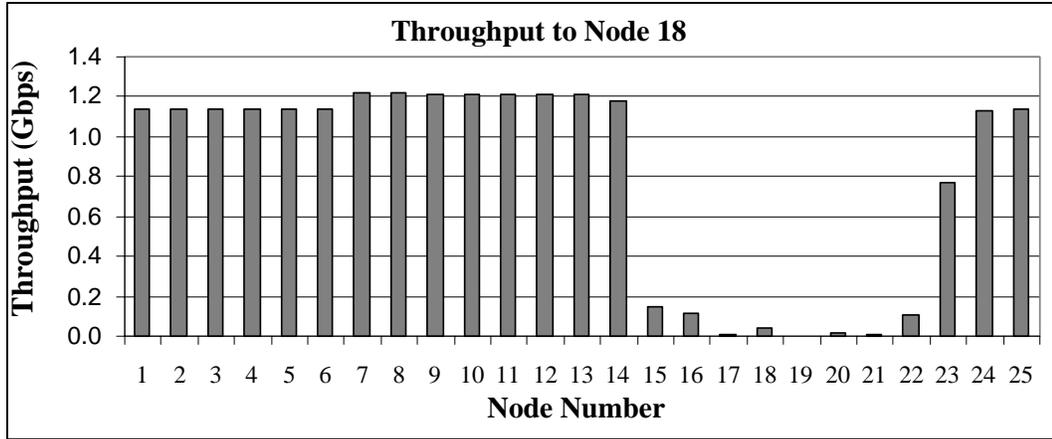


Figure 3.8: Throughput in the nodes' VOQs that use Wavelength 18 for all nodes on the network.

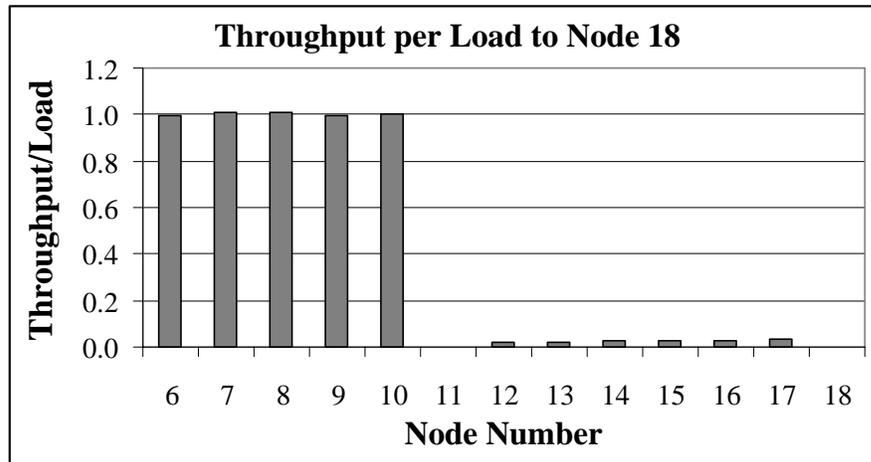


Figure 3.9: Throughput divided by load on the nodes' VOQs that use Wavelength 18. Nodes 10 and 11 are sending a large amount of traffic to Node 18, while the other nodes are only sending light amounts of traffic.

on the control channel. Just as the wavelength availability information circles both directions of the ring in two arrays, the DQBR requests also use circulating arrays of Boolean values. Each column in the rotating array represents a control channel frame, while each row represents the information corresponding to a wavelength. When a packet arrival is generated by the simulator in the VOQ of a node, the node generates the number of requests equal to the packet length measured in control channel frames. Each node monitors the requests on the two rotating control channels and increments/decrements the RC and WC counters as necessary. A node cannot transmit a packet if the corresponding WC counter is nonzero, just as specified by the DQBR fairness control protocol.

To demonstrate the fairness control, the throughput of each node is measured when the network is saturated. To do this, the conditions of the simulation are such that the total network load on the observed wavelength is significantly greater than the capacity of the wavelength. To ensure that the simulations are realistic under such conditions, the *Random Early Detection* (RED) protocol for congestion control [42] is implemented in the simulator because it is expected that a similar protocol would be used in a commercial *HORNET* network. The RED protocol randomly *drops packets* as they arrive in the queue. The probability of dropping a packet increases as the congestion in the queue increases. In reality, the congestion control protocol presented in [43] is preferred because it penalizes users that do not properly respond to the congestion control protocol. It is assumed in this work that all users will behave properly, and thus the RED protocol is used.

Figures 3.10 and 3.11 show that DQBR resolves the unfairness problem in the *HORNET* architecture. Figure 3.10 shows the throughput for nodes sending packets to Node 18 on a 25-node *HORNET* network. With DQBR, the throughput is equal for all nodes, whereas without DQBR, the nodes close to Node 18 have a very difficult time sending packets to Node 18. Also, recall from Section 2.4.2 that DQBR

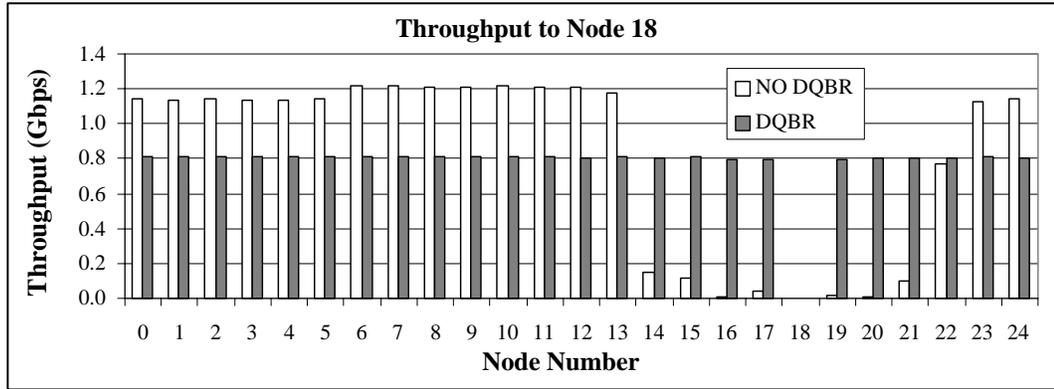


Figure 3.10: Throughput for VOQ number 18 for the 25 nodes on a *HORNET* network. VOQ number 18 corresponds to Wavelength 18, which is received by Node 18. The total network load for Wavelength 18 is 1.5 times its capacity. There is enough propagation delay between nodes to hold 50 control frames.

is designed to eliminate the unfairness condition that occurs in IEEE 802.6 due to propagation distance between nodes [36]. In this simulation, there is enough propagation distance between nodes to hold 50 control frames, yet the throughput is still equal for all nodes when DQBR is used. Thus, it is clear that propagation distance does not affect the fairness of DQBR.

Figure 3.11 shows a simulation with the same unbalanced traffic as in Figure 3.9. In this traffic case, Node 10 has 9.33 Gb/s of traffic arriving to its queue destined for Node 18, Node 11 has 4.67 Gb/s destined for Node 18, and all other nodes have very little traffic. The wavelength can only support 10 Gb/s, so it is heavily oversubscribed. As the figure shows, without DQBR controlling the fairness, the nodes close to Node 18 are unable to transmit packets on Wavelength 18, while in the DQBR network, all nodes have an equal ratio of *throughput to load* for Wavelength 18.

To justify the fairness of this situation, imagine that the simulation results of Figure 3.11 were generated by the following network conditions. There are 250 users of a *HORNET* network. All are sending 58.3 Mb/s of traffic to Node 18. Attached to

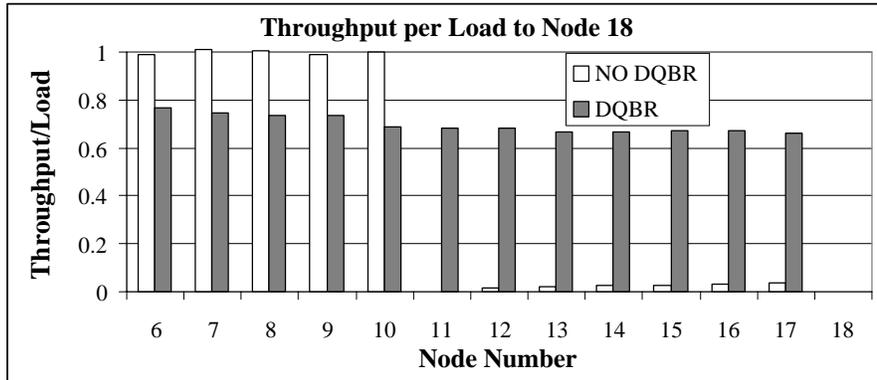


Figure 3.11: *Throughput divided by load* for VOQ number 18 for several nodes. VOQ number 18 corresponds to Wavelength 18, which is received by Node 18. The graph shows that with DQBR, all nodes have the same ratio of *throughput to load*, thus proving that DQBR solves the unfairness problem. In this simulation, the load on VOQ 18 in Node 10 is 9.33 Gb/s, and the load on VOQ 18 in Node 11 is 4.67 Gb/s. All other nodes have only a small load.

Node 10 are 160 of those users, while 80 are accessing the network through Node 11. The other ten users are each using one of the other nodes shown in the plot of Figure 3.11. Under a scheme that equalizes bandwidth to the nodes, such as DQDB’s bandwidth balancing [35, 36], the users attached to Node 10 would be required to reduce their throughput to 29.7 Mb/s each, while all other users continue to transmit at 58.3 Mb/s. This is because Node 10 would be allocated 4.753 Gb/s, allowing Node 11 to transmit at 4.664 Gb/s, and all other nodes to transmit at 58.3 Mb/s. This might be fair if nodes were users, but instead the users of Node 10 are penalized because they happen to be grouped in the same location. In contrast, DQBR allocates each node 40 Mb/s for each user, and thus each node has a *throughput to load ratio* of approximately 0.7, as shown in Figure 3.11.

To verify this result further, the average packet latency and the packet drop probability can be analyzed. The average delay suffered by packets in the VOQs for

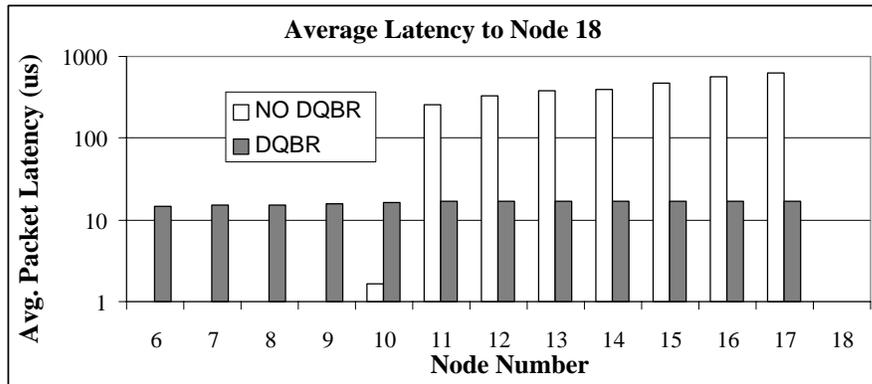


Figure 3.12: Average packet latency in each *HORNET* node for the unbalanced traffic case (Nodes 10 and 11 have a heavy traffic load for Node 18 while all others have light traffic).

Node 18 is plotted in Figure 3.12. The results are generated using the same unbalanced traffic case described above. As the figure shows, with DQBR packets suffer the same latency in all nodes. The packet drop probability at each node is shown in Figure 3.13 for the unbalanced traffic case. As the figure shows, the packet loss probability in a *HORNET* network at all nodes is nearly equal when DQBR is used. Thus, all users of the network will experience the same packet loss probability, and as a result the transport control protocol will regulate the users' load in the same way.

3.5.2 DQBR Performance Penalty

It should be obvious by inspection that the *HORNET* network without fairness control is work-conserving (i.e. if an input has at least one packet for at least one currently available output, then the input will transmit a packet with a probability of 1). If a node has a packet to transmit and there is an opening for the packet, the only event that would prevent the node from sending that packet is if the node sends another packet. Thus, 100% throughput is achievable (when overhead is not considered). However, when DQBR control is applied to the *HORNET* network, it

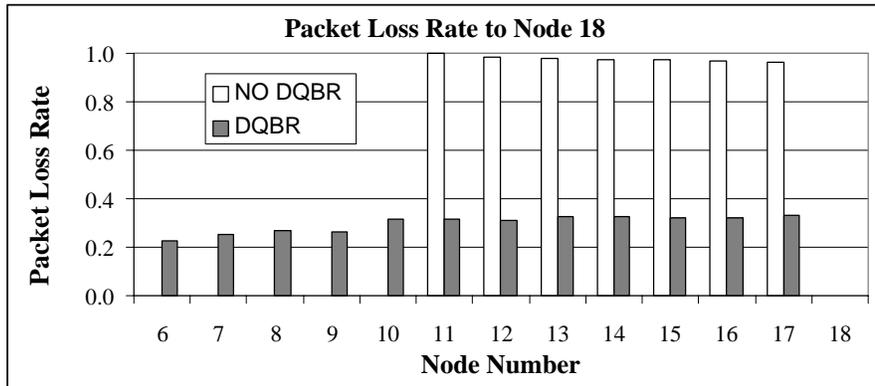


Figure 3.13: Packet loss probability in each *HORNET* node for the unbalanced traffic case (Nodes 10 and 11 have a heavy traffic load for Node 18 while all others have light traffic).

is no longer perfectly work conserving. This is because the DQBR fairness control occasionally forces nodes *not* to transmit any packets, even though there are packets in the queues and available wavelengths to carry those packets. The reason a node does this is because it may be forced to allow an availability on a wavelength to go by for downstream nodes to use. In most cases, these wavelength availabilities would be utilized by nodes downstream. However, there is a finite probability that the downstream node(s) that generated the corresponding request may decide to transmit a packet on a different wavelength, and thus leave the wavelength availability unused. The simulator computed the total throughput for the simulations presented in Figure 3.10. Without DQBR, the measured throughput is 0.999, while with DQBR the measured throughput is 0.965. Thus, the penalty of DQBR is only 3.5%. This is a very minor penalty, considering the tremendous benefit it provides.

3.5.3 DQBR with Variable-Sized Packets

Thus far in this section, the simulations analyzing the performance of *HORNET* with DQBR fairness control have only used fixed-sized packets that are the same size as

the time step (and thus the control channel frame), and that have no overhead. In that case, when a packet arrives, the node attempts to insert one request into the upstream control channel. However, the situation is more complicated when variable-sized packets are transmitted using the SAR-OD protocol. When a packet arrives to the node, the node should place a number of control channel requests equal to the packet's length measured in control channel frames. If the packet is not going to be segmented, then the calculation is as simple as dividing the sum of the packet length and header by the control channel frame size (in bytes). However, in the segmentation and re-assembly protocol, the node must reapply the header each time a packet is segmented, making the total number of bytes transmitted a random variable because the number of packet segmentations is random. The random variable depends on the traffic with which the node must contest, and thus is different for each wavelength as well as time variant.

The ideal solution is for the node to correctly estimate how many times a packet will be segmented to determine the amount of bytes that will be transmitted (payload plus overhead), and to place the necessary amount of requests to carry this amount of bytes. For example, if a node must segment a 560-byte packet five times (i.e. into 6 segments), and the header is 16 bytes, then it will send a total of $560 + (6 \times 16) = 656$ bytes. If the frame size is 64 bytes, then the node should place $\lceil \frac{656}{64} \rceil = 11$ upstream requests, where $\lceil \dots \rceil$ is the *ceiling* operator. If the node had not considered the extra overhead due to segmentation and re-assembly, it would have only placed $\lceil \frac{560+16}{64} \rceil = 9$ upstream requests.

Ultimately, however, determining the correct expected value for the number of times a packet will be segmented is very complex. It depends not only on the upstream traffic rate, but also on the burstiness and self-similarity of the traffic. In practice, this may be very difficult to measure. In this work, it is assumed that the packet segmentation probability is solely dependent upon the upstream traffic rate, which

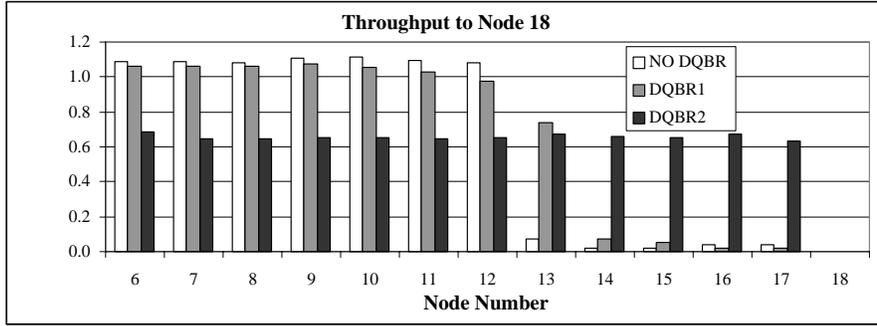


Figure 3.14: Throughput for VOQ number 18 for several nodes for the following cases: no fairness control; DQBR without considering SAR-OD (DQBR1); and DQBR while considering overhead due to SAR-OD (DQBR2). VOQ number 18 corresponds to Wavelength 18, which is received by Node 18.

can easily be measured in practice by monitoring the control channel. To determine the number of slots to request for a packet, the node uses the following expression:

$$Rq = \lceil \frac{PB}{(CCF - R_u \times HB)} \rceil$$

where Rq is the number of requests, PB is the number of payload bytes transmitted, HB is the number of bytes in the *HORNET* header, CCF is the control channel frame length, R_u is the upstream traffic rate (normalized to 1), and $\lceil \dots \rceil$ is the *ceiling* operator. The result of using this expression is shown in Figure 3.14. This figure shows the throughput in the VOQs on Wavelength 18 when no fairness control is used, when DQBR *without* considering segmentation is used (DQBR1 in legend), and when DQBR while considering segmentation is used (DQBR2 in legend). As the figure shows, if the extra overhead that occurs from packet segmentation is not considered by the DQBR protocol, fairness control does not work properly.

3.6 *HORNET* versus *RPR-over-WDM*

During the description of the *HORNET* architecture in Section 2.1, it was predicted by intuition that a *HORNET* network could deliver the same performance as an *RPR-over-WDM* network with much lower equipment cost. However, the conclusion drawn in that section is based on several simplifications, and thus should be investigated more thoroughly with simulations. In this section, a comparison between *HORNET* and *RPR-over-WDM* is presented using simulations.

3.6.1 *RPR-Over-WDM* Simulator

The *RPR-over-WDM* simulator created for this work is very similar to the simulator used for *HORNET* described in the previous sections. The major difference between the networks, and hence the simulators, is that in *RPR-over-WDM* all wavelengths are terminated in every node (i.e. packets are not wavelength routed). A node can insert packets on any wavelength and remove them from any wavelength. This means that each node has W transmitters and W receivers, where W is the number of wavelengths.

As explained in the available documentation on RPR [15], each node queues packets that arrive from upstream and are destined for a downstream node, as well as packets that are generated by local users. The queue that buffers the locally generated packets is called the *transmit* queue while the queues that buffer packets traversing the ring are called *transit* queues. In the simulations in this work, each node has one *transmit* queue and W *transit* queues for each transmission direction. Packets received by a node on wavelength w that are destined for a downstream node are stored in transit queue w . Packets in transit queue w will be retransmitted on wavelength w when they reach the front of the queue. The *transit* queues are given priority over the *transmit* queue in this work (the actual fairness protocol is outlined in the currently

evolving RPR proposals). If *transit* queue w is empty, then the *transmit* queue can transmit a packet on wavelength w . Since there are W transmitters, it is assumed that it is possible for a node to transmit $W - n$ packets from its *transmit* queue simultaneously, where n is the number of *transit* queues with packets. In reality, this may be a generous assumption for *RPR-over-WDM* because an implementation of this may be difficult.

To make an accurate comparison, it is imperative for the packet arrival process to be exactly the same in the *RPR-over-WDM* simulations as it is in the *HORNET* simulations. Thus, even though the transmit queue in each node is not a collection of destination-based VOQs, the *RPR-over-WDM* simulator nonetheless iterates over the set of destinations when generating packets. Also, the time step duration is smaller in the *RPR-over-WDM* simulations, but the packet arrivals occur at the same time interval (not time-step interval) as in the *HORNET* simulations (i.e. one *HORNET* control channel frame duration). The time step is smaller in the *RPR-over-WDM* simulator because the simulator does not use a framing protocol on the links. This is advantageous for the *RPR-over-WDM* results because it gives better granularity.

Just as in the *HORNET* simulations, it is assumed that a certain amount of additional overhead is necessary for RPR to transmit each packet. Although commercial implementations may differ, for these simulations it is assumed that 24 bytes of overhead are necessary for every packet. This overhead is used to identify and delineate packets inside the transport framing protocol, as well as source/destination information, and a CRC. It is likely that a small amount of additional overhead would result from the framing protocol, such as the 3% overhead in SONET framing, but that overhead is ignored because it is relatively small, and because in the future, improved framing protocols may lower it even further.

To compare the performance with the *HORNET* simulations, the average packet latency in each node's transmit queue is determined. Delays incurred in the transit

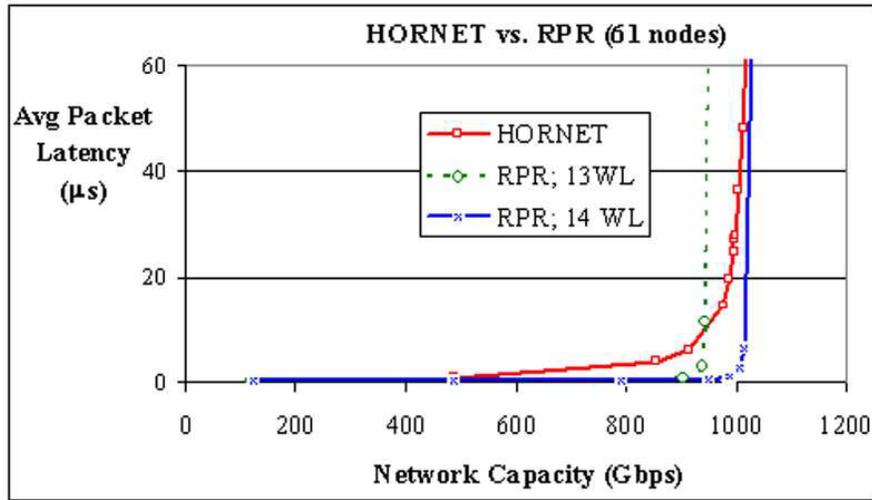


Figure 3.15: Simulated performance of *HORNET* and *RPR-over-WDM* on a 61-node bi-directional ring network. *RPR-over-WDM* is simulated with both 13 and 14 wavelengths.

queue are not taken into account because they will be insignificant, since the transit queues take priority over the transmit queue in all cases.

3.6.2 Simulation Results

The *HORNET* simulator was used to determine what size (i.e. number of wavelengths, number of nodes) a *HORNET* network would need to be to exceed 1 Tb/s capacity. As Figure 3.15 shows, a *HORNET* network of 61 nodes and 61 wavelengths crosses the 1 Tb/s capacity mark. Each node contains two tunable transmitters and two fixed-wavelength receivers (i.e. one for each direction). For this simulation, overhead is included, variable-sized packets are used, and the DQBR fairness control protocol is applied.

The *RPR-over-WDM* simulator was then used to determine *how many wavelengths* are necessary in an *RPR-over-WDM* network of the same number of nodes to exceed

1 Tb/s capacity. As Figure 3.15 shows, 14 wavelengths in each of the two fiber rings are required for a capacity of 1 Tb/s. Thus, each of the 61 nodes must have 28 photonic receivers, 28 photonic transmitters, and 28 line cards (all are 10 Gb/s components). Also, each *RPR-over-WDM* node contains the switching capacity to handle the 280 Gb/s coming into and going out of the node. Clearly, each *HORNET* node is much less expensive than each *RPR-over-WDM* node at this high capacity.

3.6.3 Equipment per Node Comparison

The simulation results of Figure 3.15 show that for a 61-node network, *HORNET* and *RPR-over-WDM* can both deliver 1 Tb/s capacity, but *HORNET* can do it with much less equipment. In reality, however, there are several external factors that contribute to the determination of the number of nodes on a network. The network operator makes the determination based on the cost of operating a node, the subscriber density in certain regions, the cost of connecting subscribers to nodes, network traffic patterns, and the desire for reconfigurability, as explained in Section 2.8.1. Thus, it is conceivable that a network provider would want to deliver 1 Tb/s capacity on a network with fewer than 61 nodes. To reduce the number of nodes in a *HORNET* network, nodes are merged together such that they have greater than one transmitter and receiver for each direction. The total capacity is not reduced because in reality, the capacity delivered by a *HORNET* network is not based on the total number of nodes, but rather it is based on the number of transmitters in the network and the number of wavelengths carried by the fiber.

As nodes are merged together, the equipment within each *HORNET* node increases quickly, as shown in Figure 3.16. In contrast, when *RPR-over-WDM* nodes combine, the equipment necessary per node does not increase as dramatically. In *RPR-over-WDM*, a node is transmitting traffic generated locally and traffic that is passing through the node from upstream, as described in Section 1.5. When one node

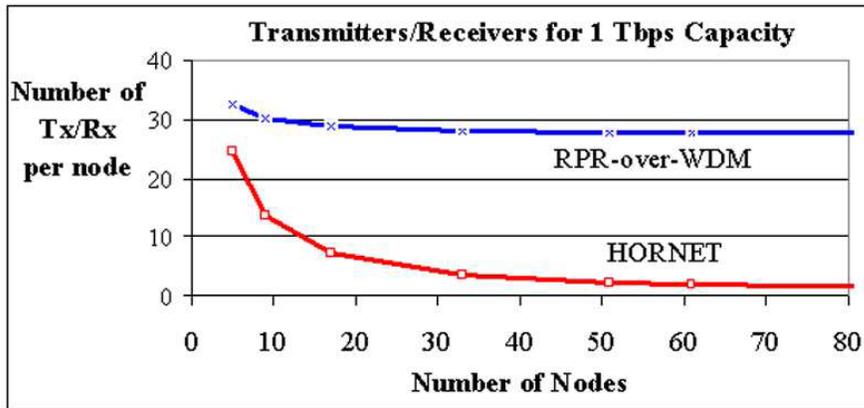


Figure 3.16: Comparison of the number of transmitters and receivers in each node for *HORNET* and *RPR-over-WDM* for varying number of network nodes.

is merged with its upstream neighbor, the traffic transmitted by the newly created node is not the sum of the traffic transmitted by the two daughter nodes. This is because much of the traffic transmitted by the downstream node is traffic that is also transmitted by the upstream node.

In contrast, when two nodes are combined in *HORNET*, the traffic generated by the new node is the sum of the traffic generated by the two daughter nodes. Therefore, the necessary amount of equipment within the new node doubles to match the total performance of the two daughter nodes. This is summarized in Figure 3.16. As the figure shows, as the number of nodes on the ring decreases, the significance of the *HORNET* advantage decreases. Nonetheless, networks even as small as 20 nodes still heavily favor a *HORNET* architecture, as the number of transmitters and receivers is less than $\frac{1}{4}$ of the quantity of that in *RPR-over-WDM*.

3.7 Summary

The *HORNET* simulator was constructed to quantify the performance of the protocols, such as the control-channel-based MAC protocol (including SAR-OD) and the DQBR fairness protocol. Also, the simulator was constructed to quantify the equipment reduction of *HORNET* compared with *RPR-over-WDM*. The simulator was designed with the intelligence to correctly model variable-sized IP packets. Using the simulator, it was determined that the optimal control channel frame size for the *HORNET* MAC protocol is 64 bytes, based on currently available data on IP packet size distributions. Also, the simulator proved that the SAR-OD protocol has better than a 15% performance advantage over a protocol that would divide variable-sized packets into fixed-sized frames.

The DQBR fairness protocol was thoroughly explored using the simulator. It was proven that DQBR can deliver equal opportunity to all users to access any wavelength in the network, regardless of their location. All users experience the same packet latency on a particular wavelength, no matter if they are located at the downstream end of the wavelength, or if they are accessing the network through a node that is heavily using the wavelength. Additionally, fairness is delivered to the users without sacrificing performance. The network only experiences a 3.5% degradation in performance due to the use of DQBR.

Finally, the simulator was used to compare the necessary equipment in a *HORNET* network and an *RPR-over-WDM* network when both are delivering 1 Tb/s capacity. It was found that in a simplified case, the *HORNET* network only requires two transmitters, two receivers, and two *HORNET* line cards in each node while the *RPR-over-WDM* network requires 28 transmitters, 28 receivers, and 28 *RPR-over-WDM* line cards per node. The more complicated scenario of varying the number of nodes on the two networks was also explored. The simulator demonstrated that

reducing the number of nodes without reducing the overall capacity of the network reduces the advantage of *HORNET* as compared to *RPR-over-WDM*. Nonetheless, for any reasonable number of nodes on the network, *HORNET* still requires *significantly less equipment* than *RPR-over-WDM*. Naturally, this results in a much lower infrastructure cost for a *HORNET* network.

Chapter 4

HORNET Subsystems

4.1 Introduction

As discussed in Section 2.2, the commercial deployment of a *HORNET* network requires the development of three subsystems that are not in widespread use today. The three subsystems are a fast-tunable packet transmitter, an asynchronous packet receiver, and a linear optical amplifier. Fortunately, a significant amount of research has been conducted on all three subsystems in recent years. This section reviews the research conducted by other institutions as well as the research performed for the *HORNET* project for each of the three subsystems.

4.2 Fast-Tunable Packet Transmitter

The transmitter in a *HORNET* node sends each packet on the *wavelength* that is *received* by the packet's *destination node*. Thus, the transmitter must have the ability to *tune* its output wavelength. The requirements on the tunable transmitter are critical. First, since the transmitter sends packets on every wavelength in the network, the tunable laser must have the tuning range to cover the entire network transmission

spectrum, as well as the resolution to target each wavelength within the band. Second, since the transmitter may be forced to tune its output wavelength between nearly every packet transmission, the laser must have the *agility* to tune between wavelengths very quickly. The node cannot transmit while the laser is tuning, and thus the tuning duration is overhead. Clearly, it is desirable to keep the tuning time as small as possible to keep the overhead low. In fact, the payload of a 100-byte packet at 10 Gb/s will be 80 ns in duration. If the tuning time of the laser is even as low as 20 ns, it contributes *20% to the overhead* of the packet (100 ns duration; 20 ns of overhead, 80 ns of payload). Therefore, the design of the tunable packet transmitter for *HORNET* must consider the *tuning range, tuning precision, and tuning speed*.

As Figure 4.1 shows, the tunable packet transmitter consists of three important components. The *tunable laser* is controlled by the *laser-tuning controller*, and the *data modulator* writes the bits onto the optical output of the laser. The data modulator is a conventional commercial component and is therefore not discussed in this section. The tunable laser is a relatively new product on the commercial market, and the laser-tuning controller is a custom component that has only been developed for research applications. This section focuses on the tunable laser because it is an interesting next-generation component, and on the laser-tuning controller because it is a novel subsystem that must be developed for a commercial deployment of *HORNET*.

4.2.1 Tunable Semiconductor Laser

After a couple of decades of research, semiconductor tunable lasers have somewhat recently become commercially available. The discussion in this report is limited to the tunable lasers that are used for transmission purposes, as opposed to those used for test and measurement purposes. There are two classes of commercially available tunable lasers. Both types of lasers are marketed on the basis of offering flexibility to network operators, and not as tunable packet transmitters.

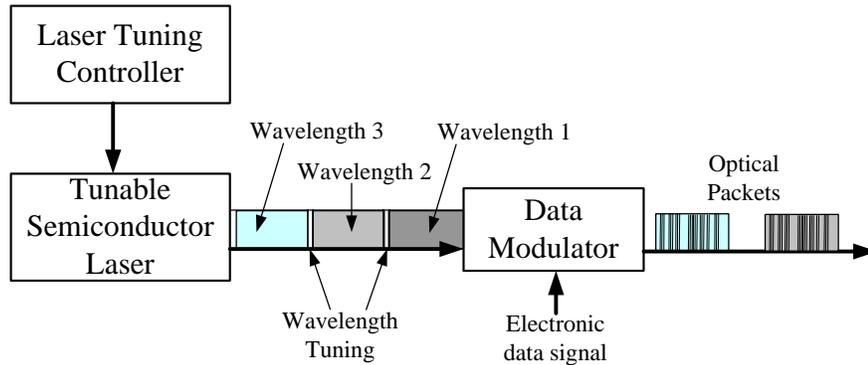


Figure 4.1: The tunable packet transmitter subsystem contains three components: the tunable laser, the laser-tuning controller, and the data modulator.

The first type of tunable laser that is available today is a MEMS laser [44, 45]. As described in the references, the laser is tuned by mechanically changing the length of the laser cavity. This is typically done by applying a voltage to a suspended film that is acting as a reflector. The applied voltage causes the film to flex, making the cavity either shorter or longer, depending on the voltage. Varying the length of the cavity changes the value of the wavelength mode that it selects. Although these MEMS tunable lasers are fine products for certain applications, they will not perform well as tunable packet transmitters. A typical tuning duration on the order of only a few nanoseconds is critical for the practicality of the tunable transmitter in *HORNET*, and thus a mechanically tuned laser will not suffice.

Fortunately, the second class of commercially available tunable lasers is more amenable to fast-tuning. *Distributed Bragg Reflector* (DBR) lasers are semiconductor lasers that are opto-electronically tuned by an injected current. The injection of carriers modifies the optical properties of the laser cavity, thereby changing the output wavelength. There are three types of DBR lasers that have been developed: the *Super-Structure Grating* DBR (SSG-DBR) laser [46, 47], the *Sampled Grating*

DBR (SG-DBR) laser [48, 49], and the *Grating-Assisted Coupler with Sampled Reflector* (GCSR) laser [50]. Though all three types of DBR lasers are interesting, this report will concentrate on the SG-DBR laser because it appears to be the most practical structure. However, other structures may emerge in the future, as the tunable semiconductor laser is still a popular research topic [46, 51, 52, 53, 54].

The design concept of a DBR laser is relatively simple. One of the reflectors of the laser cavity is a Bragg grating. The Bragg grating selects a single spectral mode for the output wavelength of the laser. Injecting carriers into the Bragg grating modifies the optical properties of the grating. When the optical properties are modified, a different spectral mode is selected, and as a result the laser emits on a different wavelength. The primary challenge in designing a DBR laser for use in a network like *HORNET* is designing the grating in a way to enable the laser to be tuned over a very wide range (e.g. across at least 40 nm) without sacrificing the precision to tune between closely spaced wavelengths (i.e. 0.8 nm or better).

The SSG-DBR, GCSR, and SG-DBR lasers all have their own unique approach to solving this design problem. The structure of the SG-DBR laser, made available by *Agility Communications*, is depicted in Figure 4.2 [55]. The laser cavity is bound by sampled gratings that act as reflectors. To understand the advantage of using a sampled grating, consider the effect in the frequency domain when an analog signal is sampled. The result of the sampling is multiple copies of the analog signal's frequency components that are periodically spaced. The same effect occurs when using a sampled grating [48]. Instead of selecting one mode, the sampled grating selects a *comb* of periodically spaced modes [49], as depicted in Figure 4.2. If each of the two sampled gratings is designed with a different spacing between the gratings, then the combs of characteristic modes for each of the two gratings will have different periods.

The output wavelength is set by the modes with the strongest overlap between the two combs that are within the gain bandwidth of the laser. Since the combs have

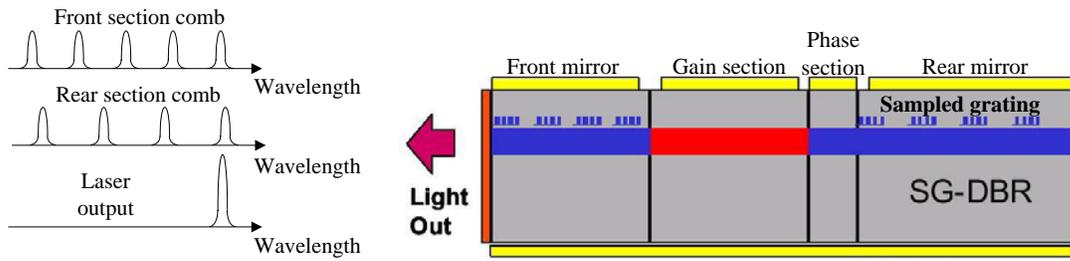


Figure 4.2: The Sampled-Grating DBR laser.

different periods, the laser tuning occurs through the *Vernier Effect* [49]. This means that injecting only a small amount of carriers into one of the grating sections and thus slightly altering the optical properties of the grating tunes the laser by a significant amount. As a result, the laser can be tuned over a very wide range with only a modest current injection, and can also be tuned precisely between closely spaced wavelengths. Additionally, since the tuning is based on carrier injection, the output wavelength can change quickly, in contrast to the MEMS lasers.

4.2.2 Laser-Tuning Controller

The function of the laser-tuning controller is to inject the proper currents into the tunable laser to achieve the desired output wavelength. Although the three DBR laser structures mentioned above are different, the operation of each is quite similar. Each takes four injection currents, where two of the currents tune the laser's spectral modes, and the third current (the phase section current) is used to slightly adjust the length of the cavity. The fourth current (the gain current) is used to provide gain to the laser cavity, although its magnitude also has a noticeable effect on the output wavelength.

Since the laser output wavelength is dependent upon the values of the injected currents, the tuning process is considered to be analog. However, when the *packet*

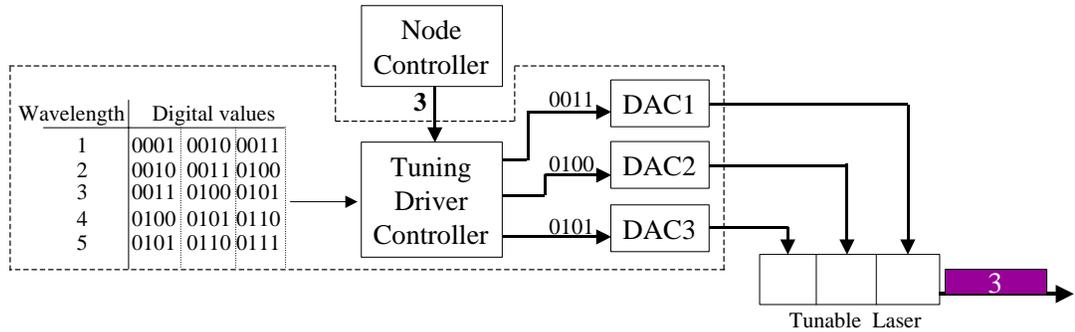


Figure 4.3: The laser-tuning controller for the fast-tunable packet transmitter used in *HORNET*.

switch selects an output wavelength for a packet, it is incapable of generating the analog currents for the laser. Thus, the chief component in the laser-tuning controller is a digital-to-analog converter (DAC). Figure 4.3 illustrates the design of the laser-tuning controller developed for *HORNET*, which was first shown by Shrikhande, et. al. in [56].

After determining the output wavelength for the packet, the packet switch sends the wavelength number to the laser-tuning controller. Within the controller, a lookup table is used to convert the wavelength number into three digital words. Each digital word corresponds to an analog current value for one of the three laser sections. The digital words are placed at the inputs of the three DACs. The resulting output currents tune the laser to the desired wavelength. Experimental results generated using this controller design with a GCSR laser are reported in [56]. Very similar results have recently been generated in the *HORNET* project using the SG-DBR laser.

Over the last few years other research institutions have also investigated fast-tunable laser subsystems. The research described in [9] aims to develop an all-optical packet switch for next-generation backbone optical networks. The tunable laser is used as a component in a *tunable wavelength converter* subsystem. Research focusing

on the fast-tunable laser component for the project is summarized in [57]. As described in the paper, a digital laser-tuning controller design similar to that described above is used to tune the laser. However, in [57] the controller and the GCSR tunable laser are packaged together, which tremendously improves the performance. In fact the results presented in that work feature tuning times of less than 5 ns from one particular wavelength to any of 36 other standard wavelengths within the conventional transmission band. Another project, described in [11] also uses fast-tunable lasers for tunable wavelength converters in all-optical packet switches.

Another similar project [10] more thoroughly investigates tuning performance of a GCSR laser for the entire conventional transmission band. In that work, every possible tuning combination of standard wavelengths is characterized. The results are less optimistic than those reported in [57]. In the work of [10], only about one-half of the wavelength pairs resulted in less than 20 ns tuning time. Almost all are less than 30 ns. However, the laser and controller are not integrated, and details about the controller design are not provided. Thus, a primary reason for the more pessimistic results may be the controller design, and not the inherent tuning properties of the laser.

It is clear from the results obtained in the *HORNET* project and the other projects described above that the technology for a fast-tunable packet transmitter exists, but has not quite matured yet. Nonetheless, when *HORNET* is at the stage of commercial deployment, one can be almost certain that commercial fast-tunable transmitter subsystems will exist. In addition, since *tunable* semiconductor lasers today are *less than* twice as expensive as conventional semiconductor lasers, the subsystem will surely be priced competitively when the product is in demand.

4.3 Asynchronous Packet Receiver

The asynchronous nature of the *fast-tunable packet transmitter* brings about the need for an *asynchronous packet receiver* in *HORNET*. Consider the packets on a wavelength that will be optically dropped by a node and received by the node's receiver. Consecutive packets in the link are likely to have been transmitted by different source nodes. One packet in the link is transmitted by a particular node n , while the packet that follows it is likely to have been transmitted by a different node m . Both packets begin in alignment with the control channel SOF indicator. However, although the control frame synchronization protocol described in Section 2.7 enables the packets to be well aligned with the control frame, the alignment is certainly not good enough to maintain perfectly synchronous bit alignment. Thus, the packet sent by Node m has random bit-phase as compared to the packet transmitted by Node n , which precedes it. As a result, the node receiving these two packets must have a receiver designed to *asynchronously* receive packets. In addition to the problem of the asynchronous relationship between consecutive packets, the exact baud rate of each of the two consecutive packets may be slightly different (often within 0.001%) because they were transmitted by two different nodes.

To receive asynchronously arriving packets, the receiver must perform bit-level-synchronization on each arriving packet. Just as tuning time is overhead, the time required to achieve bit-synchronization is overhead because payload data cannot be properly received during those moments. Therefore, the asynchronous packet receiver for *HORNET* must be designed such that the bit phase and frequency are acquired in very little time, preferably in only a few bytes.

Currently, no commercial product that *asynchronously* receives high-data-rate packets is available. This is not surprising because no such commercial market exists. As a result, it is necessary to investigate the asynchronous packet receiver subsystem

to verify that such a product will exist when necessary. This section discusses the research performed for the *HORNET* project as well as research conducted at other institutions, all of which is aimed at the development of the *asynchronous packet receiver subsystem*.

4.3.1 *HORNET* Research on Asynchronous Packet Receivers

Two classes of potential solutions for an asynchronous packet receiver exist: analog and digital. Digital solutions intuitively appear to be the better choice for a product, but analog solutions are much easier to implement, and are thus well suited for projects that lack the ability to produce high-speed digital integrated circuits, such as the *HORNET* project. Two analog techniques were used for experiments in the *HORNET* project. It is not expected that either of the two techniques will evolve into a commercial product for receiving packets asynchronously, but the techniques are very useful because *HORNET* experiments cannot be performed without a means of receiving packets asynchronously. Also, developing and utilizing the techniques provides valuable insight into the asynchronous packet receiver subsystem.

The first technique used in *HORNET*, which is called the *Embedded Clock Tone* (ECT) technique, was first reported in [58]. In the ECT method, the transmitter frequency multiplexes its local clock with the payload of the data packet. In the packet receiver, the packet and embedded clock are separated using a low-pass filter for the data and a very narrow band-pass filter for the clock tone. The relationship between the clock phase and the data bit-phase is known because it is a design aspect of the transmitter. Thus, the receiver can easily be designed to use the received clock to recover the payload data.

The advantage of the ECT technique is its simplicity. However, the disadvantages of the technique are substantial. First, consider the design of the transmitter. The clock tone and payload data are frequency multiplexed just before the electronic

signal modulates the optical data modulator. The amplitude of the resulting signal is limited by the maximum output of the modulator-driver amplifier and by the modulation depth of the modulator. Generally, the entire amplitude of the signal is used for the payload data. However, with the ECT technique, the clock takes some of the modulation depth away from the data signal, thus reducing power in the transmitted payload data signal.

The second negative aspect of the ECT technique is the large overhead due to clock recovery time. A narrow band-pass filter separates the clock tone from the received signal. It is an inherent fact that a narrow band filter has a slow response time because the response time is proportional to the inverse of the bandwidth of the pass band. The rise time of the output signal measured in [58] is approximately 16 ns, which equates to 20 bytes at 10 Gb/s. The recovery time cannot be improved by better circuit design or technology because the relationship between the bandwidth and the response time is a fundamental fact.

A second technique was developed for the *HORNET* experiments in order to avoid the modulation depth penalty of the ECT. This technique is referred to as *nonlinear clock extraction* because it uses a nonlinear circuit element to re-create the clock signal from the incoming data signal. The nonlinear clock extraction subsystem is illustrated in Figure 4.4. A strong candidate for the nonlinear element is a frequency doubler, which typically contains a rectifier sandwiched by input and output band-pass filters. For each transition in the payload data stream, the rectifier produces one cycle of the re-created clock signal. A very narrow band-pass filter is required at the output of the nonlinear clock element. The filter rejects unwanted signal components and acts to average the clock cycles generated by the doubler. An averaging element is necessary because the doubler only produces clock cycles at the locations of bit transitions. A narrow band-pass filter converts bursts of the clock tone into a continuous clock tone through its ability to average. The filter must be narrow enough to withstand strings

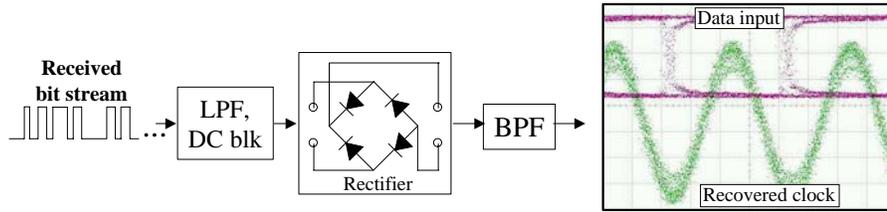


Figure 4.4: The design of the nonlinear clock extraction technique, which re-creates the perfectly synchronized clock tone from the incoming data.

of bits in the payload bit stream with no transitions. Once again, this is dictated by the well-known relationship between filter response time and filter bandwidth.

The resulting output clock tone phase has a deterministic relationship to the payload data bit phase, and thus the clock can be used to recover the data bits in the receiver. However, the nonlinear clock extraction design suffers from the same recovery overhead penalty as the ECT technique because a narrow band-pass filter is necessary. Thus, the method is an excellent choice for performing *HORNET* experiments, but it is not a likely candidate for a commercial product.

4.3.2 Research on Digital Asynchronous Packet Receivers

Other research institutions with the necessary expertise and development platforms have investigated solutions for a *digital* asynchronous packet receiver. A wide range of designs for a wide range of requirements has been developed. For example, a receiver that synchronizes to 622 Mb/s data in 1.3 clock cycles is described in [59]. Also, the SiGe asynchronous packet receiver discussed in [10] can recover 40 Gb/s data with only 5 ns of overhead. However, the most interesting result from the point of view of *HORNET* is presented in [60].

The authors of [60] designed and successfully implemented an asynchronous packet receiver that recovered 4 Gb/s data with no synchronization overhead. The circuit is implemented in 0.5 μm CMOS, so in principle the design can be upgraded to 10 Gb/s

by using an improved CMOS technology, such as 0.18 μm . The receiver uses an over-sampling technique to perform the data recovery as follows. The circuit generates 24 phase-shifted oscillators at a frequency of $\frac{R_b}{8}$, where R_b is the bit rate of the received data stream. The 24 oscillators are used to sample 8 bits at a time, and thus the bit stream is *over-sampled* by a *factor of three*. After every 24 samples, the chip uses an algorithm to decode the 8 bits it received during that time.

The only disadvantage to the design appears to be the fact that transitions are required in each 8-bit segment analyzed by the algorithm. This means that a coding technique must be used to ensure that there are no strings of 8 bits without transitions. Simple coding techniques such as 8B/10B are sufficient, but a lot of overhead is incurred from such a technique. Research is still under way to improve the design of the receiver, and thus in the near future the receiver may not suffer from this requirement.

It is safe to conclude from the research results presented in this section that the asynchronous packet receiver can and will exist as a commercial product when *HORNET* is ready for it. The exact design of the receiver is uncertain, but it is clear that the technology exists today to produce such a product. As soon as *HORNET*, or a similar networking system creates the demand for the device, it will be produced.

4.4 Linear Optical Amplifier

The need for an asynchronous packet receiver in *HORNET* is an example of the difficulties brought about by the unique aspects of the network architecture. A similar problem arises with the use of conventional optical amplifiers in *HORNET*. Consider a control channel frame as it circumnavigates the ring on the asynchronous WDM link in a *HORNET* network. As the frame passes through a particular node, the node removes all of the power on the drop wavelength(s). Now consider the power on the

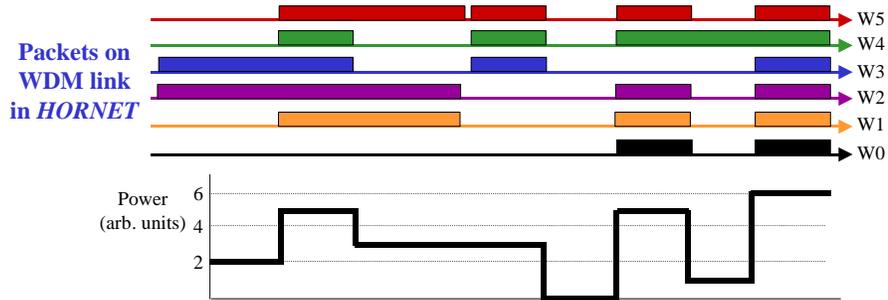


Figure 4.5: The total optical power at any location in the asynchronous WDM link in *HORNET* is random. $W0 = \text{Wavelength } 0$.

drop-wavelength between the SOF indicators of the control channel frame. Immediately after passing through the node, the power is zero. The power within the frame on the wavelength remains zero until a node somewhere on the network transmits a packet on the wavelength as the control channel frame passes through the transmitting node. In general, this occurs at a random location on the ring. As a result, the power on the wavelength at any time at any location (except immediately after the drop-node) is random. Thus, the total power on the WDM link in a *HORNET* network is random at all locations on the ring. Figure 4.5 depicts the randomness of the power on the link.

Conventional networks easily avoid this situation by using scrambling and idle packets. Scrambling is a technique in which the data sequence is multiplied by a binary representation of a polynomial in order to randomize the bits in the sequence. The data stream is unscrambled using the same polynomial in the receiver. The scrambling technique eliminates long strings of binary *zeros* that occur when no data is to be transmitted. Idle packets can also be inserted when there is no data to be transmitted. These are packets that have a good balance of binary *ones* and *zeros*, but that contain no user data.

Obviously, these techniques cannot be used in a *HORNET* network. Consider the

circulating control frame again. After the frame passes through a particular node, the node removes all of the power on the drop wavelength(s). The node could then insert idle packets to keep the total power constant. However, if the node inserts an idle packet, no other node can utilize this time period on this wavelength to transmit a packet. It is essential for the drop-node to leave the wavelength empty so that another node on the ring can transmit a packet. It is noteworthy that if a fast-tunable optical band-stop filter existed, a node could use it to erase an idle packet and insert a user packet. However, this technology is not on the horizon, so it is not yet considered to be a viable solution.

The randomized power on the WDM link is detrimental because conventional EDFAs used in today's optical networks do not function properly under such input conditions. Recently, however, three options have emerged as potential solutions for the EDFA dynamics problem in *HORNET*. None are mature technologies at the time of this report, but as this section will show, the problem of optical amplifier dynamics in *HORNET* has a solution, and as a result is not a technological roadblock.

4.4.1 EDFA Dynamics

The design of a generic EDFA is shown in Figure 4.6 (a). The Erbium-doped fiber (EDF) acts as the gain medium by absorbing power from the optical pump and transferring it to the optical signal. The EDF is a three-level energy system, as shown in Figure 4.6 (b). The Erbium ions absorb light at 980 nm and also at 1480 nm and thus move to a higher energy level. Ions at the higher energy levels 3a and 3b quickly decay into energy level 2. This second level is at a wavelength range of approximately between 1530 nm and 1565 nm for conventional EDFAs [2]. Ions in energy level 2 decay to energy level 1 either through spontaneous emission or through stimulated emission. When the signal of a wavelength within the gain bandwidth region passes through the EDFA, stimulated emission occurs, thus providing gain to the signal.

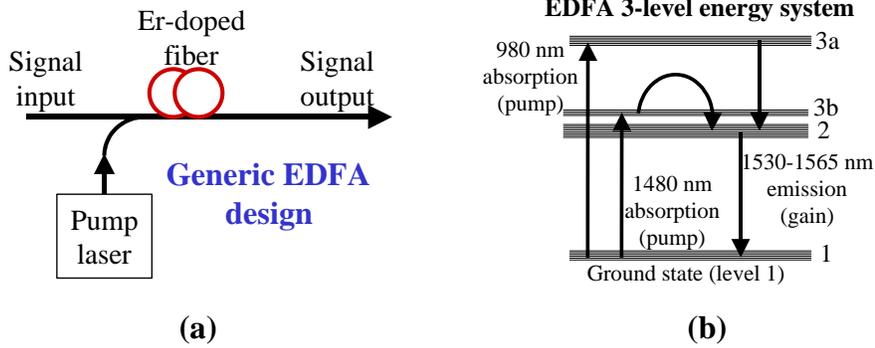


Figure 4.6: (a) Design of a typical EDFA. (b) 3-level energy structure of an EDFA.

The magnitude of the signal gain is dependent upon the probability of a stimulated emission event. Naturally, such an event is dependent upon the amount of ions energized to the second energy level, and thus the gain is also dependent upon this. An important parameter to consider is the *fraction* of ions energized to level 2. This parameter is called *inversion* and is represented as n_2 . The effect of inversion on the amplifier gain is described well by [2]. Only ions in energy levels 1 and 2 are considered because the time constant of the decay from levels 3a and 3b is significantly smaller than the time constant of the decay from level 2 to level 1 (τ_2). Therefore, the 3-level system can actually be approximated as a 2-level system [23].

To analyze how random power on the *HORNET* WDM link affects the EDFA gain, the dynamics of the inversion should be considered. The rate equation for the inversion n_2 is used for this analysis [23]:

$$\frac{\partial n_2(z, t)}{\partial t} = -\frac{n_2(z, t)}{\tau_2} - \frac{1}{\rho A} \sum_{i=1}^W u_i \frac{\partial P_i(z, t)}{\partial z} \quad (4.1)$$

where ρ is the density of the active Erbium ions, A is the fiber core cross-section, u_i is a unit vector indicating the direction of propagation of the i th wavelength, W is the number of wavelengths (including the optical pump), and P_i is the power of the i th wavelength in the WDM spectrum (including the pump). P_i is normalized to the

photon energy, and is thus expressed in units of photons per unit time.

The photon propagation equation for the i th channel is

$$\frac{\partial P_i(z, t)}{\partial z} = u_i[(\gamma_i + \alpha_i)n_2(z, t) - \alpha_i]P_i(z, t). \quad (4.2)$$

Integrating Equation 4.1 over the length of the Erbium fiber results in

$$\left(\frac{d}{dt} + \frac{1}{\tau_2}\right)\bar{n}_2(t) = -\frac{1}{\rho A \ell} \sum_{i=1}^W P_i^{in}(t) \{ \exp[\bar{g}_i(t)\ell] - 1 \} \quad (4.3)$$

where $\bar{n}_2(t)$ is the average inversion level across the fiber and ℓ is the length of the fiber. The term $\exp[\bar{g}_i(t)\ell]$ is the linear gain of the amplifier. As shown in [23], the term $\bar{g}_i(t)$ is dependent upon the inversion as

$$\bar{g}_i(t) = (\gamma_i + \alpha_i)\bar{n}_2(t) - \alpha_i \quad (4.4)$$

where γ_i is the emission constant for Wavelength i and α_i is the absorption constant for Wavelength i .

Equation 4.3 can be rewritten in a more informative expression. It is desirable for this analysis to separate the pump power from the signal power. When that action is taken, the equation can be rewritten as

$$-\frac{1}{\rho A \ell} P_{pump}^{in}(t) \{ \exp[\bar{g}_{pump}(t)\ell] - 1 \} = \frac{d\bar{n}_2(t)}{dt} + \frac{\bar{n}_2(t)}{\tau_2} + \frac{1}{\rho A \ell} \sum_{i=1}^W P_{sig(i)}^{in}(t) \{ \exp[\bar{g}_{sig(i)}(t)\ell] - 1 \}. \quad (4.5)$$

The term on the left side of the equation represents the absorption of the optical pump by the EDF, while the term on the far right represents the gain of the WDM signal wavelengths due to stimulated emission. The term $\frac{\bar{n}_2(t)}{\tau_2}$ represents spontaneous emission from energy level 2. In the case of steady state, $\frac{d\bar{n}_2(t)}{dt} = 0$, and the pump absorption term on the left is balanced by the sum of the spontaneous emission and the stimulated emission terms on the right. In general, the spontaneous emission term can be ignored in the amplifier dynamics analysis [23].

Once again, return to the random power on the *HORNET* WDM link. Due to the random nature of the optical power on the WDM link, the sum of the powers on the wavelengths (the term on the far right) will be dynamic. It is clear that when the value of this sum changes, $\frac{dn_2(t)}{dt}$ must take on a nonzero value until the absorption and emission terms can be brought back into balance. Since the gain is dependent upon $n_2(t)$, as shown in Equation 4.4, the amplifier gain will change when $\frac{dn_2(t)}{dt}$ is nonzero. It can be shown that when the WDM signal power decreases, the value of $\frac{dn_2(t)}{dt}$ is positive and thus the gain will ultimately increase. Likewise, when the WDM signal power increases, the gain of the amplifier will decrease. The magnitude of the change is heavily dependent on the saturation level of the amplifier, as is discussed in detail in [23].

This result is verified thoroughly with simulations in [23] and experimentally in [61, 62]. Figure 4.7 shows an example of EDFA gain dynamics obtained in experiments on *HORNET*. As the figure shows, when the power on Wavelength 1 becomes zero, the total input power *decreases*. As a result, the gain of the amplifier *increases*, and thus the output power of Wavelength 2 increases. Similarly, when the total power increases, the gain of the amplifier decreases, and as a result the output power on Wavelength 2 decreases. In this experiment, the peak power on Wavelength 1 is 9.5 dB (a factor of 9) larger than the peak power of Wavelength 2. Thus, when the power on Wavelength 1 goes to zero, the total power drops by 10 dB. At the peak total power, the amplifier is operated at the recommended operating conditions.

Receiving a signal like the one on Wavelength 2 in Figure 4.7 is very difficult because the SNR [63, 62] and the optical detection threshold are changing with the signal amplitude. The experimental results shown in [26] confirm this. Therefore, it is necessary to find an amplification solution that will hold the gain constant when faced with dynamic input power. Such an amplifier is called a *linear optical amplifier* because the output power increases linearly with input power, and thus the gain is

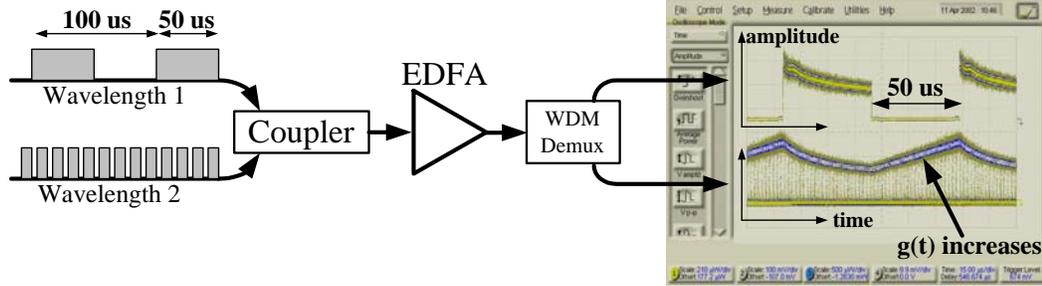


Figure 4.7: The gain of an EDFA changes when the input power changes. In this experiment, the peak power on Wavelength 1 is 9.5 dB higher than the peak power on Wavelength 2.

constant for any input power.

As shown in this section, the gain is directly dependent upon inversion. Therefore, if the inversion can be kept constant despite dynamic input power, then the gain will remain constant. Three solutions have recently emerged that are based on this principle. The first solution uses a concept from laser physics called gain-clamping to maintain constant gain in an EDFA. The second uses the same scientific concept, but the gain medium is a semiconductor instead of fiber. The third solution, which will be discussed briefly, is to control the optical pump in the EDFA to keep the inversion constant.

4.4.2 Gain-Clamped EDFAs

It is a well-known result from laser physics that when a photonic gain medium is lasing, the inversion of the medium does not change, even if the pump power changes. This is easily proven using the laser rate equations for photons

$$\frac{d\phi}{dt} = Kn\phi - \frac{\phi}{\tau_c} \quad (4.6)$$

where ϕ is the number of photons, n is the inversion, K is a constant describing the gain medium, and τ_c is a cavity loss time constant of the gain medium. At the

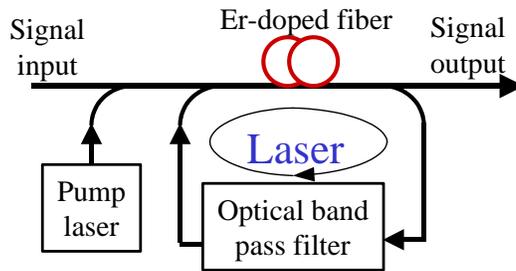


Figure 4.8: Design of a gain-clamped EDFA.

threshold of lasing, $\frac{d\phi}{dt} = 0$, and thus

$$Kn_{th} = \frac{1}{\tau_c} \quad (4.7)$$

where n_{th} is the inversion at threshold. After lasing occurs, the photon level will reach a steady state value. At steady state, once again $\frac{d\phi}{dt} = 0$ and

$$Kn = \frac{1}{\tau_c}. \quad (4.8)$$

The inversion is the same at laser threshold as it is at any steady state laser power. Therefore, the inversion is *clamped* once lasing occurs.

The generic design of a gain-clamped EDFA (GC-EDFA) is shown in Figure 4.8. A laser is set up in the gain medium by the optical feedback loop. Consider the GC-EDFA with no signal or pump applied. As the pump power increases, the amplified spontaneous emission (ASE) from the amplifier increases. Some of the ASE will pass through the feedback loop where an optical band-pass filter selects a small wavelength region. Light at this wavelength continues to circulate through the loop, increasing each time due to the gain of the EDFA. Ultimately, there is a certain pump threshold that will create a laser in the feedback loop at the wavelength selected by the loop filter. The design of a GC-EDFA is thoroughly described in [25].

Once the laser is set up in the gain medium, the WDM signal can be applied to the GC-EDFA. The laser power in the feedback loop will fluctuate as the input

power changes, but the inversion will remain constant, thus resulting in constant gain performance. The work in [25] simulates the performance of GC-EDFAs and verifies that in fact the gain of the amplifier remains constant even when the input power fluctuates randomly. The randomized WDM power at the EDFA input in that work is similar to what is expected in *HORNET*. GC-EDFAs have also been experimentally demonstrated in an optical packet environment like *HORNET* in [62]. The experimental results further confirm that GC-EDFAs maintain constant gain and relatively constant optical SNR. The simulation work in [25] also reveals two current shortcomings of GC-EDFAs. First of all, the output power of the EDFA is reduced because of the gain-clamping design. Secondly, the subsystem suffers from relaxation oscillations when the laser amplitude changes in order to stabilize the gain. Nonetheless, GC-EDFAs are a competitive approach to solving the problem of EDFA dynamics in *HORNET*.

4.4.3 Gain-Clamped Semiconductor Optical Amplifiers

Throughout this report the EDFA has been the only type of amplifier discussed. However, a second well-known type of optical amplifier exists that uses a semiconductor medium to provide gain to the optical signal. Similar to the EDFA, gain occurs through stimulated emission, but the pump is not a photonic pump. Instead, an electrical current is injected into the material to provide the energy to excite the ions to the proper energy levels. One important distinction separates the performance of EDFAs from semiconductor optical amplifiers (SOAs). The time constant observed in the EDFA gain dynamics of Figure 4.7 is relatively slow. However, in SOAs the time constant related to gain dynamics is so fast that even nanosecond-scale changes in input power affect the gain. This means that the bit modulation on a wavelength can modulate the gain of the amplifier. Thus, all bit-streams in the WDM signal modulate each other, causing excessive cross talk. This characteristic of SOAs has

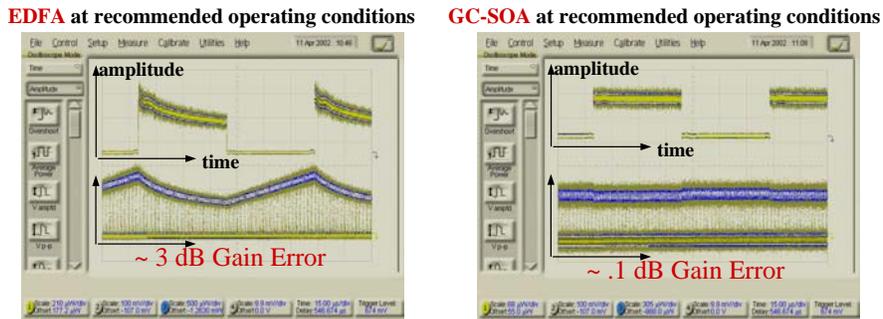


Figure 4.9: The gain-clamped SOA maintains constant gain under dynamic input conditions, whereas the conventional EDFA has dynamic gain.

prevented them from being used in WDM systems.

However, if gain-clamping is incorporated into the SOA to keep the inversion constant, then the cross-talk problem is solved in addition to the problem of random power fluctuations in the *HORNET* link. A clever design for a GC-SOA is presented in [24]. In this design, a vertical cavity laser is built into the SOA. When the vertical cavity laser is lasing, the inversion of the gain medium of the SOA remains constant. This results in constant gain, as shown in Figure 4.9. The figure shows the results of using a conventional EDFA at its recommended operating conditions and the results of using a GC-SOA at its recommended operating conditions. The total power at the input of the amplifier changes by 10 dB when the power on Wavelength 1 changes. The experiment was performed for *HORNET* with a GC-SOA donated by Genoa, a maker of these amplifiers. Clearly, the GC-SOA maintains constant gain despite the large variation of input power.

Two shortcomings exist with the current version of GC-SOAs. As with all semiconductor optical amplifiers, the noise figure is higher than for EDFAs. This will be very difficult to change. However, a more important improvement that may be achieved in the future is higher output power. The GC-SOAs available at the time

of this report have a maximum average output power of 7 dBm (typical EDFAs for high-wavelength systems operate near 20 dBm). This is too small for a network like *HORNET*, which may feature up to and possibly beyond 64 wavelengths. Nonetheless, this will likely improve in the future, making GC-SOAs an excellent candidate for optical amplifiers in *HORNET*.

4.4.4 Transient-Controlled EDFAs

Return for a moment to the rate equation for the EDFA, Equation 4.5. It was previously said that if the WDM signal input (the term on the far right side of the equation) changes, then $\frac{dn_2}{dt}$ is nonzero, and as a result the gain will change. However, notice that when the WDM signal power is modified, the change can be offset by modifying the pump power (the term on the left side of Equation 4.5). If the pump power is changed quickly enough and by the correct magnitude, then $\frac{dn_2}{dt}$ will not deviate far from zero, and the gain will remain relatively constant. This is the principle behind transient-controlled EDFAs. The input power is monitored by the EDFA subsystem and the optical pump(s) is modulated to counteract any changes detected. These amplifiers have not yet been demonstrated in optical-packet-based networks like *HORNET*, but they have been successfully demonstrated under dynamic conditions [26]. They may very well have the potential to be a good solution for optical amplification in *HORNET*, but only experimentation can verify this.

4.5 Summary

As was shown in this section, the novel architecture and protocols of *HORNET* place a burden on the three basic photonic subsystems in *HORNET*: the transmitter, the receiver, and the optical amplifier. A *fast-tunable packet transmitter* and an *asynchronous packet receiver* are necessary to empower the network to efficiently utilize

wavelength routing. Also, a *linear optical amplifier* is required because of the random power fluctuations on the optical ring in *HORNET*. To determine if the architecture of *HORNET* is feasible considering the requirements of the subsystems, these three subsystems were thoroughly investigated.

It was determined that the DBR semiconductor laser is a good candidate for the key component of the fast-tunable packet transmitter. Specifically, the SG-DBR laser was highlighted for its ability to tune quickly over wide or precise tuning ranges because of its opto-electronic tuning mechanism and its Vernier Effect design. Also, the design and operation of the laser-tuning controller developed for *HORNET* was presented. The experimental results obtained with the controller and a DBR laser proved that the fast-tunable packet transmitter is a viable subsystem for *HORNET*. Additionally, convincing results from other related research efforts were presented that further demonstrate the feasibility of the subsystem.

The asynchronous packet receiver, which is the complement to the fast-tunable packet transmitter, was also thoroughly investigated. First, analog solutions that were used in the *HORNET* experiments were presented. Most likely, however, the best solution for a receiver in *HORNET* will be a digital asynchronous receiver. One particular CMOS solution was investigated that uses multiple oscillators to oversample the received bit-stream. Engineering improvements are necessary, but it seems clear nonetheless that a commercial asynchronous packet receiver will be available when *HORNET* is ready for deployment.

Finally, it was shown that power fluctuations in the optical ring in *HORNET* result in gain fluctuations in conventional amplifiers. These fluctuations cannot be avoided because of the *HORNET* protocols, so a new optical amplifier must be developed. Fortunately, three possible solutions are already under development: gain-clamped EDFAs, gain-clamped SOAs, and transient-controlled EDFAs. Experimental results have been obtained on the use of both gain-clamping solutions in an optical packet

environment. Transient controlled EDFAs have not been investigated in such an environment, but they appear to be a possible solution. Once again, it is clear that a viable solution for the linear optical amplifier in *HORNET* will exist in the very near future. Thus, no subsystems will hinder the development and commercialization of the *HORNET* architecture.

Chapter 5

HORNET Testbed: Experimental Demonstrations

5.1 Testbed Description

Over the duration of the project, we have constructed a 4-node bi-directional HORNET testbed (see Figure 5.1 below). The node construction changes depending on the protocols or sub-systems that are being tested. In general, a node consists of 4 main modules: (1) optical/WDM module that consists of add-drop multiplexers, couplers, delay lines and compensating DCF, (2) tunable transmitter consisting of the fast-tuning driver, tunable laser and the MZM modulator, (3) clock recovery module that performs packet clock recovery and (4) the electronic data and control processing module that implements the network protocols such as MAC, fairness, reservation protocols, packet processing functions such as look-ups, switching and queuing as well as data-handling such as serialization/de-serialization, clock recovery, framing/de-framing etc. Individual components used in the testbed such as integrated circuits, lasers, WDM equipment are off-the-shelf components while the design of printed circuit boards that hold these components and link them in a usable manner

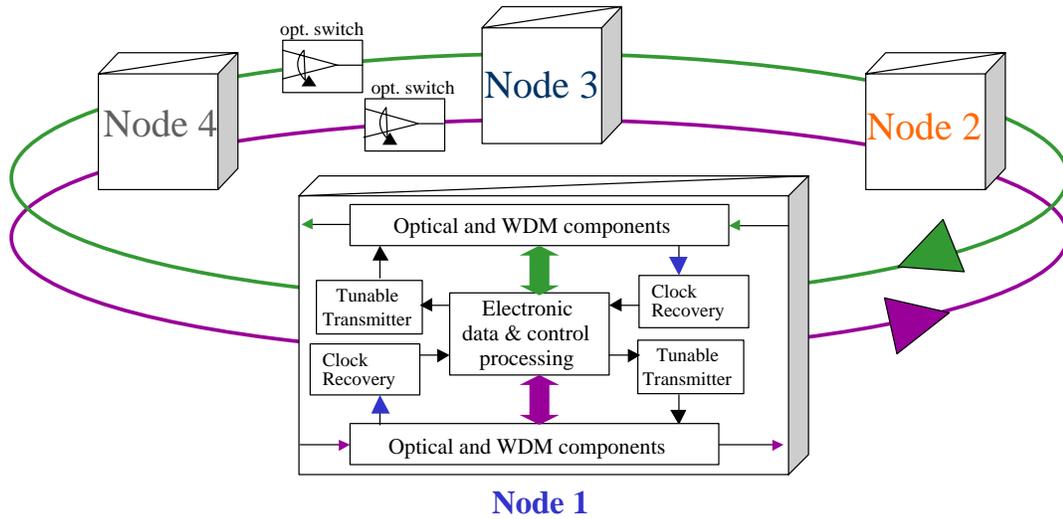


Figure 5.1: Generic Testbed Diagram

was performed in the OCRL. The optical switches shown in the figure are used to simulate fiber cuts in the testbed. In the following sub-sections, we will describe the testbed implementation in greater detail and describe the results obtained.

5.2 Protocol Demonstrations

5.2.1 *HORNET* Media Access Control Protocol and Survivability

Figure 5.2 below shows the detailed testbed implementation that was used to demonstrate HORNET's 2FBPSR architecture as well as the control channel MAC protocol. It contains four nodes, two of which have tunable transmitters and control electronics, one of which is for control only, and one of which is only used to drop wavelengths. Spools of fiber cable are inserted so that propagation delays are present in the testbed.

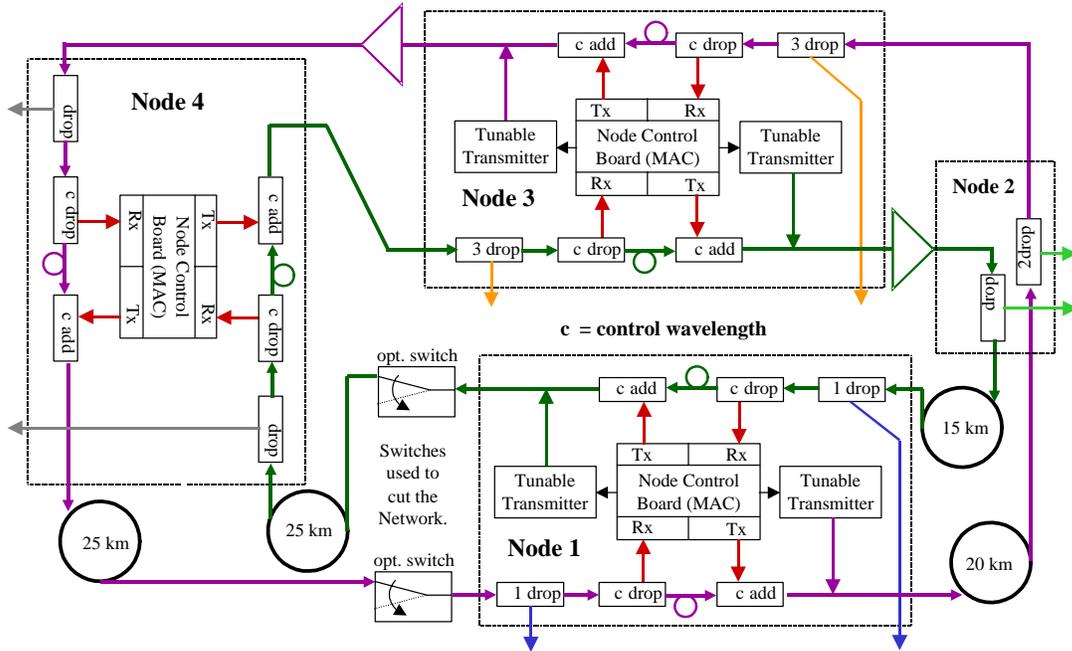


Figure 5.2: The HORNET experimental testbed

Because the testbed has only a few nodes, relatively large spools are used to give realistic propagation delays.

Since there are four nodes on the network, each node sends packets to three other nodes. Under normal operation, the nodes send packets to two of the destinations using the counter-clockwise (CCW) ring, and to the other destination using the clockwise (CW) ring. If a cut occurs in the ring, the nodes adjust the paths as necessary. The tunable transmitters send packets 200 ns in duration on alternating wavelengths (i.e. to alternating destinations) while using the MAC protocol to avoid collisions.

A photograph of the electronics in a node is shown in Figure 5.3 below. Nodes 1 and 3 each have a node controller circuit board and two tunable-transmitter boards. The node controller receives and re-transmits the control channel. It inspects the control channel for any messages from other nodes and for the wavelength availability information for the MAC protocol. It also runs a synchronization protocol at the

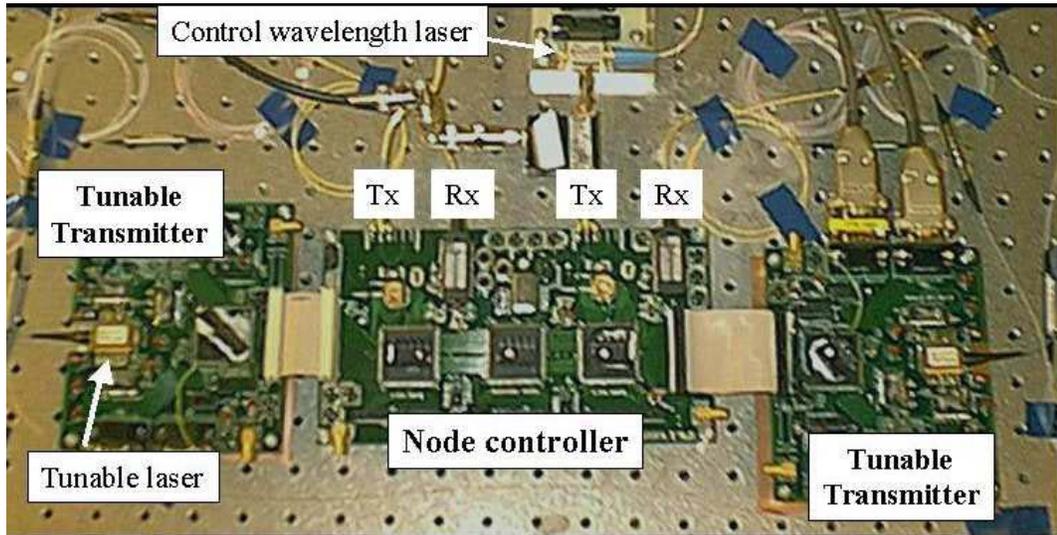


Figure 5.3: Photograph of the electronics in a HORNET testbed node.

startup of the network. Once the network is synchronized, if the controller detects an interruption in the control channel, it assumes that a cut has occurred on the fiber cable between it and its upstream neighbor. It readjusts its routing information and inserts a message onto the control channel. When the other controllers see this message, they readjust their routes and relay the message onto the next node. Readjusting the routes is accomplished by calculating which destinations are on which side of the cut, a simple modulo subtraction operation.

The node's protocols (startup and synchronization, MAC, survivability) are implemented in programmable logic devices (PLDs) on the control board clocked at 125 MHz. A Gigabit Ethernet chip set is used for the transmission and reception of the control channel in the testbed. Since the testbed protocols can be implemented in PLDs, it is clear that more complex components that are typically used in commercial networking equipment can handle practically scaled versions of the protocols. To force the network to find a cut and to restore the broken paths, two optical switches controlled by a function generator periodically cut and repair the ring between Nodes

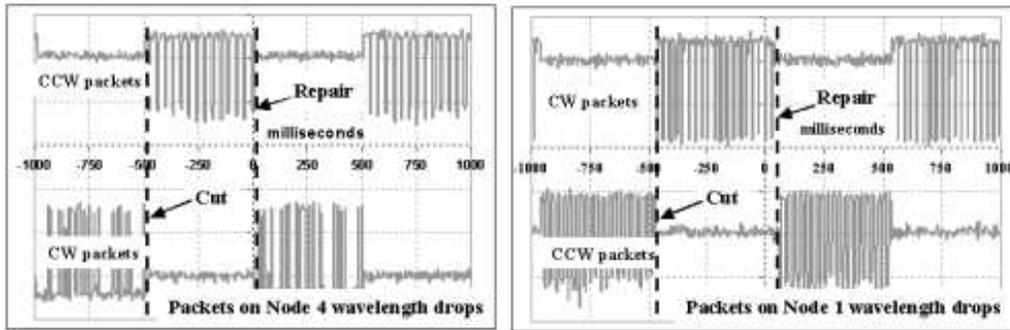


Figure 5.4: (a)Packets transmitted to Node 4 from Node 1. After a cut, Node 1 must send packets to Node 4 in the CCW direction. (b) Packets transmitted from Node 3 to Node 1.

1 and 4, as shown in Figure 5.2. When the cut occurs, Node 1 needs to use the CCW ring to reach Node 4 because the CW ring has been cut between the two nodes. Figure 5.4(a) shows this result. The packets dropped by each of the WDM filters in Node 4 are detected by two APD detectors and viewed on an oscilloscope. As the figure shows, when the cut occurs, Node 4 stops receiving packets from Node 1 on the CW ring and begins receiving packets from Node 1 on the CCW ring. When the cut is fixed, Node 1 stops sending packets in the CCW ring and begins sending them again in the CW ring. The transmitters in Node 3 are disconnected from the ring during this observation so that only the packets from Node 1 are observed at Node 4. Please note that DC blocks in the receiver cause the signal level to drift, and that the oscilloscope cannot sample fast enough to avoid aliasing. This explains the irregular appearance of the waveforms in Figure 4.

When the cut occurs, both Node 1 and Node 4 detect it. Both send control messages around the ring away from the cut notifying other nodes. Node 3 then receives the message and adjusts its routes. Figure 5.4(b) shows the occurrence from the perspective of Node 1. It originally receives packets from Node 3 on the CCW

ring. However, after the cut, Node 3 is forced to use the CW ring. When it learns that the cut is repaired, it resumes transmitting in the CCW direction.

Because *HORNET* does not use point-to-point link-based protocols, no setup time is required to begin using a new path. Thus, the restoration of a path happens nearly instantly. The only cause for downtime between two nodes is the propagation delay of the control messages around the ring. Figure 5.5 shows a zoomed-in view of the cut event and the repair event from the perspective of Node 1 as it receives packets from Node 3. When the cut occurs at the input to Node 1, the packets from Node 3 are no longer received on that path. Node 1 immediately sends a control message around the CCW ring. After propagating through approximately 20 km, the message reaches Node 3. Node 3 then stops sending packets to Node 1 in the CCW direction and instead sends them in the CW direction. The first packet in the CW direction must propagate through 15 km of fiber before reaching Node 1. Since the cut message propagates through 20 km of fiber and the first packet in the restoration path propagates through 15 km of fiber, the path restoration process is delayed by 35 km of fiber. Since light travels through 1 km of fiber in 5 ms, there is approximately 175 ms of delay between the moment that Node 1 receives the last packet from Node 3 in the CCW direction and the moment when Node 1 receives the first packet from Node 3 in the restored path. Because of the slow rise and fall time of the optical switch, the precise time at which Node 1 determines that the link has been cut is difficult to decipher. Nonetheless, it is clear from Figure 5.5(a) that there is approximately 175 ms between the time of the cut and the time when the first packet arrives along the restored path. This means that Node 3 was unable to successfully send packets to Node 1 for approximately 175 ms.

Figure 5.5(b) shows the events in the receivers of Node 1 when the cut is repaired. When Node 3 receives the message that the cut is repaired, it stops sending packets to Node 1 in the CW direction and begins sending them in the original direction. The

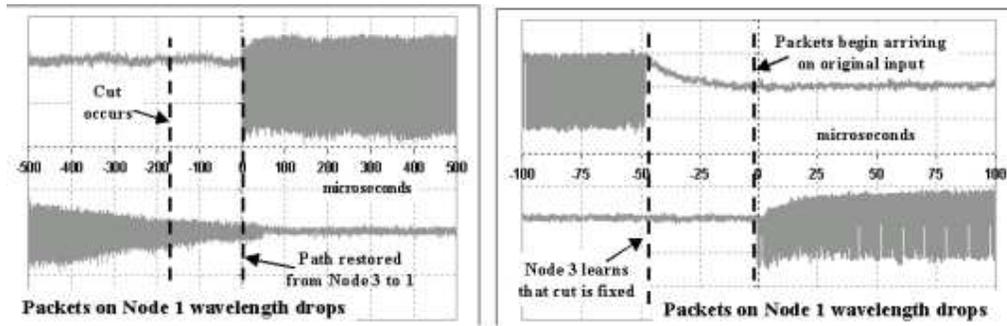


Figure 5.5: (a) Restoration delay for the path from Node 3 to Node 1; (b) Transition of routes in Node 3 after the cut is reported as fixed (delay is only due to differences in fiber length along paths).

final packet sent in the CW direction travels 15 km before reaching Node 1, while the first packet in the CCW direction travels 25 km before reaching Node 1. Thus there is 50 ms of delay between the last CW packet and the first CCW packet, as can be observed in Figure 5.5(b). This does not correspond to time during which Node 3 is unable to reach Node 1. It is only a result of the difference in path lengths. If the CCW path were shorter, there would be some overlap. This implies that care should be taken in practical networks to avoid a temporary mis-ordering of packets when a cut is repaired. This can easily be accomplished by waiting a fixed, pre-determined amount of time before switching back to the original transmission direction.

To test the control channel MAC protocol, the testbed was arranged as shown in Figure 5.6 below:

The node under test receives the control channel from upstream nodes. Upstream nodes also insert dummy packets on different wavelengths, leaving certain slots empty. In the figure, upstream packets are seen simply as a high-level while packet inserted by the node under test can be seen as a solid block. The node under test first finds an opening on λ_2 and starts inserting a long packet. The node then detects an incoming packet on λ_2 and it stops transmitting on λ_1 to avoid a

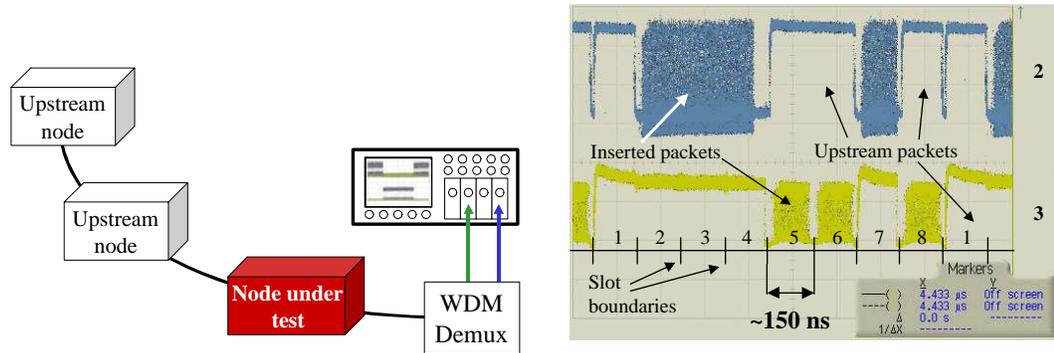


Figure 5.6: Testbed setup for MAC protocol Demonstration

collision. Instead it tunes to λ_2 and inserts two short packets, one per slot. It then tunes to λ_1 and continues its interrupted transmission, and so on. This proves the feasibility of the Segmentation and Reassembly On Demand (SAROD) protocol described earlier.

5.2.2 Control Channel Frame Synchronization Protocol

The control channel design significantly impacts both the performance and the cost of the network. This section discusses the synchronization of the control channel with the packets on the payload wavelengths. Included in this discussion is an experimental demonstration of a frame synchronization protocol developed for *HORNET*.

In every *HORNET* node, the control channel is processed and retransmitted while packets on the payload wavelengths pass through an all-optical path. The control channel must be retransmitted in perfect alignment with those packets. However, two issues can prevent that from happening. The first issue is a lack of synchronization between the incoming and the retransmitted control channel at each node, while the second issue is the difficulty in manufacturing a node with a perfect match between the payload path and the control channel path. Both of these issues are solved with the establishment of a frame synchronization protocol for *HORNET*.

The node's control process is in general not perfectly synchronized with the incoming control channel, and thus the process will begin at a random moment with respect to the moment of arrival of the SOF indicator, which drives the control process. Within each node, the random time difference between the actual arrival of the SOF indicator and the detection of the indicator is uniformly distributed across one clock cycle (the node uses a byte-clock). The random misalignment adds stochastically at each node, resulting in a large variance after several nodes of propagation. It can be shown that after an optical packet propagates through 32 nodes, there is a probability of 0.001 that it will be misaligned from the control channel SOF indicator by at least 11 control channel bytes. However, this issue can be easily solved by using a phase-locked loop (PLL) within the control channel receiver to synchronize the control channel process with the incoming control channel bit stream. Thus, the first requirement of the frame synchronization protocol is the use of a PLL to obtain synchronization from the incoming control channel.

The second issue that causes control channel frame misalignment is designing, manufacturing, and maintaining a perfect match in propagation delay between the control channel path and the payload wavelength path. Figure 5.7 illustrates the two paths, including splice locations. To make the paths match, splices and fiber lengths must be tightly controlled. More importantly, the design of the electronics and micro-code are critical because every modification in the design process and *every upgrade after the product release* may cause a path difference. Any error due to the micro-code will be present in every node, and thus the resulting misalignment will add as packets traverse the ring.

This issue is solved in the frame synchronization protocol by *automatically* calibrating the control channel path propagation delay to match the payload wavelength path. Figure 5.8 shows the important components involved in the calibration. The two highlighted components, the PLL phase selector and the delay states, are used to

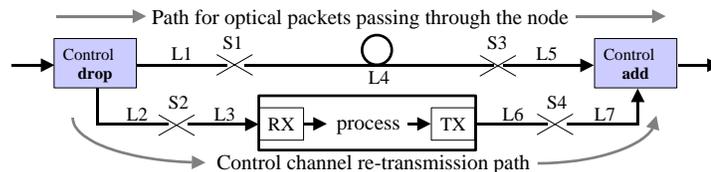


Figure 5.7: The control channel path and the payload wavelength path. S_n denotes splice locations, L_n denotes fiber lengths.

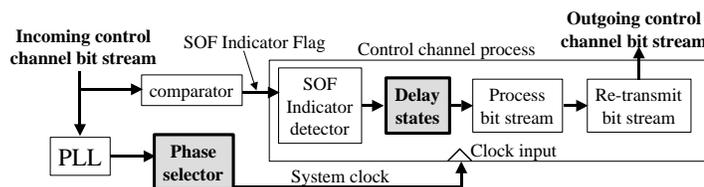


Figure 5.8: The output phase of the PLL and the delay states are controlled by the node to provide perfect control channel frame synchronization.

adjust the propagation delay through the control channel path. The node programs the delay states to adjust the propagation delay in increments of a process clock cycle. Also, the node can control the output phase of the PLL, which dictates the moment at which the incoming SOF indicator comparator output flag is sampled. Sampling the SOF comparator output flag near the beginning of its duration will shorten the propagation delay of the control channel path, just as sampling the flag near its end will lengthen the propagation delay.

The calibration requires two steps to achieve nearly perfect SOF indicator alignment. The first step is a laboratory calibration (*lab cal*) to put the node in a position to perform its auto alignment when in the system. This is a manual step performed by an operator before the node is installed in the network.

The lab-cal system setup is shown in Figure 5.9. The operator arranges the *lab cal* system such that the SOF indicator flag and the front edge of the dropped packet arrive to the processor at exactly the same instant. The operator then adjusts the node's logical delay states and clock phase until the retransmitted SOF indicator and

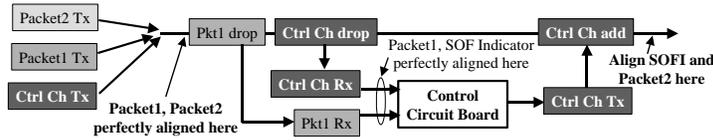


Figure 5.9: The setup for the *lab cal* procedure.

the through-packet are perfectly aligned at the node output as they are intended to be in the network. This provides a reference state for the node.

Once the node is placed in the network and is turned on, one of the first things it must do is to perform the in-system calibration (IS-cal). The network contains at least one master node, which is notified of the new node on the network. The master node sends a long stream of short packets to the network's new node. The node measures the time duration between the arrival of the front edge of the packets and the arrival of the SOF indicator. The time is measured to within the phase adjustment granularity of the PLL (most likely $\frac{1}{8}$ of a clock cycle). Since the retransmission of the SOF indicator is currently set (by the *lab cal*) for the condition where the SOF indicator and the packet arrive simultaneously, the node knows that it should adjust the control channel propagation delay by the time difference that it measures between the incoming packets and SOF indicators.

To measure the time difference between the arrival of the SOF indicator and the calibration packet from the *master node*, the node cycles its PLL output clock through all phases, taking several samples between each PLL phase adjustment. As shown in Figure 5.10, the adjustments alter the relationship between the clock phase and the incoming SOF indicators and packets. It can be shown that the actual time difference between the SOF indicator arrivals and the calibration packet arrivals is the average over all phases of the number of samples between the two arrivals.

For example, in Figure 5.10, when the phase of the sampling clock is zero, the controller measures *one sample* between the arrivals of the signals. The same result

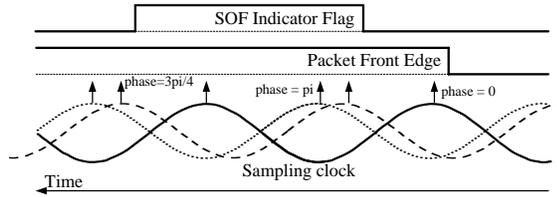


Figure 5.10: During the *IS-cal*, the node measures the time difference between the arrival of the SOF indicator flag and the packet front edge by cycling its process clock phase through all possible phases.

will occur for phases of $\frac{\pi}{4}$ and $\frac{\pi}{2}$. At phases of $\frac{3\pi}{4}$ through $\frac{7\pi}{4}$ the node measures *zero samples* between the two arrivals. The node determines that the time difference between the arrivals is $\frac{3}{8}$ of a clock cycle (average difference in samples over the eight phases). Once the node has determined the time difference between the arrivals of the SOF indicators and calibration packets to within the granularity of the phase adjustments, it adjusts the number of delay states and it reprograms its PLL output phase in order to adjust the propagation delay of the control channel path.

Frame Synchronization Demonstration

An experimental testbed was assembled to demonstrate the *HORNET* frame synchronization calibration procedure. As shown in Figure 5.11, three experimental *HORNET* nodes are connected together. The nodes use a PLL with an adjustable output phase to synchronize the control process with the incoming control channel, as specified by the protocol. Gigabit Ethernet (GbE) is used for the control channel, and thus the SOF indicator is the GbE 'comma' byte (*1100000101*). The lab cal procedure was performed on the nodes to set the reference condition. The IS-cal was then performed on Node 1. The IS-cal of Node 2 is described below.

The phase of the local clock in Node 2 was cycled through 8 phases. The measured alignment of the packet front edge and SOF indicator with the different phases of the sampling clock is shown in Figure 5.12. For this node, the samples result in a

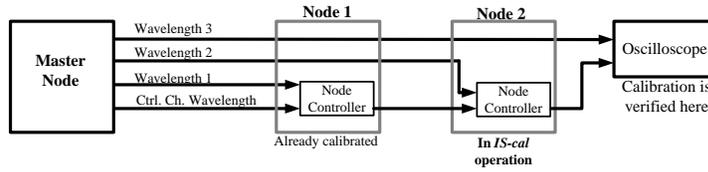


Figure 5.11: The setup of the *IS-cal* procedure for a node downstream of a previously calibrated node.

difference of one cycle for phases of $0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi,$ and $\frac{5\pi}{4}$. For the the phases of $\frac{3\pi}{2}$ and $\frac{7\pi}{4}$, the difference is zero. Thus, the node determines that the phase needs to be advanced by $\frac{6}{8}$ of a clock cycle. However, note that if the node advances (subtracts) its phase by $\frac{6}{8}$ of a clock cycle, the outgoing SOF indicator is actually *delayed* by $\frac{2}{8}$ of a cycle. This always occurs when the SOF indicator bit boundary is crossed (i.e. when the sampled difference changes from one to zero clock cycles). Thus, when the node determines that the boundary was crossed (as revealed in this example by the samples for $\frac{3\pi}{2}$ and $\frac{5\pi}{4}$, the node also subtracts one logical delay state. Figure 5.13 shows the result of the experimental demonstration. Figure 5.14 compares the alignment of the SOF indicator and a packet after two nodes of propagation with and without the frame alignment protocol. The time-lapse image of Figure 5.14 (a) shows the random misalignment that occurs without the protocol.

The results of this experiment shown in Figures 5.13 and 5.14 show that the alignment accuracy is within *one bit* of the control channel bit rate (1 ns in this case, since GbE is used). This is because the adjustment precision of the PLL is $\frac{1}{8}$ of a clock cycle, or one bit. In general, the accuracy may only be as good as a few bits because of the possibility that the correct alignment would have the clock sampling the 'edge' of the SOF indicator (in such a acase the sampling clock is adjusted slightly). As long as the accuracy is within *one byte*, then only one byte of guard band is necessary, and thus only one byte of overhead is used.

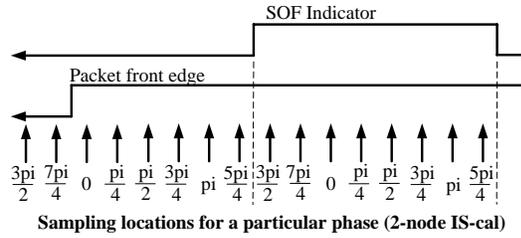


Figure 5.12: The location of the samples for all phases for the two incoming waveforms in the *IS-cal* procedure of the second node.

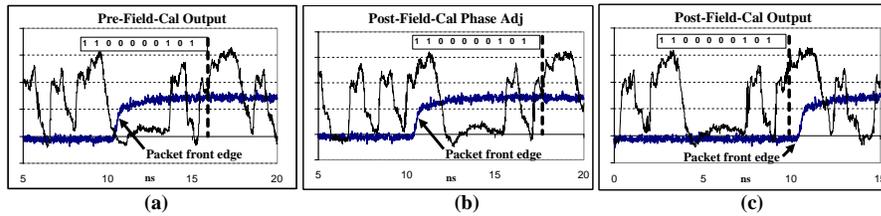


Figure 5.13: (a) Alignment of retransmitted control channel SOF indicator with a packet passing through the node before the *IS-cal*; (b) After the phase adjustment portion of the *IS-cal*; (c) After the complete *IS-cal*.

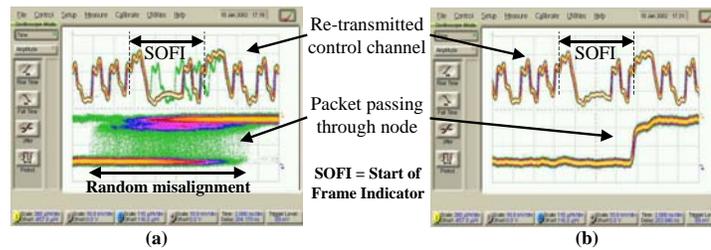


Figure 5.14: Time-lapse image of the retransmitted control channel and packets after two nodes of propagation. (a) Random misalignment with no frame synchronization protocol. (b) Perfect alignment with the protocol.

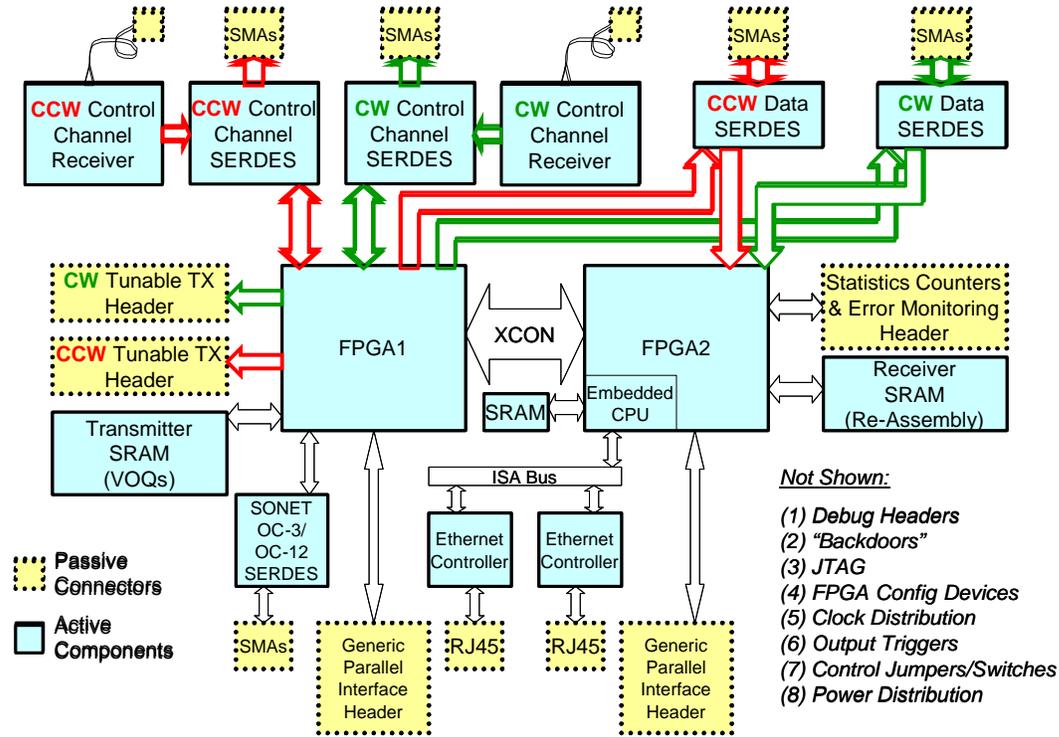


Figure 5.15: HORNET data and control processing module Printed Circuit Board

5.2.3 Data and Control Processing Module: single-PCB implementation

During the course of the project, we have learnt from previous versions of the *HORNET* node design to make newer, better implementations. For example, we progressed from using Cypress Programmable Logic Devices (PLDs) to bigger, denser Field Programmable Gate Arrays FPGAs, made by Altera. We have also extended the capabilities of *HORNET* to support more interfaces on the Access side of the network. The new hardware has evolved from concept to reality over the past quarter. Figure 5.15 below shows a functional block diagram of a PCB that combines all the data and control processing functions into one unit.

The top half of the diagram contains the electronics to interface to the *HORNET* MAN, while the bottom half contains the interfaces to the Access network. As you can see from the diagram, there are both clockwise and counterclockwise components on the MAN side of the board. Logically, the two FPGAs can be considered as one large processing node.

The main data flow from access to *HORNET* can be described as follows: data from an access link enters the *HORNET* node (via an Ethernet controller, for example). The Ethernet frame is stripped and examined by the on-chip embedded microprocessor on FPGA2 and then placed into an appropriate queue for transmission. The two FPGAs constantly monitor the states of the transmit queues and the information about wavelength availability coming in on the control channel. Based on circuit and packet scheduling decisions, the tunable transmitter is tuned and the appropriate transmitter queue is read, pumping data onto the *HORNET* Ring.

Similarly, to summarize the flow from *HORNET* drop filter to access: a data packet is dropped at its destination node. Clock and data are recovered, the data is framed and examined and placed in the appropriate queue for reassembly. When the message has been received, it is passed from the queue, through the embedded microprocessor, to the Ethernet Chip and onto the access network.

In addition to the data paths, control channel information is terminated, processed, and re-transmitted out of this board. In addition, statistics and error monitoring are implemented within the FPGAs. The main datapaths are summarized in Table 5.1 below:

Figure 5.16 below is a photograph of the data and control module, and Figure 5.17 contains the revised node block diagram.

Data Path	Description
LAN side Ethernet -> Embedded MPU	Two ISA based Ethernet controllers on the board allow us to source and sink Ethernet traffic.
LAN Side SONET Interface	A multi-rate SERDES on the LAN side allows us to bring circuit-based data into and out of the Access/LAN side of HORNET.
MAN Side Control Channel CW and CCW	Sumitomo Photoreceivers and Vitesse SERDES allow both clockwise and counter-clockwise control channel signals to be terminated and retransmitted.
FPGA1 <-> FPGA2 Interconnection	In order to resemble one logical device, the two FPGAs are tied to each other with a very wide bus.
MAN Side CW and CCW Data	Multi-rate SERDES on the MAN side allow high speed transmission and deserialization for HORNET data. Keep in mind that clock recovery happens off-board.
Various SRAM Interfaces	SRAM provides storage for Virtual Output Queuing (VOQs), Re-Assembly Buffers, and program memory for the embedded micro-processor.

Table 5.1: HW implementation data paths

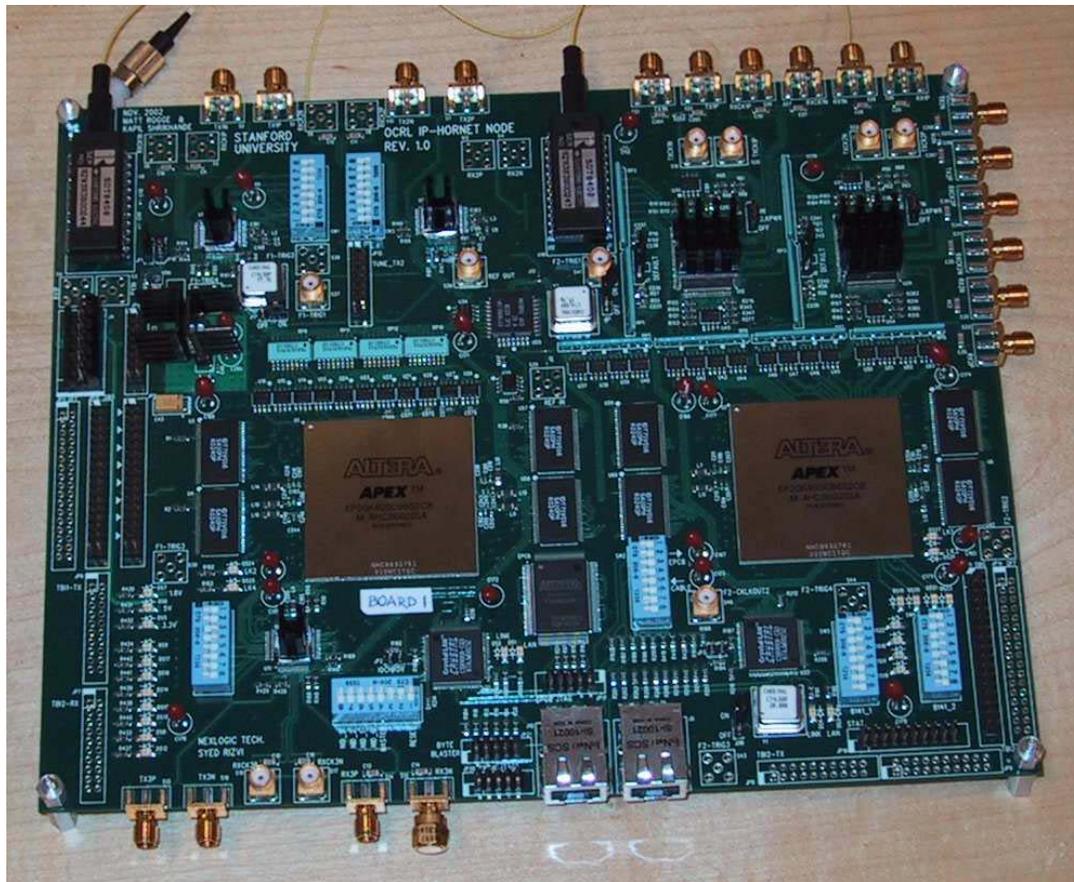


Figure 5.16: Photo of HORNET data and control processing module Printed Circuit Board

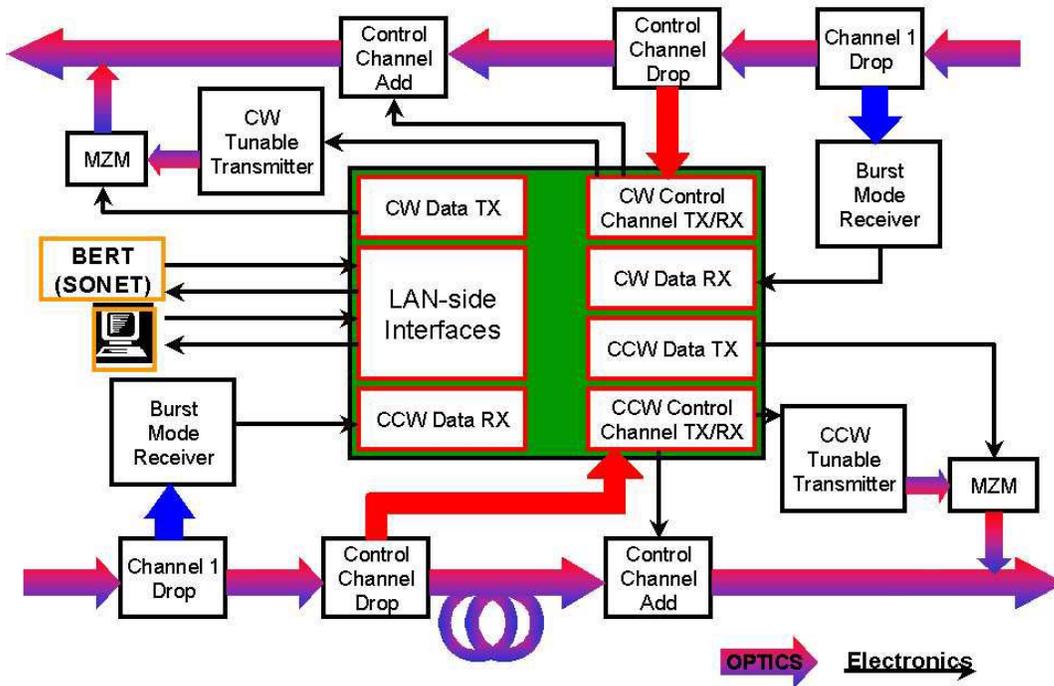


Figure 5.17: HORNET Block diagram with data and control processing module Printed Circuit Board

Chapter 6

Deployment Issues

6.1 Introduction

This chapter contains work related to evaluating the practicality of *HORNET* as a real-world network. In particular, we developed a realistic power budget for *HORNET*. The Power budget contains theoretical calculations and simulations on scalability and network physical limitations of HORNET. The unique physical topology of HORNET poses several interesting system design challenges, many of which are addressed in this chapter.

6.2 HORNET Power Budget Analysis

In order to understand the system design challenges related to HORNET, we first need to review HORNET's physical node architecture. Figure 6.1 is a block diagram of the design of the HORNET node. At the input (left), a wavelength drop removes the control channel wavelength from the WDM ring. The wavelength for the control channel should be well separated from the payload wavelengths (e.g. 1310 nm) to allow inexpensive transmitters to be used for the transmission of the control channel.

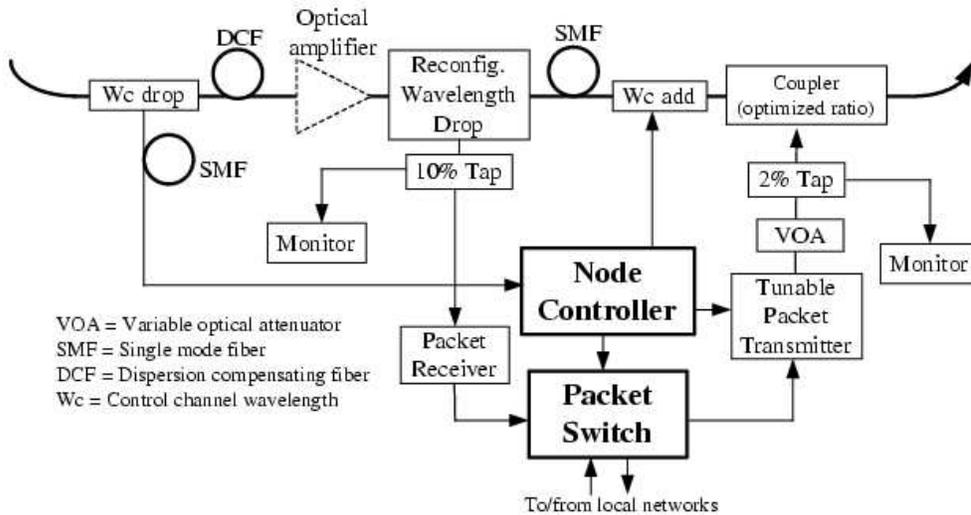


Figure 6.1: Block diagram of the HORNET node

The incoming control channel is received in the Node Controller where the bit stream is analyzed for wavelength availability information, DQBR requests, and any other control information.

After passing through the control channel wavelength drop, the signals on the payload wavelengths traverse an optimized length of dispersion compensating fiber (DCF) optic cable. This fiber cable is necessary to keep the packets on all of the payload wavelengths aligned with the control channel frames. As the optical signals propagate through the single mode fiber (SMF) optic cable in the backbone ring, the wavelengths propagate at different speeds. Thus, the optical packets on the payload wavelengths become misaligned by different amounts with the control channel frame boundaries. The optimized length of DCF re-aligns the packets.

Following the DCF, an optical amplifier is used to boost the power of all of the signals on the payload wavelengths. Note that it is important to drop the control

channel wavelength before the amplifier because it only provides gain to the payload wavelengths. The amplifier will likely significantly attenuate signals on wavelengths outside of the payload wavelength band, such as at 1310nm, because of filtering components within the amplifier subsystem. Optical amplifiers are not necessarily contained in all nodes (thus the amplifier is illustrated with a dotted line in Figure 1.)

After the WDM signal receives the necessary boost, a wavelength drop removes the wavelength(s) from the ring that is (are) destined for the node. Optical packets dropped into the node are received by an asynchronous packet receiver and are then sent to the packet switch, where they are switched onto the access network to which they are destined. Ideally, the wavelength drop is reconfigurable, such that it can allow the network to provision a particular set of wavelengths for a node in order to efficiently accommodate varying traffic patterns. The reconfigurable optical drop should be designed such that it can drop between 1 and M wavelengths, where M is the most wavelengths the node will require, and is typically much smaller than W (the total number of wavelengths in the network).

The payload wavelengths then pass through a wavelength add that multiplexes the control channel wavelength onto the backbone. It is imperative for the control channel frames to be multiplexed in perfect synchronization with the packets on the payload wavelengths, so an SMF delay line is located just before the wavelength add. The delay line (in addition to the other components between the control channel wavelength add and drop) holds the packets on the payload wavelengths while the control channel is being processed. Although the delay line can be designed to approximate the necessary delay, it is very difficult to maintain a perfect match between the payload wavelength path and the control channel propagation path, especially considering that the electronic design of the control channel propagation path will be

upgraded several times after the product is deployed. To solve this problem, a calibration routine was developed to allow the node controller to automatically adjust the propagation delay through the control channel path such that it nearly perfectly matches the payload wavelength path.

Near the output of the node, the fast-tunable packet transmitter inserts packets onto the backbone ring on the wavelength that is received by the packet's destination node. A variable optical attenuator (VOA) is placed at the output of the tunable transmitter to control the output power. This is necessary because the node must transmit its packets at a power level to match the power level of the packets passing through the node. This power level is dependent upon the location of the nearest optical amplifier (recall that amplifiers are not necessarily located in all nodes).

The initial design goals of the *HORNET* network are to carry 64 wavelengths in each ring and to support tens of nodes. Unlike conventional metro networks, wavelengths are not regenerated in every node, or even in every few nodes. It is possible for a packet to traverse the entire network without being converted to an electronic signal. Since this includes several tens of nodes, optical signal-to-noise ratio (OSNR) is a concern in the design of the *HORNET* network. Thus, a mathematical model was created to analyze the system performance of the *HORNET* network.

The bi-directional dual-ring architecture of *HORNET* requires the signal to take the shorter route to the destination. Therefore, for a *HORNET* ring of N nodes, the maximum number of nodes needed to be passed through for the signal before being received is $\tilde{N}/2$. This can be a challenging issue for power budget if the number of nodes is to be up to $\tilde{100}$, meaning the number of nodes to be propagated is $\tilde{50}$. Considering the loss due to transmission fibers, loss incurred due to various optical components, and loss due to dispersion compensating fiber (DCF) modules needed in *HORNET* architecture for each node, optical amplifiers are definitely one of the essential components in the network. The introduction of amplifiers, however, also introduces

optical noise accumulation and thus degradation of OSNR. It is therefore very important when one has to consider the specifications of optical amplifiers required for the *HORNET* architecture as well as the optimal performance achievable via current amplifier technologies. Moreover, because of the non-ideal behavior of optical amplifiers and tunable lasers that need to be taken into account, design/operational margin has to be provided in the system in order to ensure network functionality at any time. In this section, we will investigate practical issues of power budgets in *HORNET* architecture. Note that this can potentially be applicable to other transparent optical networks as well.

HORNET employs tunable transmitters and fixed-wavelength receivers (in the future they can be upgraded to reconfigurable wavelength drops) for the signal channels. The signal channels will thus remain in the optical domain until reaching the destination nodes. As shown by Figure 6.1, there are various components in each node along the path of the transit channel that can bring loss to the signal together with the loss of the transmission fibers. Depending on the choice of components and the quality of the splicing points, the total loss of a typical *HORNET* node can vary from 5–10dB. In other words, it is usually the case that the loss incurred in the node is higher or at least comparable to the loss incurred in the transmission fibers. Considering the typical situation in which the signal has to travel through tens of nodes before dropped, it is definitely not a trivial task for the system designers to make sure the signal quality remains satisfactory throughout all the downstream nodes. Note that we do not consider the signal quality of the control channel since it is dropped at every node and power budget is usually not a concern. In order to evaluate the power budget in *HORNET*, the insertion loss of every component in the node should be carefully specified. The estimated insertion loss of each component along the link or of interest is shown in Table 6.1 below:

The components listed in Table 6.1 are those along the path of the transit channel.

Component	Estimated Loss (dB)	Comment
Wc Drop	1	
DCF Module	0.6/km	For compensating GVD of SMF
Reconfigurable Wavelength Drop	4	Not Mature yet
SMF	0.1	Fiber loss negligible due to short length
Wc Add	1	
Coupler	-	Ratio to be optimized
Splicing	1	Estimated total splicing loss

Table 6.1: Estimated loss for components in a HORNET node

Note that the VOA after the tunable laser has to be adjusted such that the power of the add channel is the same as the transit channel. Nevertheless, it is possible to eliminate the VOA if one can design the coupler ratio such that the power of the transit and add channels to be the same. This should be a more favorable design option if applicable since one can usually assign more percentage of the coupler to the transit channels and therefore reduce the loss of the transit channels due to the couplers. In our following analysis, however, we still assume the existence of the VOA and assume the coupler ratio to be 90/10 (transit/add). One can also assume that couplers with the desired coupling ratio of the transit and add channels are available. In that case, the system performance should be derived given the desired coupling ratio.

Optical amplifiers are the enabling technology for transparent WDM networks. Nevertheless, optical amplifiers add noise to the optical channels and degrade OSNR. This should be taken into account whenever one considers scalability for any transparent optical network architecture. In metropolitan area networks where the design is highly cost-sensitive, we would like to reduce the number of optical amplifiers and/or

to employ amplifiers of lower cost and acceptable performance, whereas the network functionality will not be affected significantly. Since amplifiers of lower cost usually mean inferior noise figure and lower gain, required specifications for the amplifiers should be met given a specific network architecture. One good example mentioned earlier is the gain-clamped semiconductor optical amplifier with lower gain, mediocre noise figure, and a competitive cost. It therefore requires a thorough analysis as for under what conditions could the SOAs be employed, or that one should simply employ gain-clamped EDFAs or transient-control EDFAs to achieve desired system performance. Unfortunately, even for gain-clamped EDFAs and transient-control EDFAs where they provide better performance than the gain-clamped SOAs in general, the implementation needed to keep constant gain comes with the tradeoff of lower gain and higher noise figure due the creation of the lasing wavelength, and the introduction of extra couplers for power monitoring, respectively. Therefore, system designers should bear in mind that amplifiers with noise figure near the quantum limit are not available if they are designed to have gain-control functionality under dynamic power environment like *HORNET*, and thus should give reasonable requirements for the specifications of the amplifiers.

In order to investigate the limit of scalability of the *HORNET* architecture, we consider the general case in which ASE noise from optical amplifiers is the main source of noise. The ASE noise generated from optical amplifiers accumulates linearly with the number of nodes since the noise is regenerated in the same way as the signal. Since the received power of the signal is kept the same, we expect the OSNR degrades linearly with the number of nodes through which the signal propagates. However, ASE noise is not the only noise source in the network. As seen in the node configuration, there will be finite isolation of the reconfigurable wavelength drop on the port of the transit channels. Therefore, power of the dropped channel will leak into the main ring and interfere with the ADD channels at the same wavelength.

Fortunately, the interference only happens once - in the received node where the wavelength is supposed to be dropped. Isolation of 20 $\tilde{3}$ 0 dB is expected for state-of-the-art wavelength drop, and we assume the worst case 20 dB in our following analysis. Later we will see that the interference actually imposes an upper limit of the OSNR.

In order to reduce the network cost, we hope that the number of amplifiers can be reduced by placing an amplifier every two or even three links instead of one amplifier per link. However, several issues have to be considered: (1) There should be optical amplifiers before the signal level falls below the quantum limit. (2) The ASE power scales linearly with amplifier gain, meaning a 30dB amplifier will generate 100 times of noise compared with a 10dB amplifier. (3) Few commercial optical amplifiers provide gain beyond 40dB. Because of the above reasons, we will investigate the possibility of up to three links sharing one amplifier since three typical links in *HORNET* will give more than 30dB attenuation already.

The OSNR versus number of nodes propagated is shown in Figure 6.2 for different amplifier parameters. Some of the key systems parameters for both cases include: Number of channels = 100; channel spacing (and thus optical bandwidth) = 50GHz; Length per link = 15km. Note that the zigzag behavior are due to the fact that the OSNR remains the same until the signal reaches the next amplifier stage. Also note that the OSNR has an upper limit of 20 dB due to the imperfect isolation of the wavelength drop as mentioned earlier. It can be seen that when three nodes share one amplifier, the OSNR degrades rapidly along the links and suffers from very high penalty when the signal propagates through more than 10 nodes. The exact requirement of OSNR really depends on the detailed receiver structure (which will be modeled with Gaussian noise assumption in the following paragraphs. Nevertheless, the exact BER performance should really be investigated based on specific receivers used). Still, it is unlikely the BER will be satisfactory given OSNR below 10dB.

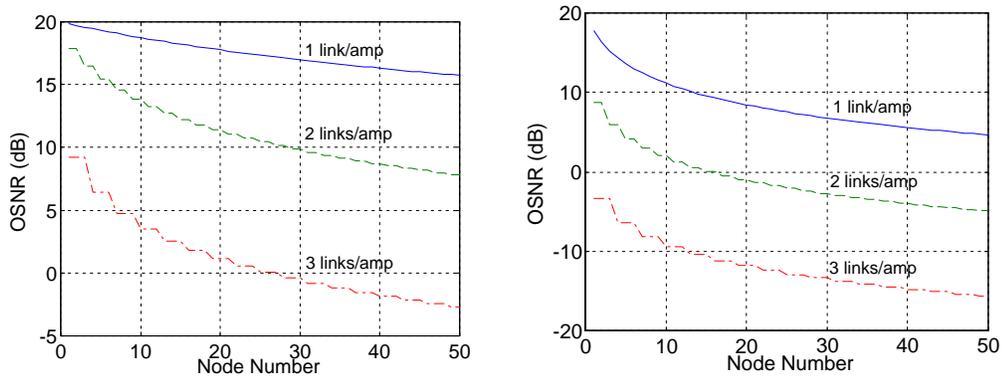


Figure 6.2: OSNR vs. Number of nodes propagated for different amplifier parameters (left) $P_{\text{sat}} = 20\text{dBm}$, $NF=5\text{dB}$ (right) $P_{\text{sat}} = 10\text{ dBm}$, $NF=8\text{dB}$

Judging from the criteria, one amplifier every link is necessary if the number of *HORNET* nodes is to be ~ 100 whereas two link per amplifier may be satisfactory for number of nodes up to 60 for a good amplifier with 20dBm output power and 5dB noise figure. The amplifier is very likely be a transient-control EDFA since gain-clamped EDFA gives worse noise figure. The amplifier with 10dBm output power and 8dB noise figure are typical parameters for SOAs. As can be seen from the figure, the amplifier can barely support a *HORNET* ring with up to 15 nodes even with one amplifier per link. Therefore, significant improvement of SOAs still has to be made if they are to be employed in a transparent network.

In order to visualize the degradation of transmission performance in the receiving end in a more quantitative way, the bit error rate performance is modeled with standard Gaussian noise assumption (i.e. the bit error rate is estimated by the Q-factor). The noise sources in the electrical domain come from shot noise, thermal noise, signal-spontaneous beat noise, and spontaneous-spontaneous noise, among which signal-spontaneous beat noise usually dominates. The transmission performances for a 10Gbps system are shown in Figure 6.3.

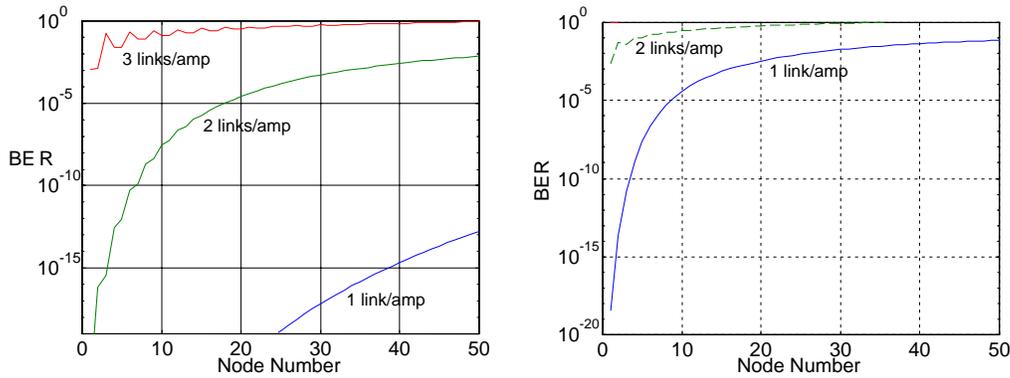


Figure 6.3: Calculated BER versus node number based on Gaussian noise assumption for a 10Gbps system for different amplifier parameters (left) $P_{sat} = 20\text{dBm}$, $NF=5\text{dB}$ (right) $P_{sat} = 10\text{ dBm}$, $NF=8\text{dB}$

When more than one nodes share one amplifier as mentioned earlier, the amplification has to grow exponentially compared to the case that each node has an amplifier, which introduces much larger ASE noise. Besides, when several nodes share one amplifier, the received power for the node downstream to the optical amplifiers are smaller due to propagation loss, which also introduces penalty in the receiver. In Figure 6.3 we can see that the bit-error-rates are obviously higher when more nodes share one amplifier. For this specific system configuration, we conclude that one amplifier should not be shared by more than 2 nodes given the loss parameters in Table 6.1. Nevertheless, the system performance could be improved if components with less insertion loss could be employed. For example, Figure 6.4 shows the OSNR and BER for the amplifier with 20dBm saturation power and 5dB noise figure if a reconfigurable drop of 1dB loss could be employed (meaning 3dB improvement in the total loss of each node). Compared with Figure 6.2 and 6.3 under the same amplifier specifications, it is clear that reducing total loss in the node is as critical in the scalability of *HORNET* as having high-performance amplifiers.

The above discussions were based on the assumption that all the signal source

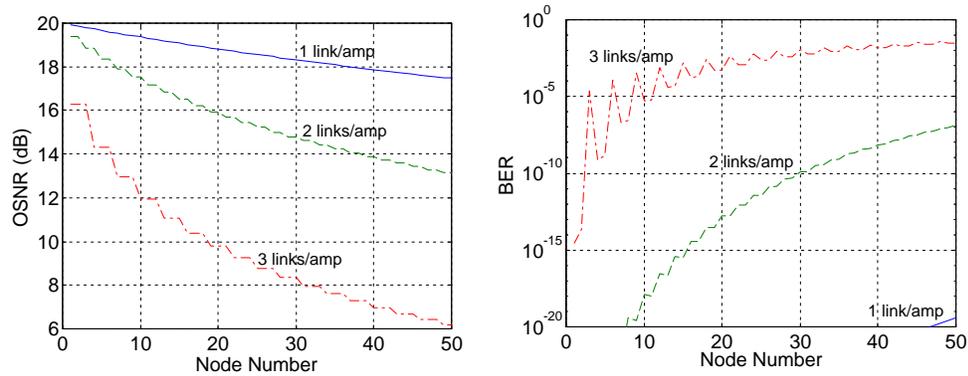


Figure 6.4: Calculated OSNR and BER versus node number for 10Gbps system if the loss of reconfigurable drops can be reduced to 1dB ($P_{sat} = 20\text{dBm}$, $NF=5\text{dB}$)

and amplifiers behave ideally, i.e. they yield exactly the power and signal gain as desired. In reality, for example, the output power of tunable lasers is hardly constant for different wavelengths, and the variations can be as much as several dB's. The power offset propagates through the fiber link but fortunately does not affect the amplifier behavior and thus is not accumulative. Since the laser output power stands for the signal power, the least output power results in the worst electrical SNR in the receiver and thus the highest BER. From a system designer's point of view, we simply have to specify a minimal laser output power among the utilized wavelengths in order to achieve the specific transmission performance.

The other main contributor that can cause deviation from the ideal case is the amplifier gain error. The amplifier gain error can be due to two different reasons: (1) Channel equalization error among different wavelengths. (2) Imperfect gain-clamping due to spectral hole burning or imperfect transient control. In both cases, the gain error can lead to power deviated from the design specifications after signals propagate through many nodes and larger (or smaller) ASE power than expected. The channel equalization error can be a more serious problem in the transparent network architecture since it tends to be accumulative if all the amplifiers are of the same

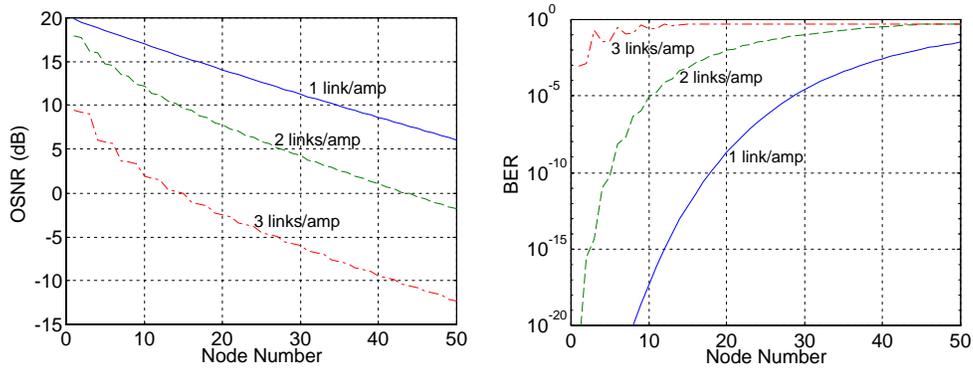


Figure 6.5: Calculated OSNR and BER versus node number for 10Gbps system with -0.2dB gain error on each amplifier ($P_{sat} = 20\text{dBm}$, $NF=5\text{dB}$)

model. In case amplifiers give larger gain than desired, it is usually not detrimental since the signal power is higher while the ASE noise is usually only slightly higher. Problems may happen if amplifiers exhibit negative gain errors accumulatively on some specific channel. In Figure 6.5 we show the case in which each amplifier on the ring has a -0.2dB gain error with 20dB amplifier gain and 5dB noise figure for a 10Gbps system. Compared to Figures 6.2 and 6.3 of similar operating conditions, we can see the number of node number can be seriously limited (within 20) in order to have a reasonable transmission performance. The effect of gain errors propagates along the ring and might have significant impact on the system if not carefully dealt with.

Chapter 7

Conclusions

7.1 Accomplishments

IP-HORNET generated a number of publications that cover a wide range of issues, from experimental testbed demonstrations, to sub-systems design, to theoretical protocol design and simulations. Following is a list of papers already published in conferences and journals, while a separate list of submitted manuscripts is also included below.

Publications:

1. L. G. Kazovsky, K. Shrikhande, I. M. White, M. Rogge and D. Wonglumsom, "Optical Metropolitan Area Networks," Optical Fiber Communication conference, Anaheim, CA, pp. WU1-1, March, 2001 (Invited paper).
2. K. Shrikhande, I. M. White, M. Rogge, F.-T. An, E. S. Hu, S. S.-H. Yam and L. G. Kazovsky, "Performance Demonstration of a Fast-Tunable Transmitter and Burst-Mode Packet Receiver for HORNET," Optical Fiber Communications conference, Anaheim, CA, pp. ThG2-1, March, 2001.
3. L. G. Kazovsky, I. M. White, K. Shrikhande and M. S. Rogge, "High Capacity Metropolitan Area Networks for the Next Generation Internet," Asilomar Conference

on Signals and Systems, Monterey, CA, p. MA1b-1, November, 2001, (Invited paper).

4. I. M. White, M. S. Rogge, Y.-L. Hsueh, K. Shrikhande and L. G. Kazovsky, "Experimental demonstration of the HORNET survivable bi-directional ring architecture," Optical Fiber Communications Conference (OFC 2002), Anaheim, CA, p. WW1, March, 2002.

5. K. S. Kim and L. G. Kazovsky, "Design and performance evaluation of scheduling algorithms for unslotted CSMA/CA with backoff MAC protocol in multiple-access WDM ring networks," JCIS 2002, Research Triangle Park, NC, USA, pp. 1303-1306, March, 2002.

6. K. S. Kim, H. Okagawa, K. Shrikhande and L. G. Kazovsky, "Unslotted Optical CSMA/CA MAC Protocol with Fairness Control in Metro WDM Ring Networks," GlobeCom 2002, Taipei, November, 2002.

7. I. M. White, M. S. Rogge, K. Shrikhande and L. G. Kazovsky, "Design of a control-channel-based media-access-control protocol for HORNET," Journal of Optical Networking, 1, pp. 460-473, December, 2002.

8. K. S. Kim and L. G. Kazovsky, "Design and performance evaluation of scheduling algorithms for unslotted CSMA/CA with backoff MAC protocol in multiple-access WDM ring networks," Information Sciences, 149/1-2, pp. 135-148, January, 2003.

Submitted papers:

1. I. M. White, K. Shrikhande, M. S. Rogge and L. G. Kazovsky, "A MAC protocol with fairness control for the HORNET metro network architecture," submitted to Computer Networks journal, 2003.

2. I. M. White, K. Shrikhande, M. Rogge, and L. G. Kazovsky, "A Summary of the HORNET Project: A Next Generation Metropolitan Area Network," submitted to Journal of Selected Areas of Communications, special issue on Optical Networks, 2003.

As this report and the many publications show, we have achieved numerous and

varied accomplishments. The major advances can be summarized into four main categories: simulation and evaluation of both higher and lower network layer issues, key subsystem (physical layer) development for next generation packet-based networks, protocol development for realizing practical, reliable metropolitan area networks, and experimental confirmation of key elements in the form of a working testbed.

1) Simulations and evaluation of issues on various network layers within *HORNET*: Network simulators were developed in-house to evaluate the special characteristics of the *HORNET* network. From a system point of view, link budgets were developed and simulated, fairness algorithms were designed and simulated, survivability, throughput, quality of service - all were developed and evaluated on in-house or third party simulation tools.

2) Key Subsystem development for next generation packet-based networks: As described earlier in this report, several key subsystems - MAC, control subsystem, Tunable Transmitter, Linear Optical Amplifier, and Asynchronous Packet Receiver, were all designed, simulated, evaluated, and tested to support practical, efficient next-generation packet-based networks.

3) Protocol Development for a reliable, practical metropolitan area networks: Key challenges often overlooked in research projects were addressed during this work. In particular, survivability was addressed, developed, implemented, and tested on *HORNET*. A novel survivability protocol suited for tomorrow's packet-based networks was demonstrated to operate accurately and efficiently. Protocols for Fairness were also developed and simulated - although they are specifically aimed at *HORNET*, the techniques apply equally well to other shared-resource networks. Quality of Service, a seemingly tough challenge for a network designed to work most efficiently with best-effort traffic, was examined, evaluated, and addressed. A reservation scheme was developed, simulated, and partially implemented to realize circuit emulation, or constant bit rate traffic over *HORNET*'s infrastructure. All of these protocols,

although designed to work with *HORNET*, have immediate relevance to other networks, both optical and wireless. In fact, much of the development was based on existing work in the networking research body.

4) Experimental Testbed: Perhaps most crucial of all the accomplishments is the experimental testbed developed to test and evaluate *HORNET*'s performance. The initial design included a plan to actually implement the network. Although this resulted in some decisions based purely on device availability (i.e. fast-tunable transmitter as opposed to fast-tunable receiver), the ultimate result was well worth the effort. Challenges unforeseen in simulations were uncovered and addressed in the process of building the *HORNET* testbed. The key subsystems, in particular, led to a much deeper understanding the technology required to make packet-based optical networks a reality.

7.2 Key Lessons Learned

Throughout the course of the project, several key issues were uncovered, explored and evaluated. The great majority of the lessons, as expected, came during the implementation of the testbed. The countless hours spent in the lab troubleshooting optical links were invaluable.

Based on the four main Accomplishments categories listed above, we can briefly summarize some of the key lessons learned:

1) Simulations and evaluation of issues on various network layers within *HORNET*: Because of the interesting nature of *HORNET*'s physical layer, we were forced to rethink the roles of the lower layers of the network. *HORNET* is fundamentally shared at the physical layer, forcing the link and MAC layers to interact quite closely with the physical layer. Although the line between MAC and Physical layer became

more blurred, we also realized the need for these layers. In particular, we saw capabilities fundamental to *HORNET*'s architecture that could only be leveraged by careful interaction between higher and lower network layers. Merged in with this, we were forced to evaluate the role of the control plane as well. In the context of Circuits Over HORNET (CoHo), for example, the higher layers requesting the circuits, the control plane reserving the resources, and the lower layers implementing the transfers all had to be tied carefully together.

2) Key Subsystem development for next generation packet-based networks: The MAC evolved over the duration of the project. It is clear to us, however, that each MAC has its advantages and shortcomings. With the final implementation in the form of a dedicated optical channel, we were forced to realize its inefficiency for networks with small numbers of wavelengths. In addition, the current MAC implementation is, in some sense, the weakest link of the network. Although the network is survivable, the information upon which each nodes bases its transmission decisions is carried on the control channel. The Tunable Transmitter subsystem has always been challenging for a number of reasons. Perhaps most fundamentally the problems arise from the devices themselves, which we cannot claim responsibility for. In other words, lasers that were designed to tune extremely quickly and reliably are quite young. As such, we expect much improvement in this subsystem in the coming years. In general, we were often forced to accept the tradeoff between faster tuning and more reliable operation. The transient nature of *HORNET*'s physical layer seemed to offer a significant challenge to the system design. However, we realized after simulation and experimental work, that the issue of gain transients in packet-based networks can be controlled. However, this is not without cost. In general, more amplifiers which have transient control (equals more noise) are required than first expected. In fact, as we learned during the Power Budget Analysis, the problem is significant enough that scalability beyond the metropolitan area is not extremely likely. *HORNET*'s physical layer requires each

node to be capable of receiving packets asynchronously. This is no small challenge at data rates of 2.5Gbps, 10Gbps, and beyond. Although several different techniques were developed, the final implementation uses an effective but less than optimal analog nonlinear technique to recover an incoming packet's clock. Again, we are somewhat limited by the devices themselves. It is feasible, and has even been demonstrated, that high speed digital circuits can rapidly acquire the frequency and phase of an incoming data stream - however, we were not able to acquire such devices and were forced to implement a less elegant scheme in the testbed.

3) Protocol Development for a reliable, practical metropolitan area networks: From the early days of *HORNET*, we were determined to produce a practical network. As we examined what types of qualities a network must have to be "accepted" as a real network, we found the simplicity of *HORNET* was sometimes misleading. In particular, survivability, fairness, and quality of service were three key points we focused on. We learned that networks must support legacy protocols, be robust, manageable, and efficient.

4) Experimental Testbed: Most of the lessons learned in dealing with the testbed cannot be documented in a report. Bringing the ideas on paper into a working testbed regularly reminded us of the vast difference between theory and practice. Perhaps most importantly, though, by building a testbed that attempts to inter-operate with other existing networks, we were forced to consider practical issues often overlooked in protocol design. For example, designing a network that can support constant bit rate circuits is only a small fraction of the challenge - implementing a network that supports this, as well as a mechanism by which to request/setup/teardown those circuits is by far more challenging.

7.3 Future Work

HORNET incorporates so many novel issues related to next generation networks, that it really is quite difficult to address them in a two year project. We have attempted to summarize just a few of the most interesting issues that should be addressed in future work on the subject. Wherever possible, the discussion is kept as general as possible.

Reconfigurability: Optical technologies that allow nodes to automatically reconfigure in a relatively short period of time will further enhance the efficiency with which resources can be shared across a common infrastructure. In the context of *HORNET*, this is fairly obvious. The current design has a rigidly defined notion of a node and how much of the ring's bandwidth it shares. If the network resources can be divided dynamically throughout the day, the efficiency and effective throughput would increase.

Flexibility in Node Design: Most network designs are either based on Centralized (server/client) or Distributed (cluster). *HORNET* is currently architected as a distributed network, with a somewhat symmetric design envisioned (nodes are approximately peers). This is in stark contrast to the current SONET design, where a large cross-connect at the point-of-presence performs most of the functions of the network. Although it is predicted that traffic in the metropolitan area will become more and more distributed, it seems apparent that the node or nodes connected to the longhaul network will be "busier" than the rest. The distributed nature of *HORNET* has several advantages, but a more flexible node design that may allow a super-node could be more efficient overall.

Interoperability: If one were to drop *HORNET* into the existing network infrastructure, its value would not be realized. In other words, current networks assume a circuit-based paradigm. At the very least, *HORNET* should be more carefully

evaluated for its interaction capabilities at both longhaul and access points.

New devices: As with any system project, the maturity of the key subsystems continues to evolve. As new capabilities arise from new devices, they should be evaluated for their usefulness in next generation networks.

Control Plane: The design and implementation of the control plane should be more fully developed. Although this is particular to *HORNET*, the problem is general as well. As networks evolve, the control plane seems to be the limiting factor in terms of interoperability. Without careful design, for example, *HORNET*'s capabilities may never be utilized by the access nodes traversing *HORNET*

7.4 Final Thoughts

The *HORNET* project has exposed and addressed numerous issues relevant to next generation networks. By pushing the technology toward more efficient, data-driven networks, while maintaining the practical requirements of a physical testbed, this project has achieved impressive results on several fronts. Our thanks go to the NGI Initiative, DARPA, and the AFRL, our sponsor.

Performance Demonstration of a Fast-Tunable Transmitter and Burst-Mode Packet Receiver for *HORNET*

K. Shrikhande, I. M. White, M. S. Rogge, F-T. An, A. Srivatsa, E. S. Hu, S. S-H. Yam, and L. G. Kazovsky

Optical Communications Research Laboratory, Stanford University (<http://ocrl.stanford.edu>)

Phone: 650.724.3409, e-mail: kapils@stanford.edu

This work has been sponsored by Sprint Advanced Technology Laboratories under contract #7063012

Abstract: We demonstrate error-free packet-over-WDM transmission using a fast-tunable transmitter and novel packet receiver. The transmitter tunes fine (0.8nm) and wide (~30nm) within 15ns, while the receiver receives unframed packets by bit-synchronizing in 40ns.

1. Introduction

HORNET (Hybrid Opto-electronic Ring Network) [1-2] is a packet-over-WDM ring metropolitan area network that uses fast-tunable transmitters and fixed wavelength, burst-mode receivers in its network Access Points (APs). This combination allows APs to transmit consecutive packets on any wavelength in the network, and therefore to any destination AP. On an architectural level, this enables *HORNET* to use a *true* packet-over-WDM stack [3-5].

This paper demonstrates the successful implementation of the *HORNET* fast-tunable packet transmitter and burst-mode receiver. We have built a scalable, practical tunable transmitter that can tune throughout the C-band, in fine (0.8nm), wide (~30nm) and intermediate hops, with a tuning time of 15ns or less. We also demonstrate a burst-mode receiver, with a clock recovery scheme that recovers the clock within 40ns. We have also performed burst-mode bit error rate (BER) tests on our fast-tunable transmitter and burst-mode receiver, functioning back-to-back. The BER tests show a constant power penalty of only 0.5dB, measured at 10^{-9} bit error rate, compared to a conventional system without laser tuning or packet clock recovery. This confirms the stability of the optical carrier after the laser has tuned. It also verifies the proper functioning of the clock recovery circuit, since it provides the input clock used by the BER tester to sample the received data.

2. Design goals

The tuning time of the transmitter should be small (tens of ns) as compared to the sub- μ s packet duration at 2.5Gb/s and 10Gb/s data rates, since the time spent tuning is wasted. This ensures low overhead. The transmitter must scale to cover the C-band and hit the desired ITU wavelengths consistently. The optical carrier should support error-free data transport, once tuning is over. We achieve these goals and yet retain a simple design to allow easy integration.

Since the transmitter may send consecutive packets on different wavelengths, a point-to-point connection cannot be maintained between source and destination, as is done in conventional networks. It is thus necessary for the receiver to synchronize itself with every incoming packet. To avoid overhead, bit-synchronization must be fast (tens of ns). Hence, conventional phase-lock loop (PLL) technology that bit-synchronizes in $\sim 1\mu$ s is inadequate. Another alternative is to use digital logic techniques, but 2.5 and 10GHz logic chips are expensive and not readily available. Hence, for *HORNET*, we use a simple, inexpensive clock recovery technique that extracts the clock frequency and phase from the incoming packet using an RF mixer.

3. Experimental Setup and Results

Fig.1 shows the testbed used to evaluate the tunable transmitter and the clock recovery circuit in the receiver. The tunable transmitter has two parts: a 4-section ADC-Altitud GCSR tunable laser and a tuning controller PCB consisting of a programmable logic device (PLD) and fast digital-to-analog converters (DACs). The laser has 3 tuning sections: a *coupler* section for coarse tuning, a *sampled-reflector* section for fine-tuning, and a *phase* section for ultra-fine tuning. The main advantage of the GCSR laser is the wide wavelength range it covers (at least 30nm) for small tuning currents (<10mA) (details in [6]). To tune the laser wide and fine and hit specific ITU wavelengths, it is necessary to control both the coupler and the reflector sections, simultaneously. We achieve this quite simply. The 'digital' tuning-current values for all wavelengths are stored on the PLD. The digital values are converted to an analog current by the DACs with 0.02mA accuracy and injected into the coupler and reflector sections of the laser. Moreover, to ensure fast tuning, the carrier density in the tuning sections must change rapidly. In [1], we have shown that overshoot/undershoot pulses on the current reduces tuning-time dramatically, by improving the rise/fall time of the signal injected into the laser and pushing/removing the carriers faster. Therefore, we also store the digital overshoot/undershoot values on the PLD. Fig. 2 shows a sample current waveform generated by the PLD + DAC set-up for one of the tuning sections. The steady-state current value holds the laser at a target mode, during the

transmission of a packet. When the laser tunes, the current switches to the next steady-state value with an overshoot/undershoot. We can alter the size and time-duration of the overshoot/undershoot (see Fig.2) because a larger current jump (or wavelength jump) requires a bigger and/or longer overshoot for fast tuning.

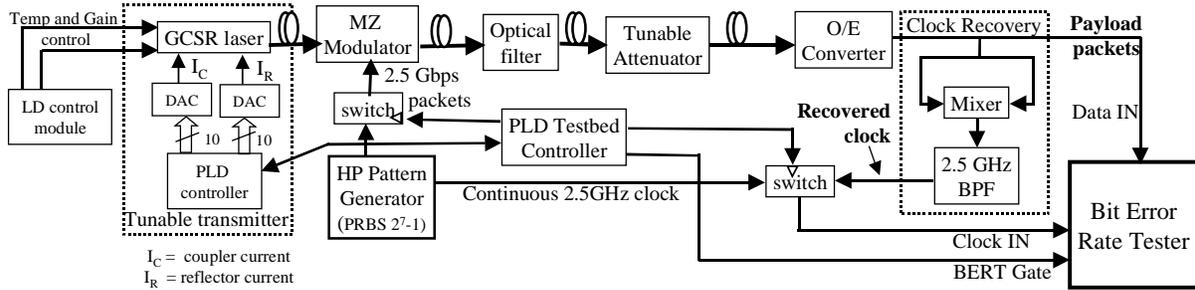


Fig. 1: Experimental setup to evaluate the HORNET fast tunable transmitter and packet receiver

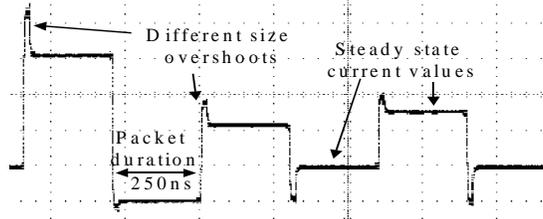


Fig. 2: A sample tuning signal applied to the laser to achieve fast-tuning

To demonstrate the capabilities of the transmitter, the tuning controller is programmed to tune the laser between six ITU wavelengths spanning 30nm, maintaining each wavelength for 250ns as if transmitting a packet, before tuning to the next wavelength. The MZ modulator at the output of the laser suppresses the optical signal while the laser tunes. Fig. 3 shows the OSA screen plot obtained by connecting the modulator output to the OSA. All six modes can be seen because the optical power is averaged over time on the OSA. The tuning sequence, shown in Fig. 3, demonstrates tuning hops ranging from 0.8nm to 28.8nm, proving the ability to hop between any combination of C-band wavelengths. The maximum tuning time measured is 15ns, which maintains the desired low overhead. We have thus built on our work in [1], to demonstrate a scalable transmitter that fast-tunes throughout the C-band.

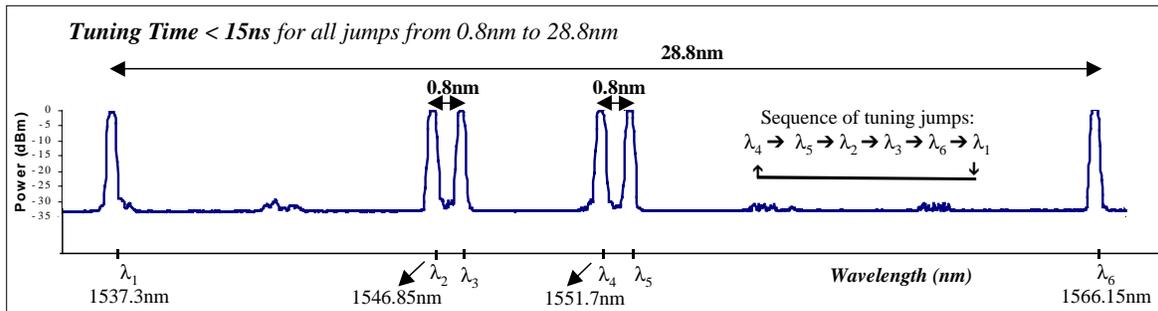


Fig. 3: Averaged optical output of the tunable transmitter while tuning between six wavelengths over ~30nm

As mentioned previously, a BER testbed is setup for performance measurements (refer Fig. 1). A PLD-based Testbed Controller manages timing and controls the various devices in the set-up. The HP Pattern Generator (PPG) outputs $2^7 - 1$ PRBS data stream at 2.5Gbps. Data packets with a time duration $< 1\mu\text{s}$ are generated by controlling the switch placed at the PPG output. This is comparable to a typical Internet packet at 2.5 Gbps. The timing is managed such that the packet arrives at the MZ modulator to modulate the optical carrier as soon as the laser is tuned. An optical filter at the output of the tunable transmitter drops a fixed wavelength. Hence, it selects packets only on the receiver's drop wavelength, to simulate a HORNET AP receiver. These packets are converted to electronic data by an O/E converter. The electrical 2.5Gbps data signal is split to feed the BERT (payload) and the clock recovery circuit. To recover the clock instantaneously, a simple yet effective circuit is used where the packetized data is placed at both inputs of a mixer (refer Fig. 1). Maximum performance is achieved with a 2cm delay on one input branch. A strong tone at the exact frequency of the payload data's bit rate is present at the output of the mixer whenever there is a packet present at the inputs. This technique is particularly effective because the phase relationship between clock and data is the same for every packet entering the receiver, so an adaptive phase

delay is not necessary. Because the data is arriving in small bursts, it is necessary to operate the BERT in ‘burst gating’ mode. A gating signal is applied to the BERT that informs it of when to inspect the data for errors. This technique is also used in recirculating loop experiments. One difficulty is that the BERT requires a continuous clock at its input. In our system, the recovered clock is present only when a packet is arriving at the receiver. Thus, a continuous clock from the PPG is ‘switched in’, whenever the recovered clock is not present, as shown in Fig. 1.

The resulting BER plot is shown in Fig. 4 (a). The line farthest to the left represents conventional systems: the transmitter is fixed (not tuned) at the drop wavelength, the PPG’s clock is used as the BERT input clock, and burst-gating mode is not used. For the second line over: data packets are generated on a fixed transmitter, the PPG clock serves as BERT input clock and burst-gating mode is used. A negligible power penalty, $< 0.2\text{dB}$, is recorded for packet transport as compared to continuous mode. Hence, from now on, we will use line 2 as our baseline. For the third line: the transmitter is tuned between each packet transmission. It transmits on the receiver’s wavelength, then on a channel 0.8nm away, then on a channel 5nm away and then back on the receiver’s wavelength. Again, negligible power penalty is incurred due to tuning between packet transmissions. This ensures the stability of the optical carrier after it has tuned to the target wavelength. The fourth line tests the performance of both the tunable transmitter and the packet receiver, back-to-back. The laser is tuned just as in the previous measurement, but now the recovered clock is used as the BERT input clock and burst-gating is used. The power penalty for entire system is $< 0.5\text{dB}$, as compared to the second line from the left, and is almost constant for a wide measurement range, without any BER floor. The small penalty exists because the recovery circuit requires additional optical power to maintain a clean clock signal at its output. Most importantly, **errorless transmission** was obtained for this setup.

Fig. 4 (b) and (c) shows the timing of the input signals to the BERT for this measurement. As the packet arrives in the receiver, the recovered clock power rises. After 40 ns , the clock is completely stable at BERT input, and thus the Testbed Controller generates the BERT gating signal, as shown in Fig. 4 (c). The recovery time of 40 ns is chosen by finding the smallest delay that still allows stable BER measurements. Thus, it is clear that a maximum of 40 ns is necessary for clock recovery. In reality this number will be likely smaller. The setup is flawed since the BERT requires a stabilization time from when the PPG clock is switched off and the recovered clock is switched on.

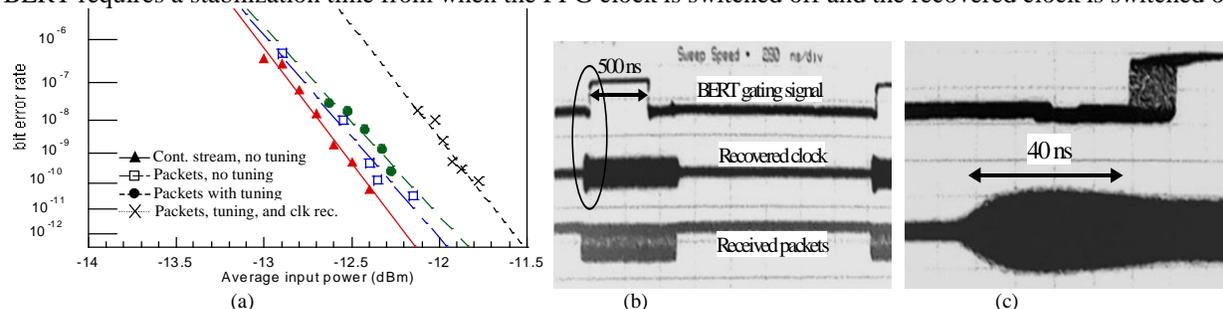


Figure 4: (a) BER plots for the experimental setup; (b) timing of the packets arriving at the receiver, the recovered clock, and the BERT gating signal; (c) zoomed in view of the BER gate and recovered clock

3. Summary

We demonstrate errorless transmission with a power penalty of only 0.5 dB for a novel tunable transmitter and a burst-mode packet receiver. The transmitter uses a PLD and fast DACs to tune a GCSR laser throughout the C-band with a maximum tuning duration of 15 ns . The receiver uses a simple RF clock recovery circuit that extracts a clock from the incoming data and uses it to recover the packet with less than 40 ns of synchronization time. This demonstration brings packet-over-WDM networks such as *HORNET* closer to a reality.

4. References

- [1]. Y. Fukashiro, K. Shrikhande, et. al., “Fast and Fine Wavelength Tuning of a GCSR Laser Using a Digitally Controlled Driver,” *Optical Fiber Communication Technical Digest*, paper WM43, pp. 338-340, Baltimore, MD, March 2000.
- [2]. I. M. White, Y. Fukashiro, et. al., “Experimental Demonstration of a Media Access Protocol for HORNET: A WDM Multiple Access Metropolitan Area Ring Network,” *Optical Fiber Communication Technical Digest*, paper WD3, pp. 50-52, Baltimore, MD, March 2000.
- [3]. J. Jackel, T. Banwell, “Burst Optical Packet Transport Over the MONET DC Network,” *ECOC*, post-deadline paper, Sept. 2000.
- [4]. D. J. Blumenthal, A. Carena, L. Rau, V. Curri, and S. Humphries, “WDM Optical IP Tag Switch with Packet-Rate Wavelength Conversion and Subcarrier Multiplexed Addressing,” *Optical Fiber Communication Technical Digest*, pp. 162-164, San Diego, CA, February 1999.
- [5]. G. K. Chang, G. Ellinas, H. Dai, B. Meagher, et al, “A Proof of Concept, Ultra-low latency optical label switching testbed demonstration for next generation Internet networks,” *Optical Fiber Communication Technical Digest*, paper WD5, pp. 56-58, Baltimore, MD, March 2000.
- [6]. P. J. Rigole, M. Shell, S. Nilsson, D.J. Blumenthal, and E. Berglund, “Fast Wavelength Switching in a Widely Tunable GCSR Laser Using a Pulse Pre-distortion Technique,” *Optical Fiber Communication Technical Digest*, pp. 231-232, Dallas, TX, February 1997.

Optical Metropolitan Area Networks

L. G. Kazovsky, K. Shrikhande, I. M. White, M. Rogge and D. Wonglumsom

Optical Communications Research Laboratory, Stanford University

Tel: 650-725-3818, Email: kazovsky@stanford.edu, Web: <http://wdm.stanford.edu>

This work has been sponsored by Sprint Advanced Technology Laboratories under contract #7063012

Abstract: This paper discusses emerging issues in the development of future optical metropolitan area networks and describes the pioneering research at Stanford University to address some of them.

1. Introduction

Optical metropolitan area networks (MANs) are evolving at a tremendous rate to satisfy the high demand and diverse needs of customers. Metro service providers face a number of challenges. MANs provide connectivity to a variety of customers and hence need to support a variety of services including IP, ATM, Frame Relay, Gigabit Ethernet and SONET (see Fig. 1). Moreover, each customer will have a different capacity and QoS requirement. MANs need to support bandwidth provisioning, with varying levels of granularity and scale the bandwidth for each user, intelligently. Additionally, any solution must be cost-effective, co-exist with POTS and provide SONET-like reliability. These issues and many others are currently being addressed in both research laboratories and industry.

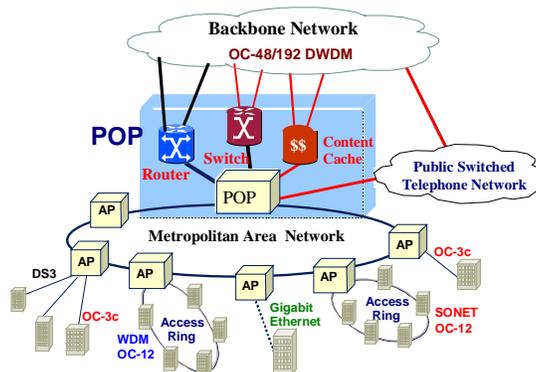


Fig. 1: Diversity in future optical MANs

As the Internet evolves, MANs will need to evolve from mere aggregation networks to more self-sustained ones. Current MANs connect LANs and campus networks to the backbone, to deliver (web) content from servers and caches located at ISP POPs across the backbone. Recently, to deliver content efficiently, content distribution networks have been proposed to deliver content and cache it closer and closer to the end user. In the future, we can expect content caches/servers to be placed at strategic points in the MAN to provide load balancing, robustness and delay guarantees for time-sensitive traffic. Additionally, with the emergence of peer-to-peer file-sharing applications, MANs will need to switch a large amount of traffic within the network.

To handle the changing needs of the future, new trends will evolve in the design of MANs: WDM layer reconfigurability; IP over WDM transport; SONET-less survivability; network-wide content distribution, storage and retrieval; and QoS, to name a few. These trends are discussed in the following sections along with a brief description of the pioneering research carried out at the Optical Communications Research Laboratory (OCRL) at Stanford University, to address some of them.

2. Networking Trends in Optical MAN Development

The metro and access segments hold the key to economic delivery of broadband services in future networks. To accomplish this, complexity and subsequently intelligence is being pushed to the edges of the network. A few key trends and our solutions are discussed below:

a) Optical Layer Reconfigurability

Optical layer reconfigurability allows a node to add/drop any WDM channel, enabling provisioning, services over wavelengths, and restoration. Our LEARN project [1] consisted of a 3-node reconfigurable ring network, over 84km of buried SMF between Sprint ATL in Burlingame and Stanford, with network-wide management. Novel

reconfigurable OADMs and an Optical Wavelength Translating Crossconnect (OWTC) were designed and built (Fig. 2 a and Fig.2 b), using a single AWG, optical switches and transponders. A node at Sprint, housing the OWTC, forms independent WDM circuits with two Stanford OADM nodes using two 1550nm wavelengths.

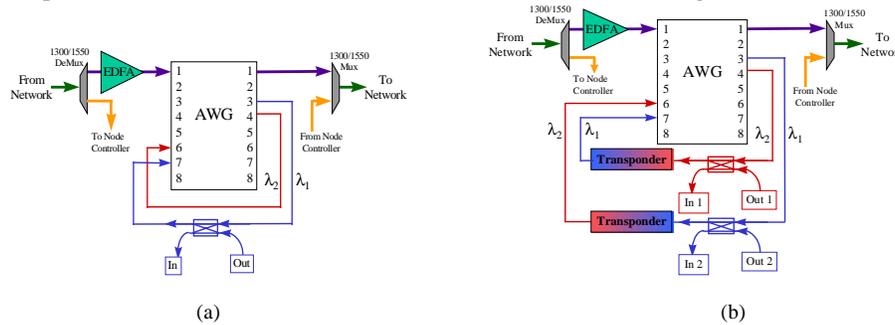


Fig. 2: (a) OADM node structure. (b) OWTC node structure.

The WDM circuits can be reconfigured by changing the switch settings within the OADM, while wavelength conversion at the Sprint node allows the Stanford nodes to communicate with each other. A 1300nm FDDI control channel is added/dropped at each node for remote management.

LEARN provides ‘point and click provisioning’ of wavelengths via a network management unit. Although necessary for circuit provisioning, manual control will prove insufficient for data-broadcast over wavelengths, content delivery and similar services where a node should be able to ‘tune’ in to a wavelength for a short period of time. To enable such dynamic reconfiguration, a *control plane* becomes necessary to manage and allocate network bandwidth fast. STARNET [2], another OCRL project, used FDDI to control and reconfigure tunable filters in a passive star network. A similar control plane is necessary in ring networks. Our solution is to build a control plane that uses IP to control WDM network elements via a standard interface. Moreover, the OADMs implemented in LEARN demultiplex all wavelengths using an AWG and use optical switches to add/drop different channels. This solution proves expensive in a MAN with a large number of wavelengths. Better scalability can be achieved by building tunable OADMs, to access a subset of the wavelengths, controlled by an IP/WDM control plane.

b) IP over WDM data-optimized transport

With the exponential increase in data traffic as opposed to voice traffic, optical MANs will be designed for IP-centric data traffic. Data-optimized MANs can take advantage of the bursty nature of IP traffic to share network resources efficiently. At the Stanford OCRL, we have designed and implemented three packet switched, IP over WDM MAN testbeds. While STARNET [2] switched packets over WDM electronically, CORD [3] demonstrated contention resolution using optical switches and delay lines for all-optical packet switching.

HORNET [4], our current MAN project uses fast tunable lasers to allow direct inter-node packet transfer. HORNET places IP packets and/or ATM cells on any wavelength, on a packet-by-packet basis, without intermediate (SONET-like) transport. This approach will scale better than current centralized switching (at the POP) networks for future MAN-centric traffic. Fig. 3 shows the design of the HORNET Access point (AP). The AP drops a fixed wavelength (Smart Drop), implements a novel Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) media access protocol (MAC) protocol to ‘listen’ to all wavelengths in parallel (Slot Manager), and switches packets onto the target wavelength using a fast tunable packet transmitter (Smart Add). APs recover packet bit clock using an embedded clock transport technique. So far, we have built an entire experimental AP and demonstrated packet-over-WDM transport consisting of the following subsystems: CSMA/CA using two wavelengths [5], fast (< 4ns) laser tuning over adjacent ITU grid wavelengths [6], and fast clock recovery (< 12 bits at 2.5 Gb/s) [4]. A novel feature of the HORNET AP is that it performs the difficult functions required for a packet over WDM transport and yet retains design simplicity. Future work includes implementing CSMA/CA for variable length IP packets and SONET-less survivability.

c) SONET-less survivability and management

An IP over WDM transport requires the restoration and management features similar to those of SONET, to be either replicated or improved upon (< 50ms restoration time). We have proposed a new Two-Fiber Bi-directional Path Switched Ring (2FBPSR) that utilizes all deployed fiber at all times and yet is fully survivable.

d) QoS

Application-level QoS is important, since it enables ISPs to give hard service guarantees (for example 10ms end-to-end delay across an ISP network) to its customers, and charge them appropriately. It is possible only if IP-QoS as defined by DiffServ and IntServ communicates with WDM layer QoS. In the meantime, coarse QoS at the WDM layer can be provided to corporate users, willing to pay for such services.

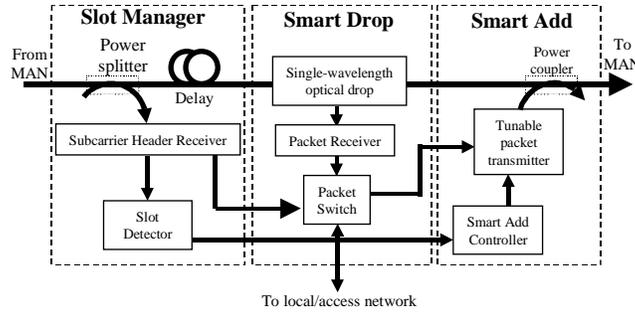


Fig. 3: HORNET Access Point design

e) Content distribution and storage over MANs

While the Internet is the ultimate tool for content delivery, accessing the content is changing dramatically. In the traditional Internet client-server model, clients retrieve data from a server by traversing the entire network in many cases, incurring delay and ISP backbone transit-tariffs. Recent solutions push content closer to the client by placing content caches at ISP POPs and use a Content Distribution Network (CDN) to update the caches with fresh data. In the future, one can expect caches to be placed at multiple locations within the MAN for load balancing, better delay guarantees for streaming media applications and for scalability. An intelligent WDM layer will be required to handle inter-cache communication such as the updating of slave caches with fresh data from the master cache and peer-to-peer cache communication.

3. Physical Layer Trends in Optical MAN Development

Although current challenges in optical MANs are related mainly to inter-networking, physical and transmission layer issues will arise and will prove to be different from their long haul counterparts. Cost is a major driving force. Directly modulated lasers, relatively coarse ($> 100\text{GHz}$) WDM and lower bit-rates help alleviate costs. New fiber types that help reduce the impact of laser chirp in directly modulated systems (such as Corning's MetroCor fiber) may be useful. Tunable elements will bring networking flexibility to MANs. Tunable DBR lasers are getting progressively cheaper and perform well [6]. What is lacking is a plug-and-play capability: each DBR laser has different tuning currents for the same set of output wavelengths, has different response to temperature, etc. Hence, the interface to tunable lasers will become important to provide plug-and-play capability. Tunable lasers based on integrated DFB LD arrays with integrated SOA and EA sections, will also be a very useful component. Tunable filters and tunable optical drops, although slow at the moment (ms or μs tuning), will be desirable for reconfiguration and restoration.

4. Summary

The optical MAN space is seeing great innovation in industry and research. Trends such as network-wide reconfiguration, IP over WDM transport, tunable OADMs and novel fibers will emerge at the networking and physical layers. Through our four MAN networking projects, the Stanford OCRL has made a significant contribution to this innovation, which will help build powerful and exciting optical MANs.

We wish to thank all current and past members of the Stanford University Optical Communications Research Lab (OCRL) for their contribution in the various projects mentioned in this paper.

References

- [1] R. T. Hofmeister, S. M. Gemelos, C. L. Lu, M. C. Ho, D. Wonglumsom, D. T. Mayweather, S. Agrawal, I. Fishman, L. G. Kazovsky, "LEARN: Lightwave Exchange Add/Drop Ring Network," *Optical Communications Conference - Postdeadline Papers* (Dallas, TX, 1997).
- [2] L. G. Kazovsky and P. T. Poggiolini, "STARNET: A Multi-gigabit-per-second Optical LAN using a Passive WDM Star", *J. of Lightwave Tech*, vol 11, num 5/6, Special Issue on Opt Net, pp.1009, June 1993
- [3] I. Chlamtac, A. Fumagalli, L. G. Kazovsky et al, "CORD: Contention resolution by delay lines", *IEEE J. of Selected Areas Comm*, vol 14, pp. 1014, June 1996.
- [4] S. M. Gemelos, K. Shrikhande, I. M. White, D. Wonglumsom, and L. G. Kazovsky, "HORNET: A packet over WDM MAN", CNIT 99
- [5] I. M. White, Y. Fukashiro, K. Shrikhande, D. Wonglumsom, M. S. Rogge, M. Avenarius, and L.G. Kazovsky, "Experimental Demonstration of a Media Access Protocol for HORNET: A WDM Multiple Access Metropolitan Area Ring Network," *OFC 2000 Technical Digest*, Baltimore, MD, paper WD3, March 2000.
- [6] Y. Fukashiro, K. Shrikhande, M. Avenarius, I. M. White, D. Wonglumsom, M. S. Rogge, and L.G. Kazovsky, "Fine and fast tuning of a GCSR tunable laser" *OFC 2000 Technical Digest*, paper WM-43, Baltimore, MD, March 2000.

High Capacity Metropolitan Area Networks for the Next Generation Internet

Leonid G. Kazovsky, Ian M. White, Kapil Shrikhande, Matt Rogge
Stanford University
Optical Communications Research Laboratory
350 Serra Mall, Packard EE Bldg., Stanford, CA 94305
tel: 650.723.3687, fax: 650.723.9251, e-mail: kazovsky@stanford.edu

Technical Area: (8) Enabling techniques for multimedia Internet services.

Abstract: New technologies for high capacity metropolitan area networks must be developed to enable future generations of high-speed multimedia Internet applications and services. This paper discusses emerging issues in the development of future optical metropolitan area networks and describes the pioneering research at Stanford University to address some of them. In particular, the HORNET project, which boasts a newly developed architecture and novel subsystems for next generation high capacity metropolitan networks, is featured in this paper.

Summary

Next generation Internet research has been focused for the last several years on increasing capacity in the long-haul backbone networks. However, to enable end users to receive high-bandwidth multimedia Internet services, the entire path from source to destination must consist of high-capacity links and network elements. This includes the long-haul Internet backbone, the metropolitan area networks, and the access networks (e.g. fiber-to-the-home networks). The backbone networks have been researched so thoroughly that the most recent results boast of greater than 10 Tbps links [1],[2]. Meanwhile, the optical access networks are poised to be a hot topic for researchers and the industry in the very near future. Today, the key to enabling high-bandwidth multimedia Internet services is research into high capacity metropolitan area networks (MANs).

MANs differ from long-haul backbone networks mainly in the approach that must be taken at tackling the bandwidth problem. The solution in the backbone has been simply to increase the size of the 'pipe' between switches or routers. However, the metro area environment is orders of magnitude more competitive than the long-haul environment, so throwing capacity at the problem is not enough. It is important to increase capacity while maintaining the necessary efficiency required to remain competitive, and while supporting a multitude of services.

Metro service providers face a number of challenges. MANs provide connectivity to a variety of customers and hence need to support a variety of services including IP, ATM, Frame Relay, Gigabit Ethernet and SONET (see Fig. 1). Moreover, each customer will have a different capacity and QoS requirement. MANs need to support bandwidth provisioning, with varying levels of granularity and intelligently scale the bandwidth for

each user. Additionally, any solution must be cost-effective, co-exist with legacy technologies, and provide SONET-like reliability.

These issues and many others are currently being addressed in both research laboratories and industry. As the Internet evolves, MANs will need to evolve from mere aggregation networks to more self-sustained ones. Current MANs connect LANs and campus networks to the backbone to deliver Internet content from servers and caches located at ISP points-of-presence (POPs) across the backbone. Recently, to deliver content efficiently, content distribution networks have been proposed to deliver content and to cache it closer to the end user. In the future, one can expect content caches/servers to be placed at strategic points in the MAN to provide load balancing, robustness and delay guarantees for time-sensitive traffic. Additionally, with the emergence of peer-to-peer file-sharing applications, MANs will need to switch a large amount of traffic within the network. To handle the changing needs of the future, new trends will evolve in the design of MANs: WDM layer reconfigurability; IP-over-WDM transport; SONET-less survivability; network-wide content distribution, storage and retrieval; and QoS, to name a few.

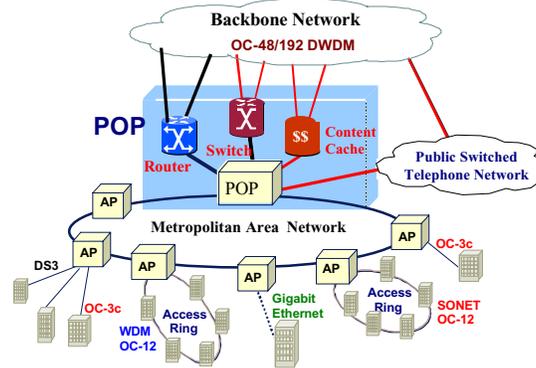


Fig. 1: Diversity in future optical MANs

For approximately a decade, the Optical Communications Research Laboratory (OCRL) at Stanford University has been working to advance technology in optical networking, particularly in the metro area. Projects such as CORD [3] and STARNET [4] have contributed to the current state of optical networking research, while results from the LEARN [5] project have impacted the latest generation of optical communications systems. The project currently underway at the OCRL is called HORNET [6], for Hybrid Opto-electronic Ring Network. HORNET uses ultra-fast tunable transmitters, novel packet receivers, and a newly developed media access control protocol to enable the network to send optical IP packets directly from the source node to the destination node. The architecture makes wise use of statistical multiplexing, which is in contrast to today's typical time-division multiplexing (TDM) architectures. In this way, HORNET is far more optimized for tomorrow's data-dominated networks than are today's TDM architectures, which were developed for yesterday's circuit switching applications.

The two most interesting new technologies investigated in HORNET are a fast-tunable optical WDM transmitter and a fast bit-synchronizing burst-mode packet receiver. The tunable transmitter is used to enable a node to transmit on any network

wavelength it chooses, allowing it to transmit a packet directly (at the optical layer) to any node on the network. It is crucial that the tunable transmitter has the ability to tune from one wavelength to another almost instantly because the transmitter will tune between consecutive packet transmissions. On 2.5 Gbps and 10 Gbps networks, typical IP packets can be less than 1 microsecond. To keep overhead low, the transmitter must have a tuning duration that is very small compared to a packet duration (i.e. only a few nanoseconds). Using currently available tunable semiconductor lasers, the OCRL designed and constructed a fast-tunable transmitter that does exactly this.

The OCRL was able to construct and demonstrate a transmitter that could tune throughout the conventional telecom band (C-band) with a *maximum* tuning duration of approximately 15 nanoseconds. The transmitter consists of a simple programmable logic processor, two digital-to-analog converters (DACs), and a tunable semiconductor laser (see Figure 2). The processor selects the wavelength, sends 10-bit digital values to the DACs, and the DACs convert the digital values to the appropriate electronic currents to tune the laser. Ultimately, the laser tuning time can be decreased if the package of the laser is optimized for high-speed currents at the tuning inputs.

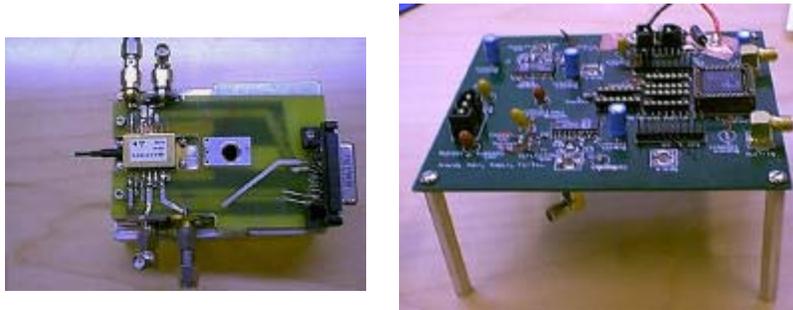


Figure 2: The laser and tuning circuit for the fast-tunable transmitter.

The HORNET project will continue until at least 2002, at which time it is expected that a complete testbed will be operating. The testbed will verify the performance of the subsystems designed and built for HORNET, and will allow the OCRL to experimentally analyze protocols as well. Similar to the previous projects, it is expected that the subsystems, protocols, and the architecture developed in HORNET will strongly influence the direction of optical networking research and will ultimately find their way into commercially deployed systems.

1. K. Fukuchi, T. Kasamatsu, M. Morie, R. Ohira, T. Ito, K. Sekiya, D. Ogasahara, T. Ono, "10.92 Tb/s (273 x 40 Gb/s) triple-band/ultra-dense WDM optical-repeated transmission experiment," *Optical Fiber Communication Conference Post Deadline Papers*, Anaheim, CA, February 2001, paper PD24.
2. S. Bigo, Y. Frignac, G. Charlet, S. Borne, P. Tran, C. Simonneau, D. Bayart, A. Jourdan, J-P. Hamaide, W. Idler, R. Dischler, G. Veith, W. Poehlmann, "10.2 Tbit/s (256 x 42.7 Gbit/s PDM/WDM) transmission over 100 km TeraLight fiber with 1.28 bit/s/Hz spectral efficiency," *Optical Fiber Communication Conference Post Deadline Papers*, Anaheim, CA, February 2001, paper PD25.
3. I. Chlamtac, A. Fumagalli, L. G. Kazovsky et al, "CORD: Contention resolution by delay lines", *IEEE Journal of Selected Areas in Communications*, vol. 14, pp. 1014, June 1996.
4. L. G. Kazovsky and P. T. Poggiolini, "STARNET: A Multi-gigabit-per-second Optical LAN using a Passive WDM Star", *IEEE Journal of Lightwave Technology*, vol. 11, num 5/6, Special Issue on Opt Net, pp.1009, June 1993.

5. R. T. Hofmeister, S. M. Gemelos, C. L. Lu, M. C. Ho, D. Wonglumsom, D. T. Mayweather, S. Agrawal, I. Fishman, L. G. Kazovsky, "LEARN: Lightwave Exchange Add/Drop Ring Network," *Optical Fiber Communication Conference Post Deadline Papers*, Dallas, TX, 1997.
6. Duang-rudee Wonglumsom, Ian M. White, Kapil Shrikhande, Matthew S. Rogge, Steven M. Gemelos, Fu-Tai An, Yasuyuki Fukashiro, Moritz Avenarius, and Leonid Kazovsky, "Experimental Demonstration of an Access Point for HORNET - A Packet-Over- WDM Multiple Access MAN," *IEEE Journal of Lightwave Technology*, vol. 18, no. 12, pp. 1709-1717.

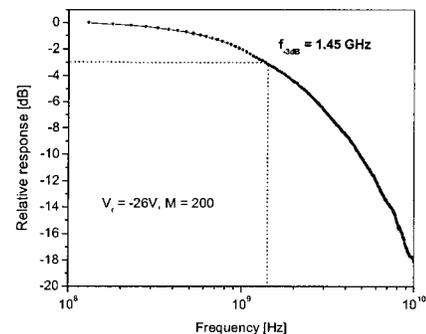
given bias, divided by the same value at -5 V (where $M = 1$). The gain is close to unity up to -20 V, and increases rapidly at higher voltages. High gains in the hundreds can easily be achieved with our silicon-based APD. The dark current is only 79 nA at $M = 10$, 150 nA at $M = 30$, and 260 nA at $M = 50$. Those values are low despite the relatively large mesa diameter of 120 μm . The dark current density is only 0.7 mA/cm² at $M = 10$. This is almost four orders of magnitude smaller than previously reported on InGaAs-on-silicon APDs,⁴ and is half the value reported for InP-based APDs.⁷

3.2 Bandwidth

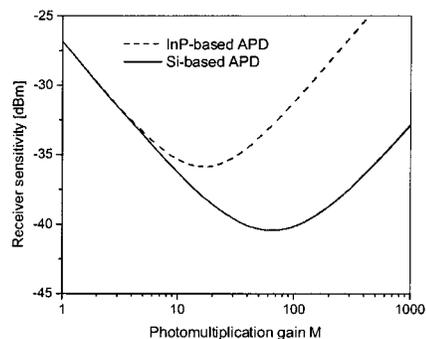
The bandwidth of the InGaAs-on-silicon APD is shown in Fig. 4 for a reverse bias of -26.3 V. It corresponds to a DC photomultiplication gain $M = 200$. The 3 dB bandwidth is 1.45 GHz, limited primarily by RC roll off due to large device capacitance. This capacitance is mainly due to Zn-diffusion from the p contact into the InGaAs absorption layer during material growth, as observed from SIMS measurements. The gain-bandwidth product is estimated to be 290 GHz.

3.3 Estimated receiver sensitivity

The expected receiver sensitivity using our Si-based APD is compared with sensitivities achieved using a conventional InP-based APD. The characteristics of the transimpedance amplifier were taken from¹: a total rms equivalent input noise current density of 250 nA/Hz^{0.5}, a bandwidth of 1.7 GHz and a bit error rate of $3 \cdot 10^{-11}$. We assume an impact ionization coefficient ratio k of 0.5 for InP, and of 0.04 for Si. The excess noise factor is calculated using McIntyre's equation,⁸ and the receiver sensitivity is determined.⁹ The calculated receiver sensitivity is shown in Fig. 5 as a function of the avalanche gain M . The InP-based APD receiver reaches a theoretic maximum



WW7 Fig. 4. Measured frequency response.



WW7 Fig. 5. Calculated receiver sensitivity.

sensitivity of -36 dBm at an optimal gain $M = 12$, in good agreement.¹ The sensitivity of the Si-based APD receiver on the other hand reaches -41 dBm at an optimal gain $M = 70$. An increase of 5 dB in receiver sensitivity represents a significant improvement in optical signal reach, reducing greatly the cost of fiber links. That increase is especially advantageous in the 1.3 μm wavelength region where no low-noise optical preamplifiers exist.

4. Conclusion

We have demonstrated a high-performance InGaAs-on-silicon APD that exhibits a very low dark current density of 0.7 mA/cm², high avalanche gain ($M = 100$), an RC-limited bandwidth of 1.45 GHz, and a gain-bandwidth product of 290 GHz. We estimate that our device can achieve a sensitivity improvement of 5 dB compared to state-of-the-art InP-based APD receivers. We are currently measuring the APD excess noise factor. We will report this measurement at the conference.

This work is partially sponsored by the US Air Force Research Lab at Hanscomb (Dual Use Science and Technology Program).

References

1. Data sheet, Agere Systems, Inc., 1319-Type High Speed Lightwave Receiver, July 2000.
2. Data sheet, JDS Uniphase Corp., ERM 578BKX 10 Gb/s APD receiver module, October 2000.
3. Data sheet, Fujitsu Compound Semiconductor, FRM5N14DS InGaAs-APD/Preamplifier Receiver, December 1999.
4. A. Hawkins *et al.*, Appl. Phys. Lett. 70(3), Jan. 1997.
5. Y.C. Zhou *et al.*, Appl. Phys. Lett., 73(16), p. 2337, 1998.
6. Y. Kang *et al.*, to be presented at LEOS '01, San Diego, Nov. 2001.
7. W.R. Clark *et al.*, Optical Fiber Communication Conference (OFC '99), 1999.
8. R.J. McIntyre, IEEE Trans. Electron. Devices, ED-13 (1996) 164-168.
9. G.P. Agrawal, "Fiber-optic communication system", Wiley, 1997.

WW

4:00 pm–6:00 pm

304A–D

Metro and Access Networks

Mark D. Feuer, JDS Uniphase, USA, *President*

WW1

4:00 pm

Experimental Demonstration of the HORNET Survivable Bi-directional Ring Architecture

Ian M. White, Matthew S. Rogge, Yu-Li Hsueh, Kapil Shrikhande, Leonid G. Kazovsky, *Stanford University Optical Communications Research Laboratory, Stanford University, Email: ianwhite@stanford.edu; Sponsored by the Defense Advanced Research Projects Agency, Contract #F30602-00-2-0544*

1. Introduction

The Internet is rapidly becoming the most pervasive medium for communications in many parts

of the world. Nonetheless, the physical layer of the Internet is constructed of young technologies that will continue to evolve over many generations. Currently, metropolitan area networking is poised to become a very quickly developing field. In today's metro networking architectures, a node connects to only a few other nodes, or possibly only to the point-of-presence (POP) node, with static point-to-point links. These architectures assume that metro networks are used only as *collection and distribution hubs* for traffic to and from local networks. However, due to distributed content and applications, the common presence of storage area networks, and an increase in packet-based wireless and multi-media applications, we believe that traffic between nodes within the metro area will become the majority of the traffic. Conventional point-to-point metro architectures are *not* optimized for this scenario.

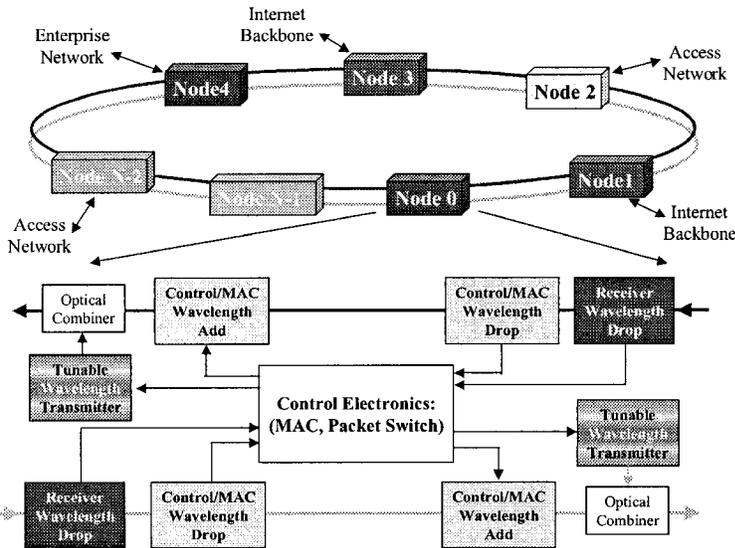
The HORNET architecture is *optimized* for future metro area traffic. Each access node on the ring network contains fast-tunable packet transmitters that can send packets into the network on any wavelength.¹ This enables every node to communicate *directly* with all other nodes on the wavelength-routed network over a completely optical path, which is a perfect architecture for highly distributed *intra-network* traffic. The fast-tunable packet transmitter has been demonstrated as a feasible subsystem in recent years.^{2,3}

Figure 1 depicts the concept of the HORNET architecture. Because an access node can insert traffic on any wavelength, and because traffic passing through the node is not electronically buffered, a media access control (MAC) protocol is required. Originally, a MAC protocol for fixed-sized cells was developed for HORNET and demonstrated.⁴ Recently, an improved MAC protocol optimized for variable-sized IP packets has been developed and implemented. In this protocol, a wavelength carrying control information is terminated at each access node where it is examined, modified as necessary, and re-transmitted. This *control channel* has a framed format, where the duration of each frame is called a 'time slot.' The control channel conveys wavelength availability information during a time slot and carries control messages between nodes, such as *fiber cut/repair messages*.

2. HORNET Survivable Bi-directional Ring Architecture

Typical 2-fiber ring architectures employ one of two strategies to maintain survivability. They either only transmit in one of the fibers, or they only use half of the potential capacity of each fiber. Both architectures require that one-half of the potential bandwidth is *not used* so that when a cut occurs, the available half can be used to restore links affected by the cut. However, with the demand for bandwidth continuing to grow, with the difficulty in burying more fiber in the metro area, and with the high cost of space in central offices, every bit of potential bandwidth should be used whenever possible.

In the HORNET 2-fiber bi-directional path-switched ring (2FBPSR) network, all of an access node's transmission capacity is used for working traffic. In general, this enables twice the throughput of the conventional network architectures (neglecting HORNET's inherent advantage of being able to dynamically adapt to traffic variations). In the HORNET bi-directional architecture, two paths exist between any two nodes.



WW1 Fig. 1. The HORNET architecture, featuring fast-tunable packet transmitters.

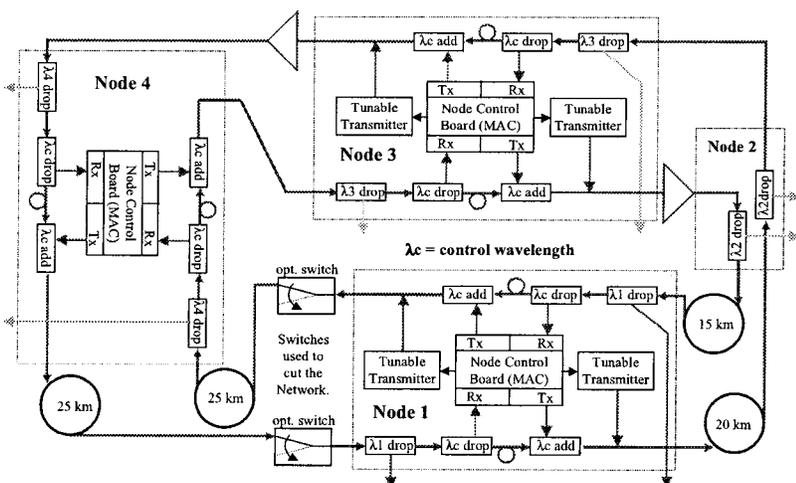
Under normal conditions, when an access node has a packet to send, it chooses the transmitter that will send the packet along the better of the two paths, as determined by a simple routing algorithm. When a cut occurs, only one of the paths remains to each destination, and thus the node is forced to use that path. The path switch occurs logically inside the node's control and routing electronics, as opposed to physically with an optical switch, as is done in many architectures. This ensures fast, reliable path switching in the event of a cut.

When a cut occurs in a conventional network, the transmission capacity of each node is unaffected because all links are fully protected. In HORNET's architecture, the effect the cut has on transmission capacity of a particular node is location dependent. For nodes far away from the cut, the transmission capacity is in general unaffected.⁵ However, nodes closer to the cut are more affected. The extreme case is the node adjacent to the cut, which only has the use of one fiber for all of its transmitted data. This in general reduces its available capacity by one half, bringing it down to

the same capacity as an access node in a conventional network (again, neglecting HORNET's inherent advantage of being able to dynamically adapt to traffic variations). This implies that the HORNET architecture can guarantee to its users the maximum capacity of a conventional network, while using up to 100% more transmission capacity for best-effort traffic, which of course is the most common traffic on the Internet today. Since cuts rarely occur, HORNET has essentially twice the capacity of conventional architectures.

3. Experimental Demonstration of the HORNET 2FBPSR Network

Figure 2 shows the testbed that was constructed to demonstrate the HORNET 2FBPSR architecture. It contains four nodes, two of which have tunable transmitters and control electronics, one of which is for control only, and one of which is only used to drop wavelengths. Spools of fiber cable are inserted so that propagation delays are present in the testbed. Because the testbed has only a few nodes, relatively large spools are used to give realistic propagation delays.



WW1 Fig. 2. The HORNET experimental testbed.

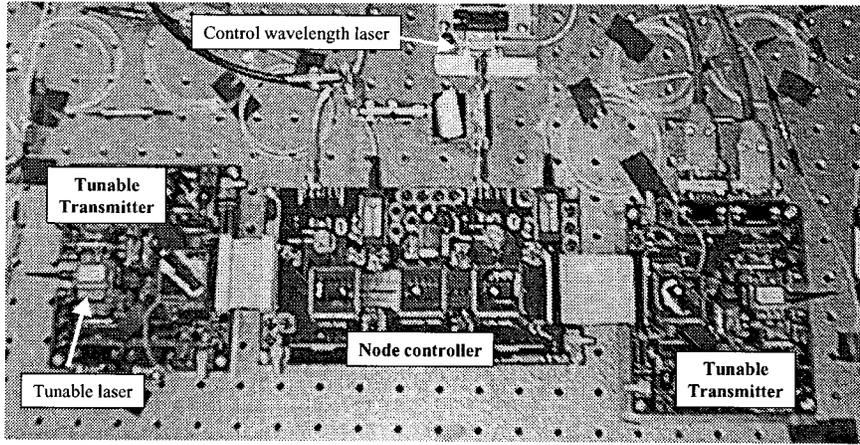
Since there are four nodes on the network, each node sends packets to three other nodes. Under normal operation, the nodes send packets to two of the destinations using the counter-clockwise (CCW) ring, and to the other destination using the clockwise (CW) ring. If a cut occurs in the ring, the nodes adjust the paths as necessary. The tunable transmitters send packets 200 ns in duration on alternating wavelengths (i.e., to alternating destinations) while using the MAC protocol to avoid collisions.

A photograph of the electronics in a node is shown in Figure 3. Nodes 1 and 3 each have a node controller circuit board and two tunable-transmitter boards. The node controller receives and re-transmits the control channel. It inspects the control channel for any messages from other nodes and for the wavelength availability information for the MAC protocol. It also runs a synchronization protocol at the startup of the network. Once the network is synchronized, if the controller detects an interruption in the control channel, it assumes that a cut has occurred on the fiber cable between it and its upstream neighbor. It readjusts its routing information and inserts a message onto the control channel. When the other controllers see this message, they readjust their routes and relay the message onto the next node. Readjusting the routes is accomplished by calculating which destinations are on which side of the cut, a simple modulo subtraction operation.

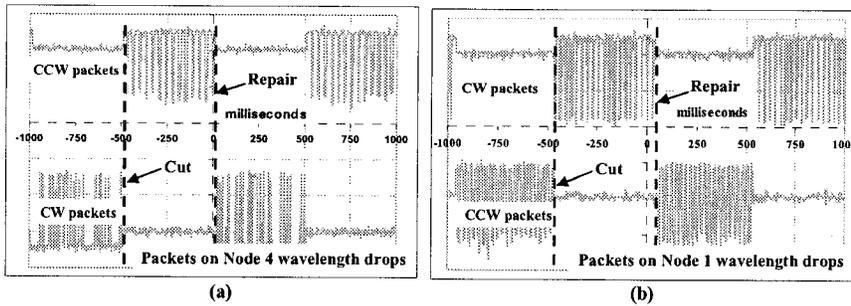
The node's protocols (startup and synchronization, MAC, survivability) are implemented in programmable logic devices (PLDs) on the control board clocked at 125 MHz. A Gigabit Ethernet chip set is used for the transmission and reception of the control channel in the testbed. Since the testbed protocols can be implemented in PLDs, it is clear that more complex components that are typically used in commercial networking equipment can handle practically scaled versions of the protocols.

To force the network to find a cut and to restore the broken paths, two optical switches controlled by a function generator periodically cut and repair the ring between Nodes 1 and 4, as shown in Figure 2. When the cut occurs, Node 1 needs to use the CCW ring to reach Node 4 because the CW ring has been cut between the two nodes. Figure 4(a) shows this result. The packets dropped by each of the WDM filters in Node 4 are detected by two APD detectors and viewed on an oscilloscope. As the figure shows, when the cut occurs, Node 4 stops receiving packets from Node 1 on the CW ring and begins receiving packets from Node 1 on the CCW ring. When the cut is fixed, Node 1 stops sending packets in the CCW ring and begins sending them again in the CW ring. The transmitters in Node 3 are disconnected from the ring during this observation so that only the packets from Node 1 are observed at Node 4. Please note that DC blocks in the receiver cause the signal level to drift, and that the oscilloscope cannot sample fast enough to avoid aliasing. This explains the irregular appearance of the waveforms in Figure 4.

When the cut occurs, both Node 1 and Node 4 detect it. Both send control messages around the ring away from the cut notifying other nodes. Node 3 then receives the message and adjusts its routes. Figure 4(b) shows the occurrence from the perspective of Node 1. It originally receives packets from Node 3 on the CCW ring. However,



WW1 Fig. 3. Photograph of the electronics in a HORNET testbed node.

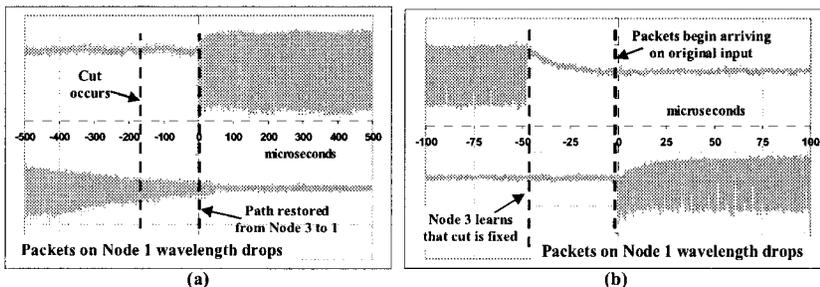


WW1 Fig. 4. (a) Packets transmitted to Node 4 from Node 1. After a cut, Node 1 must send packets to Node 4 in the CCW direction. (b) Packets transmitted from Node 3 to Node 1.

after the cut, Node 3 is forced to use the CW ring. When it learns that the cut is repaired, it resumes transmitting in the CCW direction.

Because HORNET *does not* use point-to-point link-based protocols, no setup time is required to begin using a new path. Thus, the restoration of a path happens nearly instantly. The only cause for downtime between two nodes is the propagation delay of the control messages around the ring. Figure 5 shows a zoomed-in view of the cut event and the repair event from the perspective of Node 1 as it receives packets from Node 3. When the cut occurs at the input to Node 1, the packets from Node 3 are no longer received on that path. Node 1 immediately sends a control message around the CCW ring. After propagating through approximately 20 km, the message reaches Node 3.

Node 3 then stops sending packets to Node 1 in the CCW direction and instead sends them in the CW direction. The first packet in the CW direction must propagate through 15 km of fiber before reaching Node 1. Since the cut message propagates through 20 km of fiber and the first packet in the restoration path propagates through 15 km of fiber, the path restoration process is delayed by 35 km of fiber. Since light travels through 1 km of fiber in 5 ns, there is approximately 175 ns of delay between the moment that Node 1 receives the last packet from Node 3 in the CCW direction and the moment when Node 1 receives the first packet from Node 3 in the restored path. Because of the slow rise and fall time of the optical switch, the precise time at which Node 1 determines that the link has been cut is difficult to decipher.



WW1 Fig. 5. (a) Restoration delay for the path from Node 3 to Node 1; (b) Transition of routes in Node 3 after the cut is reported as fixed (delay is only due to differences in fiber length along paths).

Nonetheless, it is clear from Figure 5(a) that there is approximately 175 ns between the time of the cut and the time when the first packet arrives along the restored path. This means that Node 3 was unable to successfully send packets to Node 1 for approximately 175 ns.

Figure 5(b) shows the events in the receivers of Node 1 when the cut is repaired. When Node 3 receives the message that the cut is repaired, it stops sending packets to Node 1 in the CW direction and begins sending them in the original direction. The final packet sent in the CW direction travels 15 km before reaching Node 1, while the first packet in the CCW direction travels 25 km before reaching Node 1. Thus there is 50 ns of delay between the last CW packet and the first CCW packet, as can be observed in Figure 5(b). This does not correspond to time during which Node 3 is unable to reach Node 1. It is only a result of the difference in path lengths. If the CCW path were shorter, there would be some overlap. This implies that care should be taken in practical networks to avoid a temporary mis-ordering of packets when a cut is repaired. This can easily be accomplished by waiting a fixed, pre-determined amount of time before switching back to the original transmission direction.

4. Summary

In this work we demonstrate the HORNET 2FBPSR survivable architecture and the associated protocols. With simple intuition, we can see that this architecture performs up to twice as well as that of conventional ring architectures under normal conditions, and in the worst case the performance is equal after a cut occurs. We experimentally demonstrate that when a cut occurs, the maximum amount of time that any two nodes lose the ability to communicate is equal to the propagation delay of the fiber between them (less than 1 ms for a typical metro ring architecture). The demonstration of the HORNET testbed presented here is a major step towards making next generation high capacity packet-based MANs a reality.

References

1. Duang-shudee Wonglumsom, Ian M. White, Kapil Shrikhande, Matthew S. Rogge, Steven M. Gemelos, Fu-Tai An, Yasuyuki Fukushima, Moritz Avenarius, and Leonid Kazovsky, "Experimental Demonstration of an Access Point for HORNET—A Packet-Over-WDM Multiple Access MAN," *IEEE Journal of Lightwave Technology*, vol. 18, no. 12, pp. 1709–1717.
2. K. Shrikhande, I.M. White, M.S. Rogge, F-T. An, A. Srivatsa, E.S. Hu, S. S-H. Yam, and L.G. Kazovsky, "Performance Demonstration of a Fast-Tunable Transmitter and Burst-Mode Packet Receiver for HORNET," *Optical Fiber Communication Technical Digest*, Anaheim, CA, February 2001.
3. O.A. Lavrova, D.J. Blumenthal, "Rapid Tunable Transmitter with Large Number of ITU Channels Accessible in Less Than 5 ns," *Proceedings of the 26th European Conference on Optical Communication (ECOC '00)*, Munich, Germany, Paper 6.3.5, pp. 23–24, September 4–7, 2000.
4. I.M. White, M.S. Rogge, K. Shrikhande, Y. Fukushima, D. Wonglumsom, F-T. An, and L.G. Kazovsky, "Experimental Demonstration of a Novel Media Access Protocol for HORNET: A Packet-Over-WDM Multiple

Design and Performance Evaluation of Scheduling Algorithms for Unslotted CSMA/CA with Backoff MAC Protocol in Multiple-Access WDM Ring Networks

Kyeong Soo Kim and Leonid G. Kazovsky

Stanford Networking Research Center, Packard Building, Room 073, Stanford, CA 94305.

Abstract—The unslotted *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) with backoff is a fully distributed, asynchronous *Media Access Control* (MAC) protocol for multiple-access *Wavelength Division Multiplexing* (WDM) ring networks with simplicity and robustness comparable to those of Ethernet [1], [2]. In this paper we present the results of performance evaluation of four scheduling algorithms — *Random Select* (RS), *Destination Priority Queuing* (DPQ), *Longest Queue First* (LQF), and *Shortest Packet First* (SPF) — designed for the unslotted CSMA/CA with backoff MAC protocol to address the issues of fairness and bandwidth efficiency. Through extensive network-level simulations for a multiple-access WDM ring with 10 nodes and 10 wavelengths on a 100 km ring, we have verified that under uniform traffic condition, the LQF shows the best performance in terms of throughput and fairness, while for delay, the DPQ shows the best results. We have also identified that the optical buffer size greatly affects the performance of the scheduling algorithms.

Keywords—Scheduling, Unslotted CSMA/CA with Backoff, MAC, RS, LQF, DPQ, SPF, WDM, Ring Networks

I. INTRODUCTION

Transmission of *Internet Protocol* (IP) packets over *Wavelength Division Multiplexing* (WDM) layer has been gathering tremendous interest among the optical networking community due to its simplicity and low overhead, resulting from the elimination of intermediate layers like *Asynchronous Transfer Mode* (ATM) and *Synchronous Optical NETWORK* (SONET). Among various network architectures available for the IP over WDM, the multiple-access ring is considered one of the most promising and economical network architectures for future optical *Metropolitan Area Networks* (MANs). In the multiple-access ring architecture, it is essential to design *Media Access Control* (MAC) protocols that are efficient in allocating bandwidth with guaranteed fair access to all nodes on the ring.

Unslotted *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) with backoff has been proposed as one of MAC protocols for IP-HORNET — IP version of *Hybrid Optoelectronic Ring NETWORK* [1], [2]. The unslotted CSMA/CA with backoff has two unique features as an optical MAC protocol: First, it is a fully distributed, asynchronous protocol that doesn't need a centralized controller or a separate control wavelength to harmonize and synchronize the operations of nodes on a ring. Second, it can naturally support variable length IP packets without

complicated segmentation and reassembly, which becomes harder as the line speed of optical wavelengths ever increases.

These features make the unslotted CSMA/CA with backoff MAC protocol very simple and scalable. For actual implementation of the protocol, however, important issues including fairness scheduling and effects of implementation parameters like optical buffer size are to be fully investigated.

Especially, design of fair and efficient scheduling algorithms is critical due to the inherent unfairness in the multiple-access optical ring (or bus) network. Because of unidirectional transmission of signal on the optical ring, the incoming frames from upstream nodes take priority over outgoing frames at a node. Hence, there arises the so-called *positional priority* problem where for a given destination and the corresponding wavelength, access nodes far from the destination node have higher priorities over those closer to destination node. Therefore without proper scheduling that counteracts this unfairness, the experienced quality of service of a connection at a node is highly dependent upon the relative position of the node with respect to its destination. In addition to fairness guarantee, scheduling algorithms should be efficient in use of available bandwidth, which means they should provide good overall throughput.

In this paper we report the design of scheduling algorithms for the unslotted CSMA/CA with backoff MAC protocol to address the issues of fairness and bandwidth efficiency in the multiple-access ring network and the results of performance evaluation through extensive network-level simulations. We also investigate the effect of optical buffer size on the performance of the scheduling algorithms, the buffer size being one of the critical implementation parameters.

The rest of the paper is organized as follows: We first review the unslotted CSMA/CA with backoff MAC protocol in Section II and describe the scheduling algorithms designed in Section III. In Section IV we present simulation results with discussions. Section V summarizes our work.

II. UNSLOTTED CSMA/CA WITH BACKOFF MAC PROTOCOL

Carrier sense and collision avoidance operations are depicted in Fig. 1. The access node listens to all wavelengths

by monitoring either sub-carriers [1] or baseband optical signals [3], depending on the implementation. When a frame is ready for transmission, the access node checks the occupancy of the target wavelength. If it is free at that instant, the access node begins to transmit the frame. However, since the access node cannot know if the opening is long enough to accommodate the entire frame, it continues to monitor the wavelength. A small ‘fixed’ optical delay line (*i.e.*, optical buffer) is placed between the point at which the node listens for incoming frames and the point at which the node inserts new frames. This allows the node to terminate its transmission before the frame interferes with the frame already on the ring. If it detects a frame arriving on the same wavelength at its input and the size of optical buffer is not big enough for successful transmission of the remaining frame with a guard band, it immediately interrupts the frame transmission and sends a jamming signal. Otherwise, it can finish the transmission of the entire frame without interruption. Note that the optical buffer size at least should be large enough for transmitting the jamming signal and the guard band before the incoming frame. The jamming signal (like in Ethernet 10/100 Base-T) could be a unique bit pattern, either at baseband or on the sub-carrier. The downstream access node recognizes the incomplete frame by the presence of the jamming signal and pulls it off the ring. The access node can reschedule the transmission of the frame for a later time.

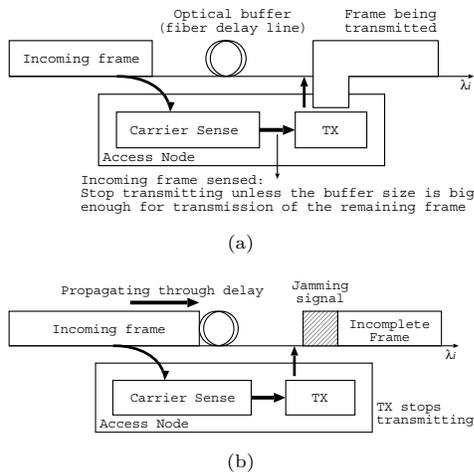


Fig. 1. Unslotted CSMA/CA with backoff: (a) Carrier sense; (b) collision avoidance.

III. SCHEDULING ALGORITHMS FOR UNSLOTTED CSMA/CA WITH BACKOFF MAC PROTOCOL

A. Random Selection (RS) Scheduling

The MAC implementation has a *Virtual Output Queue* (VOQ) for each wavelength. The RS algorithm maintains a list of empty wavelengths and corresponding “*non-empty*” VOQs. It then randomly selects a VOQ out of the list for transmission. This scheme is fairly simple and has no counter-measure for the unfairness, but we use it as a reference algorithm in performance evaluation of scheduling

algorithms.

B. Longest Queue First (LQF) Scheduling

Because of the positional priority, VOQs for those wavelengths whose destinations are closer downstream are likely to have more frames than others. We counteract this problem by giving priorities to those wavelengths with longer VOQs to guarantee fairness. In the LQF scheduling, if wavelengths are available for transmission, the scheduler selects the one with longest VOQ among them.

C. Destination Priority Queueing (DPQ) Scheduling

The DPQ scheduling algorithm tries to achieve the fairness by giving priorities to wavelengths based on their destinations rather than based on the length of VOQs. In this scheduling the wavelengths whose destinations are closer downstream are given higher priorities in order to compensate the effect of the positional priority. Compared to the LQF scheduler, the DPQ scheduler can be more easily implemented because the DPQ uses only destination information of wavelengths in scheduling, which does not change once a network topology is fixed, while the LQF resorts to VOQ length that is continuously changing at each scheduling instant.

D. Shortest Packet First (SPF) Scheduling

While the fairness guarantee is the number one priority in designing the LQF and the DPQ scheduling algorithms, in the SPF scheduling we are trying to maximize bandwidth efficiency by giving priority to wavelengths that have shorter frames in the VOQs. The rationale behind the SPF algorithm is that by sending shorter frames first, it would be possible to reduce the chance of being interrupted by incoming frames.

IV. PERFORMANCE EVALUATION OF SCHEDULING ALGORITHMS

A. Simulation Model and Operational Assumptions

We have developed simulation models for the performance evaluation of the scheduling algorithms based on *Objective Modular Network Testbed in C++* (OMNeT++) [4]. The OMNeT++ is a discrete-event-driven simulator based on C++ and supports models of hierarchically nested modules with multiple links between them, which is an essential feature for the simulation of WDM systems.

The simulation model is for a multiple-access ring network with HORNET architecture, consisting of 10 access nodes with 10 wavelengths on a 100 km ring, where each node on the ring receives frames through a fixed wavelength, but can send frames any wavelengths available through a tunable laser. IP packets are generated with packet size distribution matching that of a measurement trace from one of MCI’s backbone OC-3 links [5] and uniform destination distribution. Although the packet generator can generate packets based on either Poisson process or *Interrupted Poisson Process* (IPP), we report only the

results based on Poisson process due to space-limit in this paper.

The MAC parameters used are summarized in Table I.¹ Note that the optical buffer size of 13 octets corresponds to the minimum required for the transmission of interrupted frame, while 78, 590, and 1538 octets are the buffer sizes for successful transmission of frames with size up to 66, 578, and 1526 octets, respectively, in the worst case that an incoming frame is detected just after the beginning of frame transmission. These are the frame sizes for the popular IP packet sizes: 40, 552, and 1500 octets.

TABLE I
UNSLOTTED CSMA/CA WITH BACKOFF MAC PARAMETERS.

Parameter	Value
Line Speed	10 Gbps
Overhead	26 octets
Guard Band	12 octets (=9.6 ns)
Jamming Signal	1 octet
Optical Buffer Size	13, 78, 590, 1538 octets
VOQ Size	1e5 octets

The following performance measures are used: (1) Throughput per node, (2) fairness index [6], and (3) average end-to-end packet delay. Throughput per node is defined as total number of bits delivered during the simulation divided by the product of simulation time and the number of nodes. The fairness index is used to better quantify the fairness of each scheduling algorithm and based on the throughput of all the connections on the network.

B. Simulation Results

We show simulation results of the scheduling algorithms for optical buffer sizes of 13, 78, 590, and 1538 octets in Figs. 2, 3, 4, and 5, respectively.

The maximum achievable throughput is less than the link capacity because the presence of incomplete frames on the ring constitutes wasted bandwidth as described in [1]. As shown in the figures, the difference in maximum achievable throughput per node among the scheduling algorithms is not significant, less than 1 Gbps. However, the fairness index and the end-to-end packet delay show the clear difference among the scheduling algorithms except for the results shown in Fig. 5 where all the algorithms show similar performances. In general the LQF shows the best results with a right balance between throughput and fairness, but it comes at the expense of relatively higher packet delay, for which the DPQ is the best.

From the results, we also identify that the optical buffer size greatly affects the performance of scheduling algorithms, especially at a higher traffic region with larger than 4 Gbps/node of arrival rate, and that the effect of the optical buffer size is larger for non-random schedulers (DPQ, LQF, SPF) than random scheduler (RS). Of the

¹We adopt parameters from 10 Gigabit Ethernet for frame format (overhead) and interframe gap time (guard band) due to its similarity to the unslotted CSMA/CA with backoff MAC protocol.

buffer sizes considered, 590 octets shows the best performance because compared to the smaller buffer sizes, it increases the chance of finishing remaining frame transmission when an incoming frame is detected. With even larger buffer size (*i.e.*, 1538 octets), however, performance begins to decrease because the wasted bandwidth by large gaps in the optical buffer compensates for the aforementioned effect. Also, as traffic increases, since it's extremely hard for wavelengths with lower positional priority to get selected by schedulers, performance difference among the scheduling algorithms becomes negligible. Note that, however, these results strongly depend on packet size distribution and the operational assumption we take for handling optical buffer status. For example, if we keep track of all incoming frames in the optical buffer and use openings between them for frame transmission, the performance would improve as the buffer size increases. But this highly increases the implementation complexity, which eventually eliminates the benefits of the unslotted CSMA/CA with backoff MAC protocol.

V. SUMMARY

In this paper we have described four scheduling algorithms designed for the unslotted CSMA/CA with backoff MAC protocol and presented the results of the performance evaluation through extensive network-level simulations. From the simulation results, we have verified that in general the LQF shows the best performance in terms of throughput and fairness under uniform traffic condition, while for packet delay, the DPQ shows the best results. We have also identified that the optical buffer size greatly affects the performance of the scheduling algorithms, which depends on packet size distribution and the operational assumptions on the optical buffer handling.

ACKNOWLEDGMENTS

The authors are greatly indebted to Ian M. White, Kapil Shrikhande, and other members of HORNET team at OCRL, Stanford, for their valuable discussions and suggestions for this work.

REFERENCES

- [1] K. Shrikhande, I. M. White, D. Wonglumsom, S. M. Gemelos, M. S. Rogge, Y. Fukashiro, M. Avenarius, and L. G. Kazovsky, "HORNET: A packet-over-WDM multiple access metropolitan area ring network," *IEEE J. Select. Areas Commun.*, vol. 18, no. 10, pp. 2004–2016, Oct. 2000.
- [2] K. Shrikhande, A. Srivatsa, I. M. White, M. S. Rogge, D. Wonglumsom, S. M. Gemelos, and L. G. Kazovsky, "CSMA/CA MAC protocols for IP-HORNET: An IP over WDM metropolitan area ring network," in *Proceedings of GLOBE-COM'00*, Nov. 2000, vol. 2, pp. 1303–1307.
- [3] E. Wong, S. K. Marks, M. A. Summerfield, and R. D. T. Lauder, "Baseband optical carrier-sense multiple access – Demonstration and sensitivity measurements," in *OFC 2001 Technical Digest Series*, Anaheim, CA, Mar. 2001, WU2-1.
- [4] András Varga, *OMNeT++: Discrete event simulation system*, Technical University of Budapest, Mar. 2001, Version 2.1.
- [5] "WAN packet size distribution," <http://www.nlanr.net/NA/Learn/packetsizes.html>.
- [6] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," Tech. Rep. DEC-TR-301, Digital Equipment Corporation, Sept. 1984.

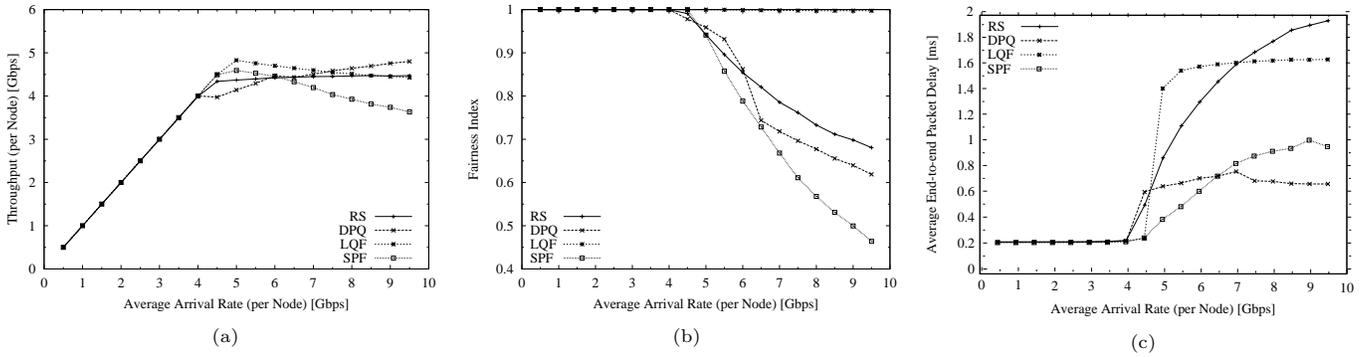


Fig. 2. Performance of designed scheduling algorithms for optical buffer size of 13 octets: (a) Throughput per node, (b) fairness index, and (c) packet end-to-end delay.

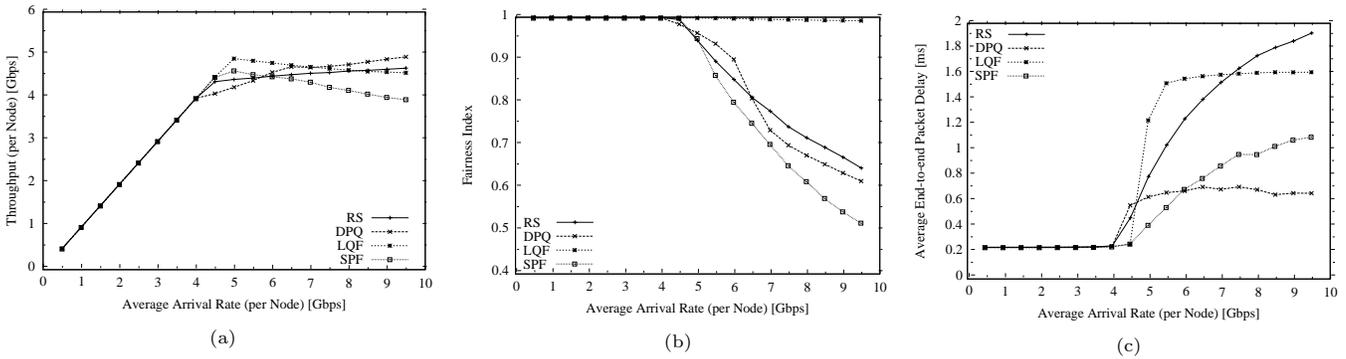


Fig. 3. Performance of designed scheduling algorithms for optical buffer size of 78 octets: (a) Throughput per node, (b) fairness index, and (c) packet end-to-end delay.

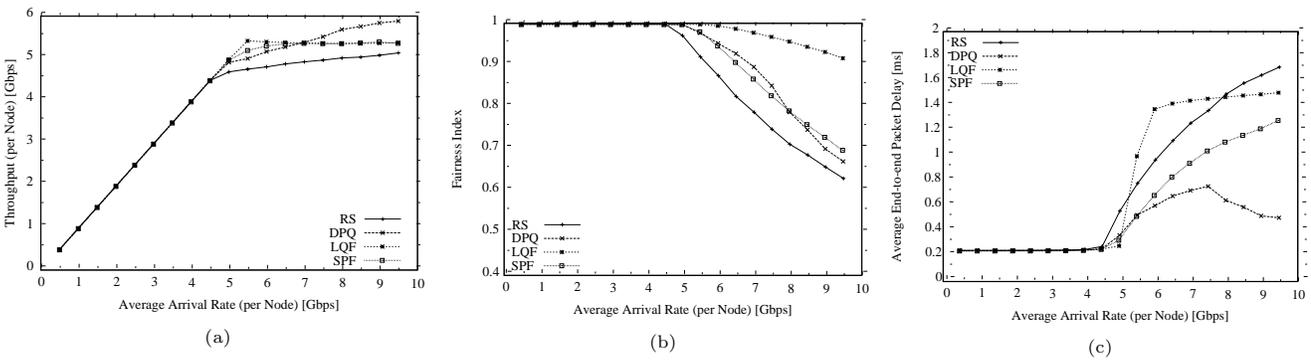


Fig. 4. Performance of designed scheduling algorithms for optical buffer size of 590 octets: (a) Throughput per node, (b) fairness index, and (c) packet end-to-end delay.

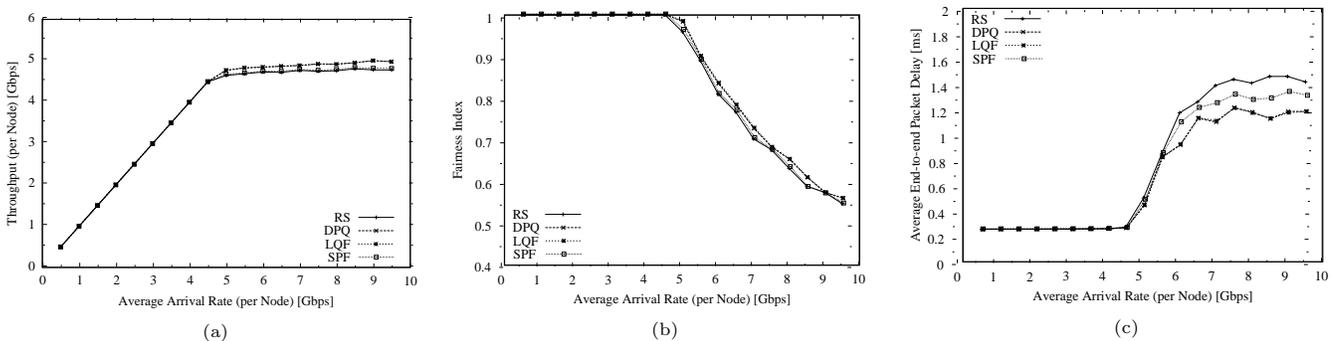


Fig. 5. Performance of designed scheduling algorithms for optical buffer size of 1538 octets: (a) Throughput per node, (b) fairness index, and (c) packet end-to-end delay.

Unslotted Optical CSMA/CA MAC Protocol with Fairness Control in Metro WDM Ring Networks

Kyeong Soo Kim*, Hiroshi Okagawa, Kapil Shrikhande, and Leonid G. Kazovsky

Optical Communications Research Laboratory, Stanford University

Stanford, CA 94305-9515, USA

{kks,okagawa,kapils,kazovsky}@stanford.edu

Abstract—Optical Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is a Media Access Control (MAC) protocol proposed for future metro Wavelength Division Multiplexing (WDM) ring networks with a fixed receiver and a tunable transmitter at access nodes [1], [2]. In this paper, we focus on the unslotted version of the optical CSMA/CA MAC which is a fully-distributed and asynchronous protocol. We present the results of design and performance evaluation of fairness control schemes based on *Longest Queue First* (LQF) scheduling and two random routing algorithms – *Full Random Routing* (FRR) and *Partial Random Routing* (PRR). Through extensive network-level simulation of a WDM ring network with 10 nodes and 10 wavelengths on a 100 km ring at 10 Gbps line rate, we demonstrate a combination of the LQF scheduling and the PRR with a retransmission counter provides good fairness (fairness index [3] of 0.9995) with high bandwidth efficiency and small delay spread, under highly unbalanced traffic conditions.

I. INTRODUCTION

Optical Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) has been proposed as a Media Access Control (MAC) protocol for HORNET (*Hybrid Optoelectronic Ring NETWORK* [1], [2]), a promising packet over Wavelength Division Multiplexing (WDM) Metropolitan Area Network (MAN) architecture, where each node is equipped with a fixed receiver and a tunable laser. Among its many variations, the unslotted version has two unique benefits as an optical MAC protocol [4]: Firstly, it is a fully-distributed, asynchronous protocol not based on a centralized controller or a separate control wavelength to synchronize the operations of nodes on the ring. This is an advantage in implementation compared to the slotted optical MAC protocols, most of which maintain synchronous slot boundaries over many wavelengths through dispersion compensation. Secondly, it can naturally support variable length IP packets without segmentation and reassembly function if desired. These features make the unslotted optical CSMA/CA an attractive MAC protocol for future optical MANs and Local Area Networks (LANs).

Because of unidirectional transmission of signal on the optical ring and collision avoidance action of the MAC protocol, incoming frames from upstream nodes take priority over outgoing frames at a node. Hence, there arises the so-called *positional*

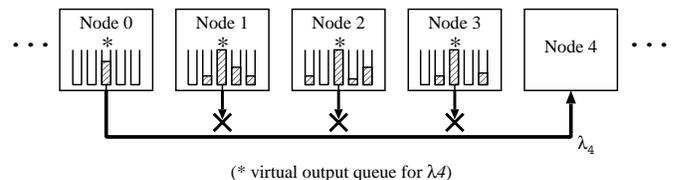


Fig. 1. An example network scenario showing severe unfairness due to positional priority and unbalanced traffic.

priority problem where for a given destination and the corresponding wavelength, access nodes farther from the destination node have higher priorities over those closer to the destination node [5]. Therefore guaranteeing fairness among different traffic streams at different nodes is critical for both the unslotted and slotted optical MAC protocols.

There have been proposed several slotted optical MAC protocols to address this fairness issue in WDM ring or dual bus networks [6], [7], [8], [9], where a dedicated control channel or separate control messages in the same data channels are used to exchange control information among access nodes. Unslotted optical MAC protocols, however, have been getting less focus in the literature in spite of the aforementioned benefits because of the complexity in their analyses by either simulations or mathematical techniques, and the seemingly lower bandwidth efficiency.

Recently we studied scheduling algorithms for unslotted optical CSMA/CA MAC protocol and demonstrated they can effectively guarantee fairness under uniform traffic conditions through network-level simulations [4], [10]. Scheduling alone, however, cannot guarantee fairness under highly unbalanced traffic conditions. For instance, as illustrated in Fig. 1, a single stream from node 0 to node 4 blocks traffic from all other nodes upstream to the same destination. Because there is only one traffic stream at node 0 in this configuration, any scheduling algorithm cannot but select the channel λ_4 all the time.

In this paper we propose and present the performance of fairness control schemes based on *Longest Queue First* (LQF) scheduling and random routing algorithms – *Full Random Routing* (FRR) and *Partial Random Routing* (PRR) – for unslotted optical CSMA/CA MAC protocol that can provide fairness among streams even under highly unbalanced traffic conditions as well as balanced traffic conditions. Unlike the existing fairness control schemes in the slotted optical MAC protocols, the proposed schemes do not need any dedicated control channels

This work was sponsored by the Stanford Networking Research Center (SNRC, <http://snrc.stanford.edu>).

*K. S. Kim is with the Advanced System Technology, STMicroelectronics.

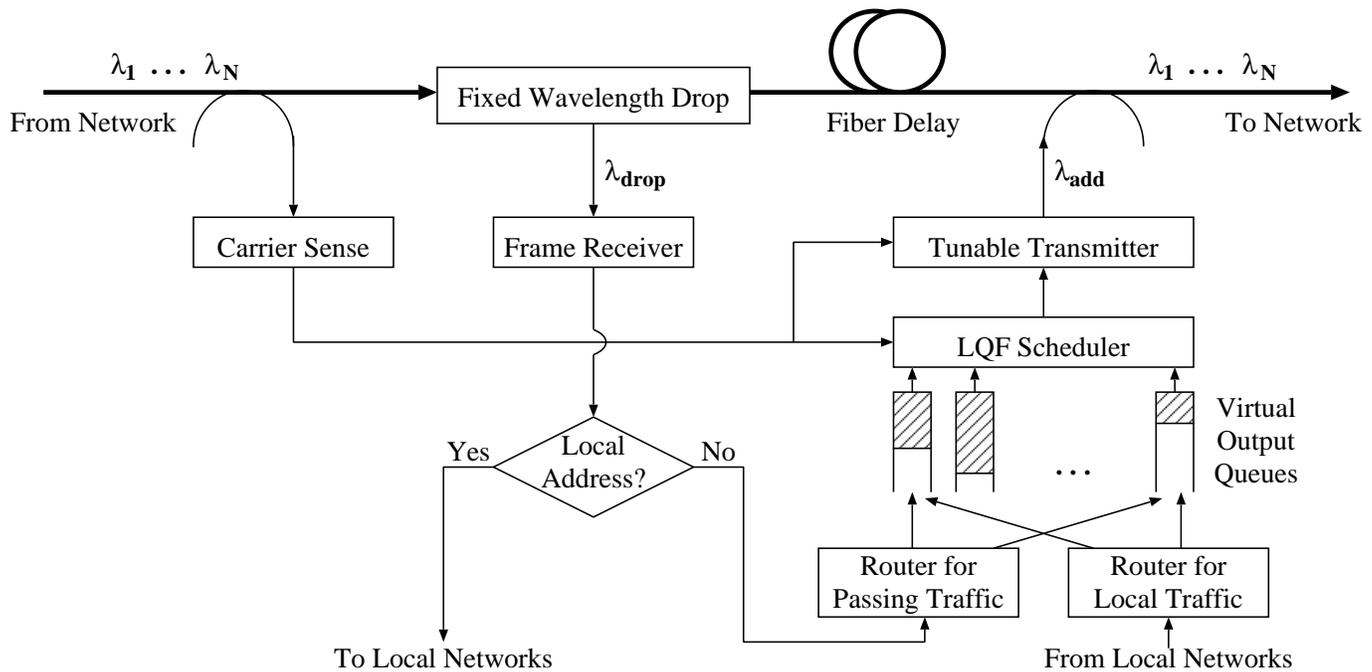


Fig. 2. Access node for unslotted optical CSMA/CA MAC with fairness control.

or messages.

The rest of the paper is organized as follows: In Section II we describe the proposed fairness control schemes with a possible implementation of access node structure. In Section III we present initial simulation results for the proposed fairness control schemes. In Section IV, based on the initial simulation results, we discuss the enhancement of the fairness control schemes with a retransmission counter, and show performance improvements of the enhanced scheme against the original one through simulations in Section V. Section VI summarizes our work and discusses future work.

II. UNSLOTTED OPTICAL CSMA/CA MAC WITH PROPOSED FAIRNESS CONTROL

Fig. 2 shows a block diagram of an access node for the unslotted optical CSMA/CA MAC protocol with the proposed fairness control scheme based on the LQF scheduling and random routing.

While the frame receiver receives the frames on a fixed wavelength, the carrier sense *listens* to all wavelength by monitoring either sub-carriers [1] or baseband optical signals [11], depending on the implementation. When there are frames ready for transmission in *Virtual Output Queues* (VOQs) and channels are available, the LQF scheduler chooses a channel for frame transmission based on channel availability and VOQ lengths.

The LQF scheduler has been chosen because it shows the best performance in terms of throughput and fairness guarantee under balanced traffic condition with the minimum optical buffer size of 13 octets [4], [10]. The LQF scheduler selects a channel with the longest VOQ to counteract the effect of positional priority because VOQs with lower positional priorities are likely to be longer than those with higher positional priorities.

After waiting for a guard band time, if the scheduled channel is still available, the access node starts transmitting the frame. However, since the access node cannot know if the opening on the channel is long enough to accommodate the entire frame, it continues to monitor the channel. For this purpose a small ‘fixed’ optical delay line (*i.e.*, optical buffer) is placed between the carrier sense and the tunable transmitter. If the carrier sense detects a frame arriving on the same wavelength and the optical buffer size is not big enough for successful transmission of the remaining frame with a guard band, it immediately interrupts the frame transmission and sends a jamming signal. Otherwise, it can transmit the entire frame without interruption.

Note that the optical buffer size should be at least large enough to transmit the jamming signal and the guard band before the incoming frame. The jamming signal (like in the Ethernet) could be a unique bit pattern, either at baseband or on sub-carrier. The frame receiver at downstream access node recognizes the incomplete frame by the presence of the jamming signal and pulls it off the ring. The access node can reschedule the transmission of the frame for a later time.

To provide fairness even under unbalanced traffic conditions like the one shown in Fig. 1, we use random routing schemes. We propose two random routing schemes, FRR and PRR, for this purpose. In the FRR, frames from local networks (local frames) and from other nodes (multihopping frames) are randomly routed over VOQs, while in the PRR, only local frames are randomly routed but multihopping frames are correctly routed based on their destination addresses. Then the scheduler schedules transmission of frames in VOQs based on its scheduling algorithm as usual.

In random routing schemes, some frames are directly delivered to their destinations, but others through several intermediate nodes until finally reaching their destinations. By this

random nature in distribution of traffic over channels, there can be some alleviation in channel overloading. Therefore we can avoid starvation of nodes closer to the destination, which leads to better fairness among traffic streams under highly unbalanced traffic conditions.

III. SIMULATION RESULTS I – FRR AND PRR

We have developed a simulation model for the performance evaluation of the proposed fairness control schemes based on *Objective Modular Network Testbed in C++* (OMNeT++) [12]. The OMNeT++ is a discrete-event-driven simulator based on C++ and supports models of hierarchically nested modules with multiple links between them, which is an essential feature for the simulation of WDM systems.

The simulation model is for a WDM ring network with HORNET architecture, consisting of 10 access nodes and 10 wavelengths on a 100 km ring network at 10 Gbps line rate, where each node on the ring receives frames through a fixed wavelength and send frames any wavelengths available through a tunable laser. IP packets are generated according to Poisson process with the packet size distribution matching that of a measurement trace from one of MCI’s backbone OC-3 links [13].

In the simulation IP packets are encapsulated in Ethernet frames before being transmitted over the fiber. Since we set the line rate to 10 Gbps for our simulation, we adopt frame format from 10 Gigabit Ethernet specifications and assume a frame overhead of 26 octets. We also assume that a guard band is 12 octets (=9.6 ns), a jamming signal 1 octet, the optical buffer size 13 octets, which is the minimum required for the transmission of interrupted frame, and the VOQ size 10^5 octets.

For traffic condition, we consider a scenario where nodes 0 to 8 communicate only with a hot-spot node 9 at rates of 1.2 Gbps bidirectionally, which overloads the channel to node 9. Note that there is only one outgoing stream at nodes 0 to 8, while at node 9, there are streams to all other nodes.

Fig. 3 shows throughputs of both upstream and downstream connections. Note that we also include the performance of the non-random routing scheme (LQF scheduling alone), which we call *fixed routing*, for the purpose of comparison. It is clear that fixed routing suffers from unfairness in upstream direction, due to which nodes closer to the destination (in this case, nodes 5 to 8) actually starve. In downstream direction, however, the fixed routing can provide good throughput and fairness because the traffic condition at node 9 is balanced and there is no contention over channels.

On the other hand, the random routing schemes can provide better fairness preventing starvation of nodes closer to the destination. The FRR shows pretty good fairness in both directions, but there are significant penalties in throughput, which results from the increasing contention in the network due to significant amount of multihopping traffic. In the case of PRR, which limits the maximum number of hops to 2 and thereby reduces the amount of multihopping traffic, we can see significant improvements in throughput over FRR but at the expense of fairness.

Fig. 4 shows end-to-end packet delay distributions of sampled upstream connections (from nodes 0, 4, and 8 to node 9, respectively). Because the FRR doesn’t limit the number of hops packets can take, the delay distribution is more widely spread

compared to the PRR. We can expect the difference in delay distributions to be bigger when we increase the total number of nodes in the network because the average number of hops for the FRR is the total number of nodes minus 1, while the maximum number of hops is limited to 2 in the case of PRR. Note that with the random routing schemes frames can be delivered out of order. So we need a resequencing buffer at the link/MAC layer. Due to its smaller delay spread, the PRR requires smaller resequencing buffer compared to the FRR.

IV. ENHANCEMENT OF PRR SCHEME WITH RETRANSMISSION COUNTER

Although PRR as described in Section III has better throughput and packet delay distribution than FRR, still there is a room for improvement in fairness guarantees.

When a node fails to transmit a frame on a channel due to incoming frames from upstream nodes, it keeps the frame in the VOQ and tries to retransmit it later when the channel is available. Careful examination of the simulation results in Section III, however, shows that when the channel is overloaded, it has little chance to transmit the holding frame therefore blocking transmission of all other frames in the VOQ, which eventually causes packet losses due to buffer overflow. This problem is severe especially when the positional priority of the node is very low or the frame in transmission process is very long.

To solve this long transmission blocking problem, we introduce a *Retransmission Counter* (RC) that limits the maximum number of retransmissions. The transmission procedure using the RC is as follows: The RC increases whenever frame transmission fails due to incoming frames. If the value of the RC reaches a certain limit, the transmitter discards the frame, schedules another one from the VOQs, and tries to transmit it as usual.

If TCP flow control is used for a connection in the upper layer, packets lost with this scheme will get retransmitted. Note that, in such a case, the PRR provides alternative paths (channels) to retransmitted packets, which would increase the chance of successful transmissions.

We call this new scheme *PRR with RC*.

V. SIMULATION RESULTS II – PRR WITH RETRANSMISSION COUNTER

Simulation settings are the same as in Section III. We set the maximum limit of the RC to 10.¹

Fig. 5 shows the throughput of both upstream and downstream connections for the original PRR and PRR with RC. Our results verify that PRR with RC greatly improves the fairness for both upstream and downstream connections. The fairness index [3] is 0.9995 for both upstream and downstream in the case of the PRR with RC, while in the original PRR, they were 0.9150 and 0.9992 for upstream and downstream, respectively. Additionally, the bandwidth efficiency has been improved with the introduction of the RC: The total throughput of the whole

¹We have verified that the maximum limit of the RC of 10 is optimal given the simulation settings through separate analysis which is not reported in this paper due to space limitation.

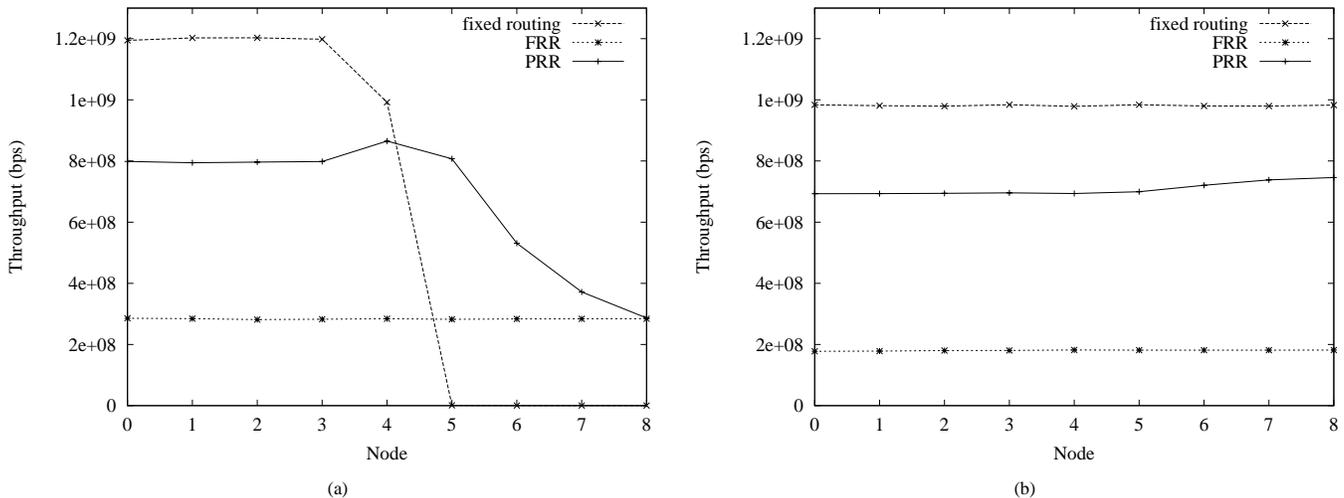


Fig. 3. Throughputs of connections for the proposed routing algorithms: (a) upstream and (b) downstream.

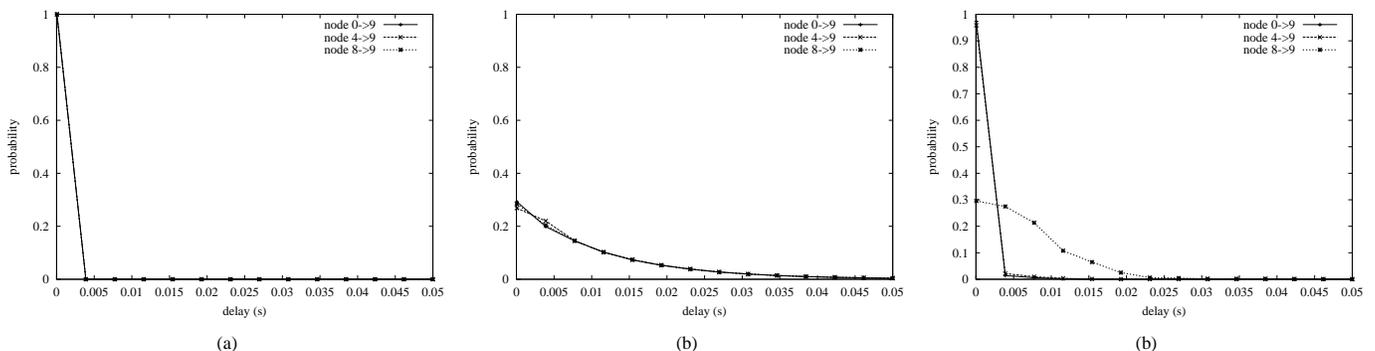


Fig. 4. End-to-end packet delay distributions of sample connections for the proposed routing algorithms: (a) fixed routing, (b) FRR, and (c) PRR.

network has increased from 6.048 Gbps to 6.111 Gbps in upstream and from 6.372 Gbps to 6.975 Gbps in downstream, respectively.

Fig. 6 shows end-to-end packet delay distributions of sampled upstream connections. It is evident that with the RC, there is virtually no difference among delay distributions for different streams and the result looks like that of the fixed routing. Therefore we infer that the major cause of the delay spread for the stream from node 8 to node 9 in the original PRR (shown in Fig. 4) is the blocking problem discussed in Section IV.

From the results, we have verified that the introduction of the RC to the original PRR scheme greatly improves the performance of the unslotted optical CSMA/CA MAC protocol in all the measures we considered – throughput, fairness, and end-to-end packet delay.

VI. SUMMARY AND FUTURE WORK

In this paper we have proposed fairness control schemes based on the LQF scheduling and two random routing algorithms – the FRR and the PRR – for the unslotted optical CSMA/CA MAC protocol. Initial simulation results show the PRR, compared to the FRR, provides better throughput and delay performance, but at the expense of fairness. To enhance the fairness performance of the original PRR, we have introduced the RC to solve the problem of long transmission blocking by

limiting the maximum number of retransmissions. Through simulations we have verified that the introduction of RC greatly improves the performance of the original PRR scheme in all the measures considered – throughput, fairness, and end-to-end packet delay. Considering that the PRR with RC does not use any reservation mechanism with separate control channels or messages, it is encouraging that the proposed scheme can guarantee good fairness, with fairness index close to 1, even under highly unbalanced traffic conditions.

The actual end-to-end performance of the optical unslotted CSMA/CA MAC protocol with fairness control can be estimated only with realistic network environment with upper layers including TCP/IP protocols. We are currently implementing new simulation models with full TCP/IP protocol stack, which will enable us to better understand the actual behavior of the MAC protocol and its interaction with TCP flow control. Also we are working on the *Adaptive Random Routing* scheme taking advantage of both high transmission efficiency of the fixed routing under balanced traffic conditions and good fairness of the PRR under unbalanced traffic conditions.

REFERENCES

- [1] K. Shrikhande, I. M. White, D. Wonglumsom, S. M. Gemelos, M. S. Rogge, Y. Fukashiro, M. Avenarius, and L. G. Kazovsky, "HORNET: A packet-over-WDM multiple access metropolitan area ring network," *IEEE J. Select. Areas Commun.*, vol. 18, no. 10, pp. 2004–2016, Oct. 2000.

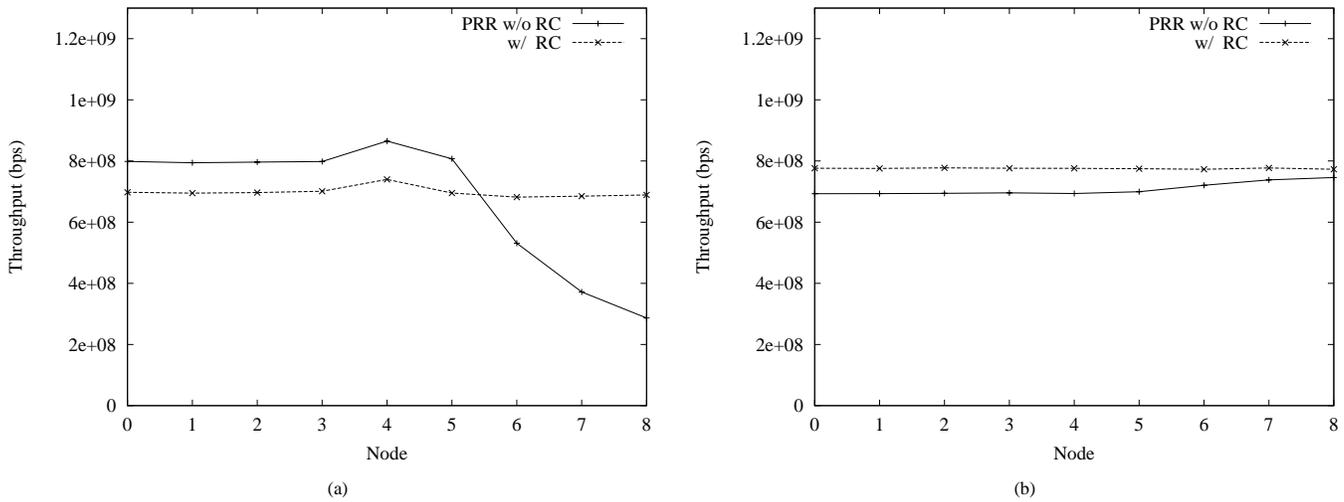


Fig. 5. Throughputs of connections for the PRR and the PRR with RC: (a) upstream and (b) downstream.

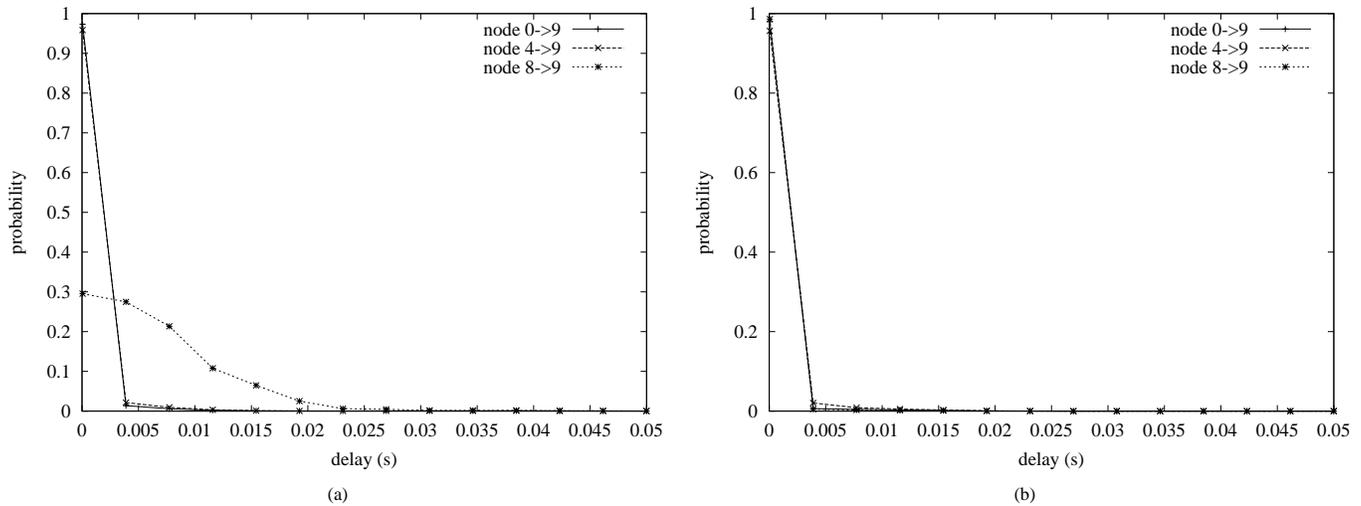


Fig. 6. End-to-end packet delay distributions of sample connections for (a) PRR and (b) PRR with RC.

- [2] K. Shrikhande, A. Srivatsa, I. M. White, M. S. Rogge, D. Wonglumsom, S. M. Gemelos, and L. G. Kazovsky, "CSMA/CA MAC protocols for IP-HORNET: An IP over WDM metropolitan area ring network," in *Proceedings of GLOBECOM'00*, Nov. 2000, vol. 2, pp. 1303–1307.
- [3] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," Tech. Rep. DEC-TR-301, Digital Equipment Corporation, Sept. 1984.
- [4] K. S. Kim and L. G. Kazovsky, "Design and performance evaluation of scheduling algorithms for unslotted CSMA/CA with backoff MAC protocol in multiple-access WDM ring networks," in *Proceedings of JCIS 2002*, Mar. 2002, pp. 1303–1306.
- [5] I. M. White, D. Wonglumsom, K. Shrikhande, S. M. Gemelos, M. S. Rogge, and L. G. Kazovsky, "The architecture of HORNET: A packet-over-WDM multiple-access optical metropolitan area ring network," *Computer Networks*, vol. 32, no. 5, pp. 587–598, May 2000.
- [6] K. W. Cheung and V. W. Mak, "EQEB – A multichannel extension of the DQDB protocol with tunable channel access," in *Proceedings of GLOBECOM'92*, Dec. 1992, vol. 3, pp. 1610–1617.
- [7] J. C. Lu and L. Kleinrock, "A WDMA protocol for multichannel DQDB networks," in *Proceedings of GLOBECOM'93*, Nov. 1993, vol. 1, pp. 149–153.
- [8] M. A. Marsan, A. Bianco, E. Leonardi, M. Meo, and F. Neri, "MAC protocols and fairness control in WDM multirings with tunable transmitters and fixed receivers," *J. Lightwave Technol.*, vol. 14, no. 6, pp. 1230–1244, June 1996.
- [9] M. A. Marsan, A. Bianco, E. Leonardi, A. Morabito, and F. Neri, "All-optical WDM multi-rings with differentiated QoS," *IEEE Communications Magazine*, vol. 37, no. 2, pp. 58–66, Feb. 1999.
- [10] K. S. Kim and L. G. Kazovsky, "Design and performance evaluation of scheduling algorithms for unslotted CSMA/CA with backoff MAC protocol in multiple-access WDM ring networks," To appear in *Information Sciences*, 2002.
- [11] E. Wong, S. K. Marks, M. A. Summerfield, and R. D. T. Lauder, "Base-band optical carrier-sense multiple access – Demonstration and sensitivity measurements," in *OFC 2001 Technical Digest Series*, Anaheim, CA, Mar. 2001, WU2.
- [12] András Varga, *OMNeT++: Discrete event simulation system*, Technical University of Budapest, Mar. 2001, Version 2.1.
- [13] "WAN packet size distribution," <http://www.nlanr.net/NA/Learn/packetsizes.html>.

Design of a control-channel-based MAC protocol for HORNET

I. M. White, M. S. Rogge, K. Shrikhande, and L. G. Kazovsky

*Stanford University Optical Communications Research Laboratory
350 Serra Mall MC9515, Stanford, CA 94305
ianwhite@stanfordalumni.org*

The *HORNET* architecture is a packet-over-WDM ring network that utilizes fast-tunable packet transmitters and wavelength routing to enable it to scale cost-effectively to ultra-high capacities. In this work, we present the design of a novel control-channel-based media access control protocol, which is optimized for variable-sized IP packets and provides fairness control. The design of the control channel, including the frame structure and a frame synchronization protocol, are described in detail. © 2002 Optical Society of America

OCIS codes: 060.2330, 060.4250.

1. Introduction

Internet traffic on next-generation metropolitan networks will have three very important characteristics. It will be dominated by bursty, packet-based data traffic, it will fluctuate heavily at random, and it will increase in the coming years up to, and eventually beyond 1 Tb/s. The architecture for next-generation metro networks must be designed in consideration of these characteristics. We have proposed a new architecture named *HORNET* (Hybrid Opto-electronic Ring Network), which satisfies all of the requirements of next-generation metro networks.^{1,2} In this work, the design of the media access control (MAC) protocol designed for *HORNET* is presented. The protocol is designed to efficiently transport variable-sized IP packets, and it provides fairness control.

Section 2 discusses the architecture of *HORNET*. Section 3 describes the design of the control-channel-based MAC protocol and reports performance results from a computer simulator. Section 4 presents the design of the control channel. Included in Section 4 are a description of the control channel frame and the design and experimental demonstration of a synchronization protocol for the control channel.

2. HORNET Architecture

The *HORNET* architecture cost-effectively scales beyond 1 Tb/s while efficiently transporting bursty, packet-based, randomly fluctuating traffic. The architectural concept is shown in Figure 1. *HORNET* is a bi-directional ring topology designed to leverage the currently deployed fiber infrastructure. Also, the architecture enables *HORNET* to be survivable, as has been experimentally demonstrated.²

As Figure 1 shows, nodes use fast-tunable packet transmitters to insert packets onto the ring. The packets are coupled optically onto the ring using a wideband coupler (a fast-tunable wavelength-selective multiplexer is currently not available). A packet is transmitted on the wavelength that is received by the packet's destination node. A wavelength drop is used to drop one or more assigned wavelengths into each node. Thus, only the packets destined for a particular node are dropped into the node. All of the packets carried by the other wavelengths pass through optically,

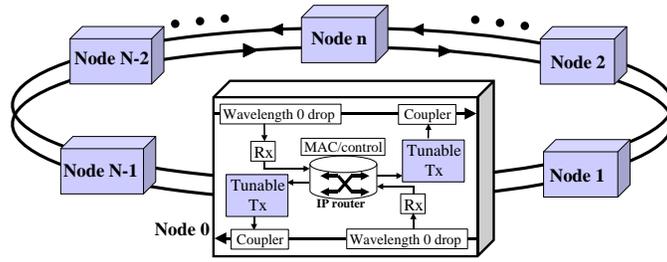


Fig. 1. The *HORNET* architecture is a bi-directional wavelength routing ring network with tunable transmitters in each node.

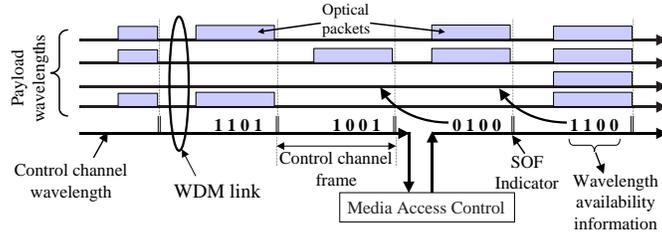


Fig. 2. The control channel conveys the availability of the wavelengths during a framed time period.

such that the node does not receive or process them. Conventional architectures require significantly more equipment in the nodes because the nodes must receive, process, and re-transmit all packets that pass through. In *HORNET*, a node only needs enough equipment to process the packets to and from its local users.

3. *HORNET* Media Access Control Protocol

The primary function of the MAC protocol in *HORNET* is to prevent collisions at the point in the node where the transmitter inserts packets. Since the transmitter can insert a packet on any wavelength, and since most of the wavelengths are passing through the node without being terminated, a transmitter could insert a packet onto a particular wavelength that collides with another packet that is passing through the node on that wavelength. To prevent collisions, the MAC protocol monitors the WDM traffic passing through the node, locates the wavelengths that are available, and informs the transmitter which wavelengths it is allowed to use at a particular moment. As a result, the transmitter will not insert a packet on a wavelength that is currently carrying another packet through the node.

HORNET uses a control channel to convey the *wavelength availability information*. The control channel is carried on its own wavelength in the WDM network. That control wavelength is *dropped* and *added* in every node so that all nodes can process and modify the control channel. It is recommended to use a control channel wavelength that is away from the payload wavelength band (e.g. 1310 nm). This allows the control channel transmitters to be far less expensive than if a wavelength were used within the WDM spectrum of the payload wavelengths.

Figure 2 illustrates the operation of the control channel for the MAC protocol. The control channel is time-slotted into frames, which are each bounded by a *start-of-frame* (SOF) indicator byte. Within each frame is a bit-stream that conveys the wavelength availability information for the time period during the following frame. This allows the node to see one frame into the future.

A node sorts its queued packets into virtually separated queues called virtual output queues (VOQs),³ the classic technique to avoid the head-of-line (HOL) blocking problem.⁴ Each VOQ corresponds to a wavelength in the network. When a node reads the bit stream, it determines the set of VOQs with a packet to transmit that overlaps with the set of available wavelengths. The node then determines which packet in the overlapping set it will transmit during the next frame. If the node decides to send a packet on wavelength w , it modifies bit w in the wavelength availability bit-stream to a '1.' All nodes clear the wavelength availability bit(s) corresponding to the wavelength(s) that they receive.

3.A. MAC Optimization for Variable-Sized Packets

The framed format of the control channel makes the MAC protocol ideal for small, fixed-sized packets. However, Internetworking Protocol (IP) packets are inherently variable in size. An estimated IP packet size distribution that is based on the data measurements reported by the National Laboratory for Applied Network Research (NLANR)⁵ is shown in Figure 3 (a). This is the distribution expected for the *HORNET* network. As the figure shows, packet sizes vary from 40 bytes to 1500 bytes. Such a wide range of packet sizes is not compatible with a framed control channel with inflexible frame sizes. A simple solution exists for this problem that avoids any changes to the MAC protocol. As is done in IP-over-ATM, the variable-sized IP packets can be segmented into small, fixed-sized cells. The size of the segmented cell and the size of the control channel frame can be designed to match each other. However, this design results in an excessive amount of overhead. When a packet or a segment of a packet is transmitted, a header must be applied. Thus, a long packet will have the *HORNET* packet header applied to it a large number of times.

To avoid this excessive overhead, a protocol was designed for the *HORNET* MAC called *segmentation and re-assembly on demand* (SAR-OD). In this protocol, a node must begin to insert a packet in alignment with the beginning of the control frame. If the packet is longer than the control frame duration, the node *continues to transmit* the packet (without segmenting the packet and re-applying the header) until either the packet is complete or until the MAC protocol informs the transmitter that another packet is coming from upstream on the transmission wavelength. When this occurs, the node *ceases the transmission* of its packet at the end of the last available frame (i.e. the one before the frame that is carrying the oncoming packet). At the end of the packet segment, the transmitter applies a byte that indicates that the segment is an incomplete packet. The node is now free to send packets on different wavelengths while it waits for an opportunity to finish the packet it had begun. At the next opportunity, the node begins transmitting the segmented packet beginning with the location in the packet at which it was segmented. When the final segment of a packet is completely transmitted, the node finishes the packet with a byte that indicates that the packet is complete.

The receiver in a *HORNET* node maintains separate virtual queues for each node on the ring. When a packet arrives at the receiver, the receiver reads the packet header to determine the source node and then writes the payload of the arriving packet into the virtual queue corresponding to the source node. If the last byte of the segment indicates that the packet is incomplete, the segment remains in the queue. The next segment arriving at the receiver from the same source node will belong to the same packet, and thus the receiver will store this segment at the queue location immediately following the previously received segment. When the packet is fully received, it is sent to the node's packet switch with the integrity of the IP packet completely preserved.

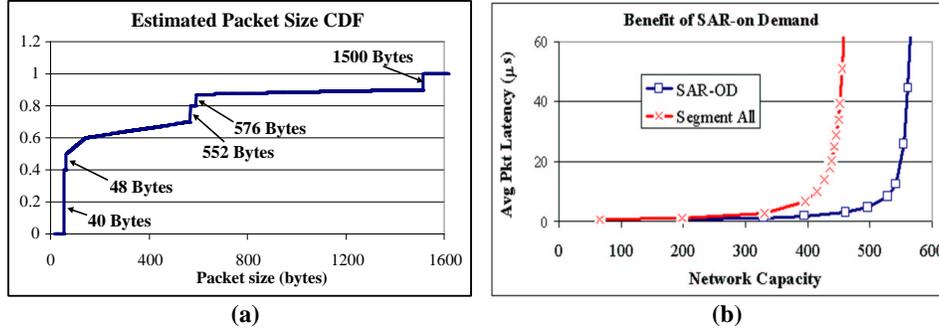


Fig. 3. (a) Estimated CDF of IP packet sizes based on the collection of data for various links as measured by NLANR. (b) Simulated performance advantage of using SAR-OD instead of automatically segmenting all packets into small, fixed-sized cells. The network in the simulation has 33 nodes and 33 wavelengths.

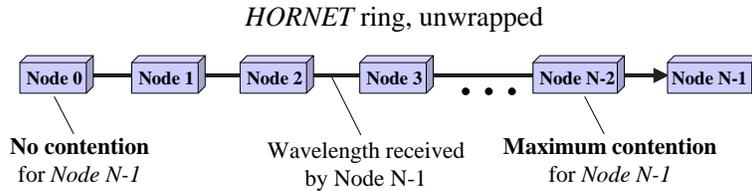


Fig. 4. The *HORNET* ring unwrapped, while focusing on the wavelength received by Node $N-1$.

Custom-designed computer simulations were used to evaluate the performance advantage of SAR-OD.¹ The performance advantage measured by the simulator is shown in Figure 3 (b). The packet size distribution shown in Figure 3 (a) is used in this comparison. The graph shows a performance advantage of approximately 15%, which is expected. The simulator measured an overhead of 10.5% when the network used the SAR-OD protocol. The overhead for a network that segments all packets can easily be calculated to be more than 25% (16 bytes of overhead in every 64-byte slot, plus unused bytes at the end of the packet). As a result, a performance advantage of at least 15% is expected.

3.B. HORNET Fairness Control Protocol: DQBR

Although there are many advantages to using the bi-directional ring architecture for *HORNET*, there is a problem that arises because of it. Multiple-access ring networks are inherently unfair. Consider the wavelength that is received by Node $N-1$, as is done in Figure 4. When Node 0 wants to send packets to Node $N-1$, it is never blocked on that wavelength. When Node 1 wants to send packets to Node $N-1$, it has to contend with (can occasionally be blocked by) the packets transmitted by Node 0. Likewise, Node 2 has to contend with Nodes 0 and 1. This pattern continues around the ring to Node $N-2$, which has to contend with all nodes. Clearly, the network is *biased against* nodes closer to the destination. As a result, the VOQs that are queuing packets for *unfortunate* source-destination pairs will experience lower throughput, resulting in *higher latency*. Clearly, fairness control is necessary for the *HORNET* MAC to avoid this negative result.

The fairness control protocol developed for *HORNET* is a novel protocol designed specifically for incorporation into the MAC protocol. It is called *Distributed Queue*

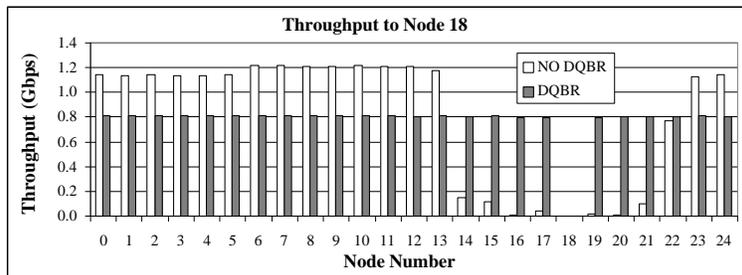


Fig. 5. Throughput for VOQ 18 for the 25 nodes on a *HORNET* network. VOQ 18 corresponds to Wavelength 18, which is received by Node 18.

Bi-directional Ring (DQBR) because the protocol attempts to transform *HORNET*'s bi-directional ring into a distributed first-come-first-serve (FCFS) queue. The protocol is an adaptation of an older protocol called *Distributed Queue Dual Bus* (DQDB),^{6,7} which was created for dual-bus metro networks.

The DQBR fairness control protocol works as follows. In each control channel frame, a bit stream of length W called the *request bit stream* follows the *wavelength availability information*, where W is the number of wavelengths. When a node on the network receives a packet into VOQ w , the node notifies the *upstream* nodes about the packet by setting bit w in the *request bit stream* in the control channel that travels upstream with respect to the direction the packet will travel. For the case of variable-sized packets, the node places the number of requests corresponding to the length of the packet measured in frames.¹

All upstream nodes take note of the requests by incrementing a counter called a *request counter* (RC). Each node maintains an RC corresponding to each wavelength. Thus, if bit w in the *request bit stream* is set, RC_w is incremented. Each time a packet arrives to VOQ w , the node stamps the value in RC_w onto the packet and then clears the RC. The stamp is called a *wait counter* (WC). After the packet reaches the front of the VOQ, if the WC equals n it must allow n frame availabilities to pass by for downstream packets that were generated earlier. When an availability passes by the node on wavelength w , the WC for the packet in the front of VOQ w is decremented (if the WC equals zero, then RC_w is decremented). Not until the WC equals zero can the packet be transmitted. The counting system ensures that the packets are transmitted in the order that they arrived to the network.

The computer simulator was used to test the performance of the DQBR fairness control scheme. Figure 5 shows the throughput for nodes sending packets to Node 18 on a 25-Node *HORNET* network. The traffic generated by the network for Node 18 is 1.5 times the capacity that the wavelength can carry (it is oversubscribed). With DQBR, the *throughput is equal for all nodes*, whereas *without* DQBR, the nodes close to Node 18 have a very difficult time sending packets to Node 18.

4. *HORNET* Control Channel Design

The control channel design significantly impacts both the performance and the cost of the network. This section discusses two important aspects of the control channel design. First, the structure and the optimal length of the control channel frame are described. Then, the synchronization of the control channel with the packets on the payload wavelengths is discussed. Included in this discussion is an experimental demonstration of a frame synchronization protocol developed for *HORNET*.

1 Byte	4 Bytes	W/8 Bytes	W/8 Bytes	Idle	SOF Indicator
SOF Indicator	Control message	Wavelength availability information	DQBR requests		SOF Indicator

Fig. 6. Information contained within each control channel frame.

4 Bytes	4 Bytes	2 Bytes	1 Byte	4 Bytes	Variable length	1Byte
Guard time	Synchronization sequence	Address information	Control information	CRC	TCP header + IP payload data	Trailer

Fig. 7. Contents of a *HORNET* packet.

4.A. Control Channel Frame Length

As was described in Section 3.A, the MAC protocol requires all packets to be inserted to coincide with the beginning of the control channel frame. If a packet that is being transmitted on a particular wavelength is completed somewhere in the middle of the control frame, then the rest of the control channel frame duration on that wavelength must go unused. The unused time period on the wavelength is considered overhead and detracts from the performance of the network. The minimization of this overhead occurs at the optimal match between control channel frame length and the distribution of IP packet sizes. Figure 6 illustrates the components of the control channel frame. Note that for a network with 128 wavelengths, the minimum frame size is 37 bytes.

The *HORNET* transmitter adds a header onto the front of the packet and a trailer to the rear of the packet. The packet size distribution used to determine the optimal control channel frame length must include the payload data, the TCP/IP header, and the *HORNET* header and trailer. The trailer indicates whether the packet is complete or segmented, as discussed in Section 3.A. The header has several purposes. It includes guard time for transmitter tuning, a sequence for bit-synchronization, source and destination address information, control information, and a cyclic redundancy check (CRC) for determining the integrity of the packet. It is anticipated that a futuristic commercial implementation of *HORNET* would use a header of 16 bytes. The structure of a *HORNET* packet is illustrated in Figure 7. The values for *guard time* and *synchronization sequence* are heavily dependent on the progression of the transmitter and receiver technology.

The expected overhead for varying frame sizes is shown in Figure 8 (a). Smaller frame sizes result in less overhead because of the significant amount of small packets (see Figure 3 (a)) and because of the fact that when a packet finishes before the end of a frame, the remainder of the frame duration is *overhead*. The calculation uses the packet size distribution of Figure 3 (a), and does not consider the overhead due to packet segmentation. For the calculation, the *HORNET* header/trailer is 16 bytes. The overhead is defined as the percentage of the transmission that does not contain payload, where the payload in this analysis includes the TCP/IP header and the data within the packet. The overhead bytes include the *HORNET* header and any *unused* bytes after the packet (before the next control frame).

The simulator can be used to verify the optimal control channel frame size. Figure 8 compares the performance of a 17-node *HORNET* network with control channel frame sizes of 40, 56, 64, and 200 bytes while using the variable-sized packet distribution shown in Figure 3 (a). As Figure 8 shows, with a large control channel frame size (e.g. 200 bytes), performance is seriously degraded because of the amount of overhead incurred when transmitting short packets, which happen to dominate the packet size distribution. Performance is relatively similar for the three short

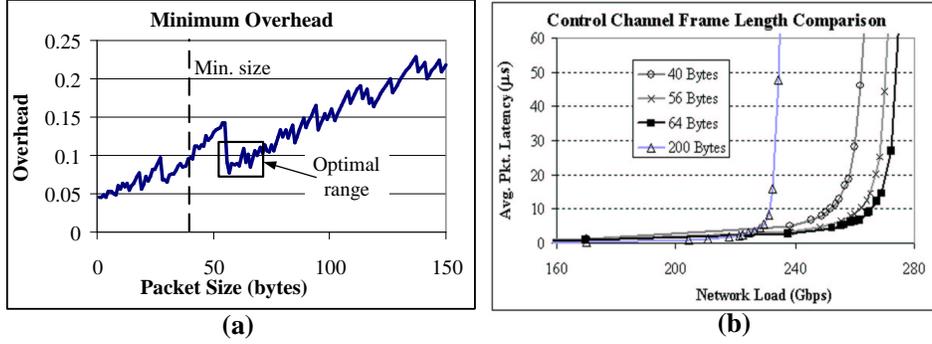


Fig. 8. (a) Expected overhead for *HORNET* with a packet size CDF shown in Figure 3. (b) Simulated performance with varying control frame sizes.

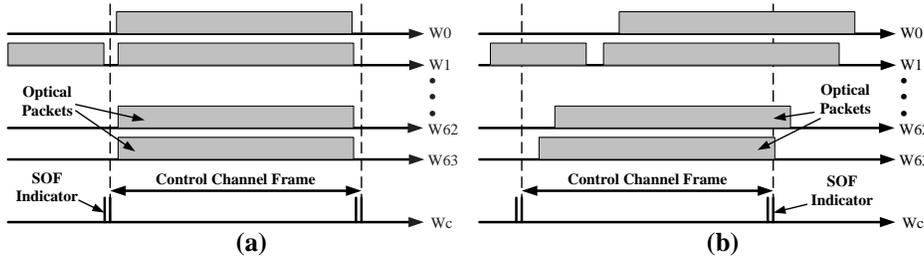


Fig. 9. Optical packets in a WDM system (a) *before* propagating through SMF, and (b) *after* propagating through SMF. W_c = control wavelength.

control channel frame sizes, but 64 bytes has the best performance.

4.B. Dispersion Management for Control Frame Alignment

Frame misalignment occurs in *HORNET* because the group velocity dispersion (GVD) of standard single mode fiber (SMF) causes optical signals on different wavelengths to travel at different speeds. The misalignment is illustrated in Figure 9. If a packet becomes misaligned with the SOF indicators, then another packet may collide with it when inserted into the ring. It is necessary to insert dispersion compensating fiber (DCF) throughout the network to reverse the effect of the GVD of SMF. Optimized lengths of DCF can be concatenated with the transmission fiber at each node. It has been shown that commercially available fibers can correct the relative drift of the payload wavelengths that can occur in *HORNET* to within 10 ps/(km of SMF).¹ If 1310 nm is used for the control channel wavelength, DCF is not sufficient to keep the SOF indicator aligned with the packets on the payload wavelength. However, the solution is simple because the control channel wavelength is separated from the payload wavelengths in every node. Fiber cable delays can be used to realign the control channel wavelength and the payload wavelengths.

4.C. Control Channel Frame Synchronization Protocol

In every *HORNET* node, the control channel is processed and retransmitted while packets on the payload wavelengths pass through an all-optical path. The control channel must be retransmitted in perfect alignment with those packets. However, two issues can prevent that from happening. The first issue is a lack of synchronization between the incoming and the retransmitted control channel at each node, while the second issue is the difficulty in manufacturing a node with a perfect match

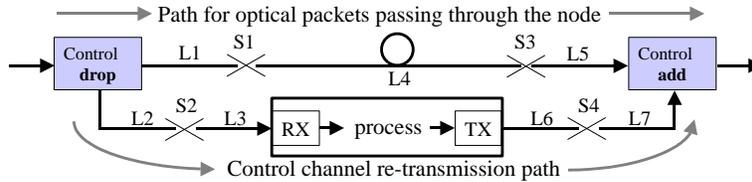


Fig. 10. The control channel path and the payload wavelength path. S_n denotes splice locations, L_n denotes fiber lengths.

between the payload path and the control channel path. Both of these issues are solved with the establishment of a frame synchronization protocol for *HORNET*.

The node's control process is in general not perfectly synchronized with the incoming control channel, and thus the process will begin at a random moment with respect to the moment of arrival of the SOF indicator, which drives the control process. Within each node, the random time difference between the actual arrival of the SOF indicator and the detection of the indicator is uniformly distributed across one clock cycle (the node uses a byte-clock). The random misalignment adds stochastically at each node, resulting in a large variance after several nodes of propagation. It can be shown that after an optical packet propagates through 32 nodes, there is a probability of 0.001 that it will be misaligned from the control channel SOF indicator by at least 11 control channel bytes.¹ However, this issue can be easily solved by using a phase-locked loop (PLL) within the control channel receiver to synchronize the control channel process with the incoming control channel bit stream. Thus, the first requirement of the frame synchronization protocol is the use of a PLL to obtain synchronization from the incoming control channel.

The second issue that causes control channel frame misalignment is designing, manufacturing, and maintaining a perfect match in propagation delay between the control channel path and the payload wavelength path. Figure 10 illustrates the two paths, including splice locations. To make the paths match, splices and fiber lengths must be tightly controlled. More importantly, the design of the electronics and micro-code are critical because every modification in the design process and *every upgrade after the product release* may cause a path difference. Any error due to the micro-code will be present in every node, and thus the resulting misalignment will add as packets traverse the ring.

This issue is solved in the frame synchronization protocol by *automatically* calibrating the control channel path propagation delay to match the payload wavelength path. Figure 11 shows the important components involved in the calibration. The two highlighted components, the PLL phase selector and the delay states, are used to adjust the propagation delay through the control channel path. The node programs the delay states to adjust the propagation delay in increments of a process clock cycle. Also, the node can control the output phase of the PLL, which dictates the moment at which the incoming SOF indicator comparator output flag is sampled. Sampling the SOF comparator output flag near the beginning of its duration will shorten the propagation delay of the control channel path, just as sampling the flag near its end will lengthen the propagation delay.

The calibration requires two steps to achieve nearly perfect SOF indicator alignment. The first step is a laboratory calibration (*lab cal*) to put the node in a position to perform its auto alignment when in the system. This is a manual step performed by an operator before the node is installed in the network.

The lab-cal system setup is shown in Figure 12. The operator arranges the *lab cal*

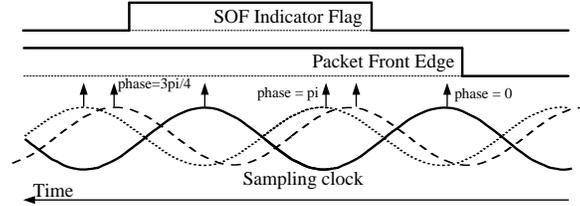


Fig. 13. During the *IS-cal*, the node measures the time difference between the arrival of the SOF indicator flag and the packet front edge by cycling its process clock phase through all possible phases.

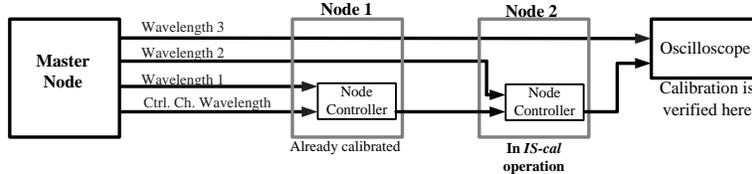


Fig. 14. The setup of the *IS-cal* procedure for a node downstream of a previously calibrated node.

4.D. Frame Synchronization Demonstration

An experimental testbed was assembled to demonstrate the *HORNET* frame synchronization calibration procedure.¹ As shown in Figure 14, three experimental *HORNET* nodes are connected together. The nodes use a PLL with an adjustable output phase to synchronize the control process with the incoming control channel, as specified by the protocol. Gigabit Ethernet (GbE) is used for the control channel, and thus the SOF indicator is the GbE ‘comma’ byte (*1100000101*). The lab cal procedure was performed on the nodes to set the reference condition. The IS-cal was then performed on Node 1. The IS-cal of Node 2 is described below.

The phase of the local clock in Node 2 was cycled through 8 phases. The measured alignment of the packet front edge and SOF indicator with the different phases of the sampling clock is shown in Figure 15. For this node, the samples result in a difference of one cycle for phases of 0 , $\frac{\pi}{4}$, $\frac{\pi}{2}$, $\frac{3\pi}{4}$, π , and $\frac{5\pi}{4}$. For the the phases of $\frac{3\pi}{2}$ and $\frac{7\pi}{4}$, the difference is zero. Thus, the node determines that the phase needs to be advanced by $\frac{6}{8}$ of a clock cycle. However, note that if the node advances (subtracts) its phase by $\frac{6}{8}$ of a clock cycle, the outgoing SOF indicator is actually *delayed* by $\frac{2}{8}$ of a cycle. This always occurs when the SOF indicator bit boundary is crossed (i.e. when the sampled difference changes from one to zero clock cycles). Thus, when the node determines that the boundary was crossed (as revealed in this example by the samples for $\frac{3\pi}{2}$ and $\frac{5\pi}{4}$, the node also subtracts one logical delay state. Figure 16 shows the result of the experimental demonstration. Figure 17 compares the alignment of the SOF indicator and a packet after two nodes of propagation with and without the frame alignment protocol. The time-lapse image of Figure 17 (a) shows the random misalignment that occurs without the protocol.

The results of this experiment shown in Figures 16 and 17 show that the alignment accuracy is within *one bit* of the control channel bit rate (1 ns in this case, since GbE is used). This is because the adjustment precision of the PLL is $\frac{1}{8}$ of a clock cycle, or one bit. In general, the accuracy may only be as good as a few bits because of the possibility that the correct alignment would have the clock sampling the ‘edge’ of the SOF indicator (in such a acase the sampling clock is adjusted slightly). As long as the accuracy is within *one byte*, then only one byte of guard

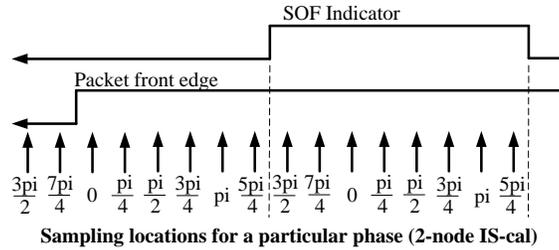


Fig. 15. The location of the samples for all phases for the two incoming waveforms in the *IS-cal* procedure of the second node.

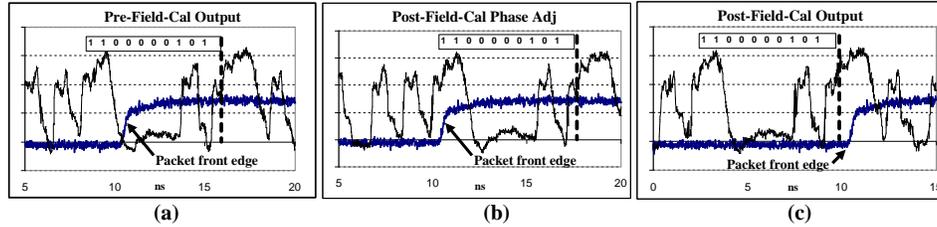


Fig. 16. (a) Alignment of retransmitted control channel SOF indicator with a packet passing through the node before the *IS-cal*; (b) After the phase adjustment portion of the *IS-cal*; (c) After the complete *IS-cal*.

band is necessary, and thus only one byte of overhead is used.

5. Summary

The *HORNET* architecture utilizes fast-tunable transmitters and wavelength routing to cost-effectively scale beyond 1 Tb/s. A control-channel-based MAC protocol has been designed for *HORNET*. The MAC protocol is optimized for IP packets by using the novel SAR-OD protocol, and it provides fairness control through the use of the novel DQBR protocol. A custom-designed simulator verified that the DQBR protocol provides excellent fairness control for the *HORNET* architecture.

The control channel design for the MAC protocol is thoroughly described in this work. It was shown through simulations that the optimal control channel frame size is 64 bytes, when considering a good estimate of IP packet sizes. Also, two causes of control channel frame misalignment were analyzed. As was discussed, the misalignment can cause collisions, which decreases the efficiency of the network. Since the network wavelengths cover a large WDM spectrum, the dispersion of

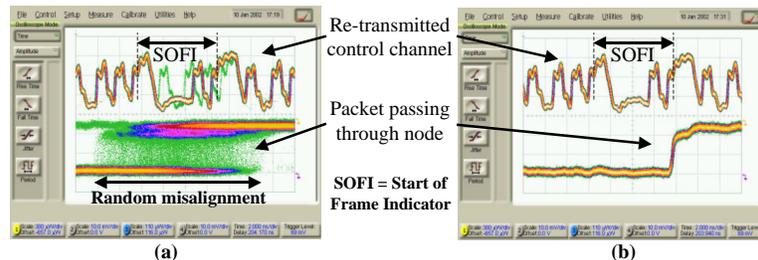


Fig. 17. Time-lapse image of the retransmitted control channel and packets after two nodes of propagation. (a) Random misalignment with no frame synchronization protocol. (b) Perfect alignment with the protocol.

SMF causes the packets on the payload wavelengths to become misaligned with the control channel frames. It has been shown that careful optimization of DCF cables in each node can eliminate this problem.

The second cause of control channel frame misalignment is the unavoidable path length mismatch between the control channel path and the payload wavelength path. A calibration routine was designed to automatically correct the mismatch. The node can reprogram itself to adjust the propagation delay of the control channel through the node so that it nearly exactly matches the propagation delay of the payload wavelengths through the node. As was *experimentally demonstrated* in this work, the protocol keeps alignment of the control channel to within *one bit*, which keeps the necessary guard band (overhead) to well within one byte.

The use of the control channel in *HORNET* enables a low cost MAC that is optimized for variable-sized packets and that provides fairness control. The design of the control channel for the *HORNET* architecture allows the implementation of the MAC protocol to be efficient and inexpensive. It is one of the most important factors in making *HORNET* a cost-effective and practical architecture for high-capacity next-generation metro networks.

6. Acknowledgments

This work was funded by The Defense Advanced Research Projects Agency and by Sprint Advanced Technology Laboratories.

References and Links

1. I. M. White. *A New Architecture and Technologies for High-Capacity Next Generation Metropolitan Networks*. PhD dissertation, Stanford University, Department of Electrical Engineering, August 2002.
2. I. M. White, M. S. Rogge, Y-L. Hsueh, K. Shrikhande, and L. G. Kazovsky. "Experimental demonstration of the HORNET survivable bi-directional ring architecture," in *Optical Fiber Communications Technical Digest*, Anaheim, CA pp. 346–349, March 2002.
3. D. Wonglumsom, I. M. White, S. M. Gemelos, K. Shrikhande, and L. G. Kazovsky. HORNET - a packet-switched WDM metropolitan area ring network: Optical packet transmission and recovery, queue depth, and packet latency. In *1999 IEEE LEOS Annual Meeting Conference Proceedings*, pages 653–654, November 1999.
4. Y. Tamir and G. Frazier. High performance multi-queue buffers for VLSI communication switches. In *Proceedings of the 15th Annual Symposium on Computer Architecture*, pages 343–354, June 1988.
5. National Laboratory for Applied Network Research, Measurement Operations and Analysis Team. <http://pma.nlanr.net/Datacube/>.
6. IEEE Standard 802.6. Distributed queue dual bus (DQDB) subnetwork of a metropolitan area network (MAN), December 1990.
7. E. L. Hahne, A. K. Choudhury, and N. F. Maxemchuk. DQDB networks with and without bandwidth balancing. *IEEE Transactions on Communications*, 40(7):1192–1204, July 1992.



ELSEVIER

Information Sciences 149 (2003) 135–149

INFORMATION
SCIENCES
AN INTERNATIONAL JOURNAL

www.elsevier.com/locate/ins

Design and performance evaluation of scheduling algorithms for unslotted CSMA/CA with backoff MAC protocol in multiple-access WDM ring networks

Kyeong Soo Kim ^{a,*}, Leonid G. Kazovsky ^b

^a Stanford Networking Research Center, Packard Building, Room 073, Stanford, CA 94305, USA

^b Optical Communications Research Laboratory, Packard Building, Room 362,
Stanford, CA 94305, USA

Received 27 September 2001; accepted 8 May 2002

Abstract

The unslotted *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) with backoff is a fully distributed, asynchronous *Media Access Control* (MAC) protocol for multiple-access *Wavelength Division Multiplexing* (WDM) ring networks with simplicity and robustness comparable to those of Ethernet [IEEE J. Select. Areas Commun. 18 (10) (2000) 2004; Proceedings of GLOBECOM'00, vol. 2, 2000, p. 1303]. In this paper, we present the results of performance evaluation of four scheduling algorithms – *Random Select* (RS), *Destination Priority Queueing* (DPQ), *Longest Queue First* (LQF), and *Shortest Packet First* (SPF) – designed for the unslotted CSMA/CA with backoff MAC protocol to address the issues of fairness and bandwidth efficiency. Through extensive network-level simulations for a multiple-access WDM ring with 10 nodes and 10 wavelengths on a 100 km ring, we have verified that under uniform traffic condition, the LQF with optical buffer size of 13 and 78 octets shows the best performance in terms of fairness, guaranteeing full fairness (fairness index ≈ 1) for arrival rates up to 9.5 Gbps/node, while for throughput and packet delay, the DPQ with the maximum optical

* Corresponding author. Fax: +1-561-594-2474.

E-mail addresses: kks@stanford.edu (K.S. Kim), kazovsky@stanford.edu (L.G. Kazovsky).

¹ He is with the Advanced System Technology, STMicroelectronics.

buffer size of 1538 octets gives the best results. We have also identified that the optical buffer size greatly affects the performance of nonrandom scheduling algorithms.

© 2002 Elsevier Science Inc. All rights reserved.

Keywords: Scheduling; Unslotted CSMA/CA with backoff; MAC; RS; LQF; DPQ; SPF; WDM; Ring networks

1. Introduction

Transmission of *Internet Protocol* (IP) packets over *Wavelength Division Multiplexing* (WDM) layer has been gathering tremendous interest among the optical networking community due to its simplicity and low overhead, resulting from the elimination of intermediate layers like *Asynchronous Transfer Mode* (ATM) and *Synchronous Optical Network* (SONET). Among various network architectures available for the IP over WDM, the multiple-access ring is considered one of the most promising and economical network architectures for future optical *Metropolitan Area Networks* (MANs). In the multiple-access ring architecture, it is essential to design *Media Access Control* (MAC) protocols that are efficient in allocating bandwidth with guaranteed fair access to all nodes on the ring.

Unslotted *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) with backoff has been proposed as one of MAC protocols for IP-HORNET – IP version of *Hybrid Optoelectronic Ring Network* [1,2]. The unslotted CSMA/CA with backoff has two unique features as an optical MAC protocol: First, it is a fully distributed, asynchronous protocol that does not need a centralized controller or a separate control wavelength to harmonize and synchronize the operations of nodes on a ring. Second, it can naturally support variable length IP packets without complicated segmentation and reassembly, which becomes harder as the line speed of optical wavelengths ever increases.

These features make the unslotted CSMA/CA with backoff MAC protocol very simple and scalable. For actual implementation of the protocol, however, important issues including fairness scheduling and effects of implementation parameters like optical buffer size are to be fully investigated.

Especially, design of fair and efficient scheduling algorithms is critical due to the inherent unfairness in the multiple-access optical ring (or bus) network. Because of unidirectional transmission of signal on the optical ring, the incoming frames from upstream nodes take priority over outgoing frames at a node. Hence, there arises the so-called *positional priority* problem where for a given destination and the corresponding wavelength, access nodes far from the destination node have higher priorities over those closer to destination node. Therefore without proper scheduling that counteracts this unfairness, the experienced quality of service of a connection at a node is highly dependent upon

the relative position of the node with respect to its destination. In addition to fairness guarantee, scheduling algorithms should be efficient in use of available bandwidth, which means they should provide good overall throughput.

In this paper, we report the design of scheduling algorithms for the unslotted CSMA/CA with backoff MAC protocol to address the issues of fairness and bandwidth efficiency in the multiple-access ring network and the results of performance evaluation through extensive network-level simulations. We also investigate the effect of optical buffer size on the performance of the scheduling algorithms, the buffer size being one of the critical implementation parameters.

The rest of the paper is organized as follows: We first review the unslotted CSMA/CA with backoff MAC protocol in Section 2 and describe the scheduling algorithms designed in Section 3. In Section 4, we present simulation results with discussions. Section 5 summarizes our work.

2. Unslotted CSMA/CA with backoff MAC protocol

Carrier sense and collision avoidance operations are depicted in Fig. 1.

The access node listens to all wavelengths by monitoring either sub-carriers [1] or baseband optical signals [3], depending on the implementation. When a frame is ready for transmission, the access node checks the occupancy of the

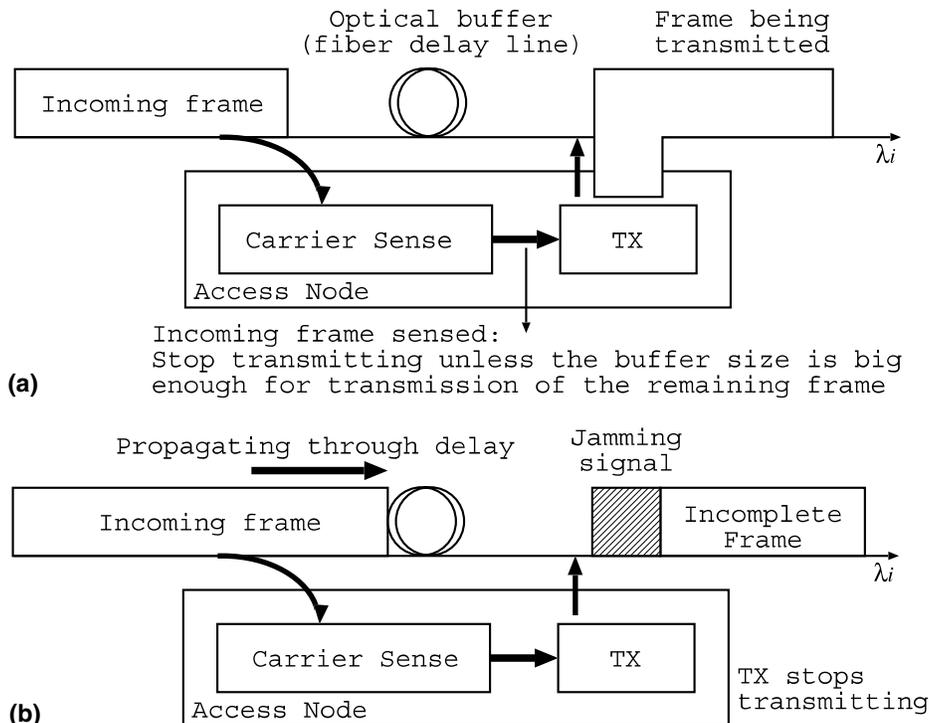


Fig. 1. Unslotted CSMA/CA with backoff: (a) carrier sense; (b) collision avoidance.

target wavelength. If it is free at that instant, the access node begins to transmit the frame. However, since the access node cannot know if the opening is long enough to accommodate the entire frame, it continues to monitor the wavelength. A small ‘fixed’ optical delay line (i.e., optical buffer) is placed between the point at which the node listens for incoming frames and the point at which the node inserts new frames. This allows the node to terminate its transmission before the frame interferes with the frame already on the ring. If it detects a frame arriving on the same wavelength at its input and the size of optical buffer is not big enough for successful transmission of the remaining frame with a guard band, it immediately interrupts the frame transmission and sends a jamming signal. Otherwise, it can finish the transmission of the entire frame without interruption. Note that the optical buffer size at least should be large enough for transmitting the jamming signal and the guard band before the incoming frame. The jamming signal (like in Ethernet 10/100 Base-T) could be a unique bit pattern, either at baseband or on the sub-carrier. The downstream access node recognizes the incomplete frame by the presence of the jamming signal and pulls it off the ring. The access node can reschedule the transmission of the frame for a later time.

3. Scheduling algorithms for unslotted CSMA/CA with backoff MAC protocol

3.1. Random Selection (RS) scheduling

The MAC implementation has a *Virtual Output Queue* (VOQ) for each wavelength. The RS algorithm maintains a list of empty wavelengths and corresponding “*non-empty*” VOQs. It then randomly selects a VOQ out of the list for transmission. This scheme is fairly simple and has no counter-measure for the unfairness, but we use it as a reference algorithm in performance evaluation of scheduling algorithms.

3.2. Longest Queue First (LQF) scheduling

Because of the positional priority, VOQs for those wavelengths whose destinations are closer downstream are likely to have more frames than others. We counteract this problem by giving priorities to those wavelengths with longer VOQs to guarantee fairness. In the LQF scheduling, if wavelengths are available for transmission, the scheduler selects the one with longest VOQ among them.

3.3. Destination Priority Queueing (DPQ) scheduling

The DPQ scheduling algorithm tries to achieve the fairness by giving priorities to wavelengths based on their destinations rather than based on the

length of VOQs. In this scheduling the wavelengths whose destinations are closer downstream are given higher priorities in order to compensate the effect of the positional priority. Compared to the LQF scheduler, the DPQ scheduler can be more easily implemented because the DPQ uses only destination information of wavelengths in scheduling, which does not change once a network topology is fixed, while the LQF resorts to VOQ length that is continuously changing at each scheduling instant.

3.4. Shortest Packet First (SPF) scheduling

While the fairness guarantee is the number one priority in designing the LQF and the DPQ scheduling algorithms, in the SPF scheduling we are trying to maximize bandwidth efficiency by giving priority to wavelengths that have shorter frames in the VOQs. The rationale behind the SPF algorithm is that by sending shorter frames first, it would be possible to reduce the chance of being interrupted by incoming frames.

4. Performance evaluation of scheduling algorithms

4.1. Simulation model and operational assumptions

We have developed simulation models for the performance evaluation of the scheduling algorithms based on *Objective Modular Network Testbed in C++* (OMNeT++) [4]. The OMNeT++ is a discrete-event-driven simulator based on C++ and supports models of hierarchically nested modules with multiple links between them, which is an essential feature for the simulation of WDM systems.

The simulation model is for a multiple-access ring network with HORNET architecture, consisting of 10 access nodes with 10 wavelengths on a 100 km ring, where each node on the ring receives frames through a fixed wavelength, but can send frames any wavelengths available through a tunable laser. IP packets are generated with packet size distribution matching that of a measurement trace from one of MCI's backbone OC-3 links [5] and uniform destination distribution. Although the packet generator can generate packets based on either Poisson process or *Interrupted Poisson Process* (IPP), we report only the results based on Poisson process due to space-limit in this paper.

The MAC parameters used are summarized in Table 1.² Note that the optical buffer size of 13 octets corresponds to the minimum required for the transmission of interrupted frame, while 78, 590, and 1538 octets are the buffer

² We adopt parameters from 10 Gigabit Ethernet for frame format (overhead) and interframe gap time (guard band) due to its similarity to the unslotted CSMA/CA with backoff MAC protocol.

Table 1
Unslotted CSMA/CA with backoff MAC parameters

Parameter	Value
Line speed	10 Gbps
Overhead	26 octets
Guard band	12 octets (= 9.6 ns)
Jamming signal	1 octet
Optical buffer size	13, 78, 590, 1538 octets
VOQ size	1e5 octets

sizes for successful transmission of frames with size up to 66, 578, and 1526 octets, respectively, in the worst case that an incoming frame is detected just after the beginning of frame transmission. These are the frame sizes for the popular IP packet sizes: 40, 552, and 1500 octets.

The following performance measures are used: (1) Throughput per node, (2) fairness index [6], and (3) average end-to-end packet delay. Throughput per node is defined as total number of bits delivered during the simulation divided by the product of simulation time and the number of nodes. The fairness index is used to better quantify the fairness of each scheduling algorithm and based on the throughput of all the connections on the network.

4.2. Simulation results

We show simulation results of the scheduling algorithms with optical buffer sizes of 13, 78, 590, and 1538 octets in Figs. 2–13.

From the results, we first identify that the optical buffer size greatly affects the performance of all nonrandom scheduling algorithms (i.e., LQF, DPQ, and SPF) especially at a higher traffic region with larger than 4 Gbps/node of arrival rate. On the other hand, the effect of optical buffer size on the performance of RS is relatively small.

In terms of fairness, LQF with optical buffer size of 13 and 78 octets is the best, achieving fairness index of 1 for all arrival rates up to 9.5 Gbps/node. For throughput and delay, DPQ with the maximum optical buffer size of 1538 octets shows the best performance. Note that with optical buffer size of 1538 octets there is no incomplete frame resulting from transmission interruption, which is the main cause of bandwidth waste in unslotted CSMA/CA with backoff MAC. For DPQ, the results show that the benefit of no incomplete frame outweighs disadvantage of the wasted bandwidth due to large gaps in the 1538-octet optical buffer.

Figs. 11 and 12 indicates that the improvement in throughput performance of DPQ with 1538-octet optical buffer is closely related with abnormal change in the fairness index around the arrival rate of 6.5 Gbps. To investigate this change in fairness index with related throughput performance, we analyze the

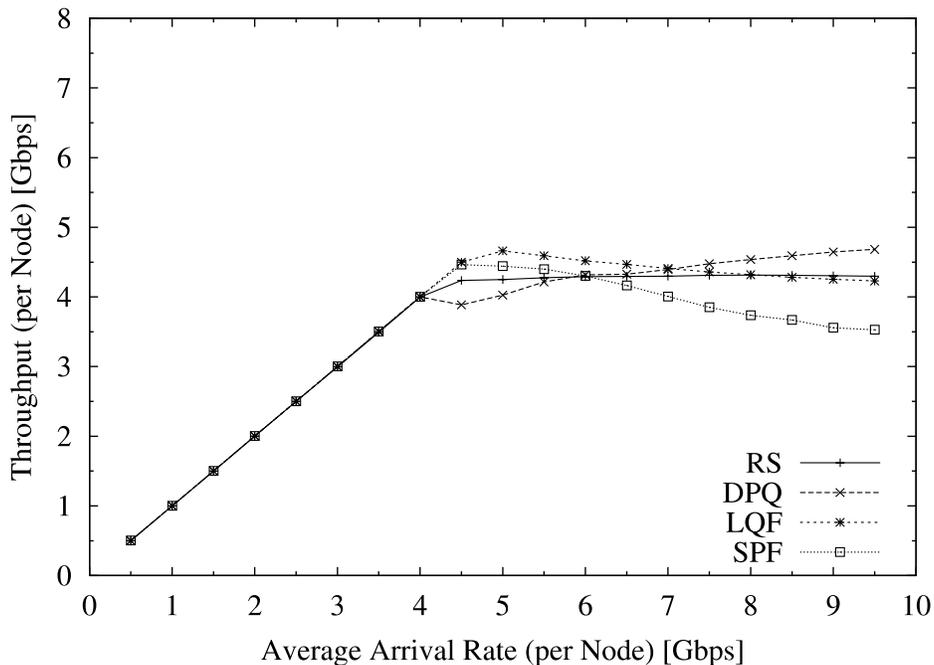


Fig. 2. Throughput per node of designed scheduling algorithms for optical buffer size of 13 octets.

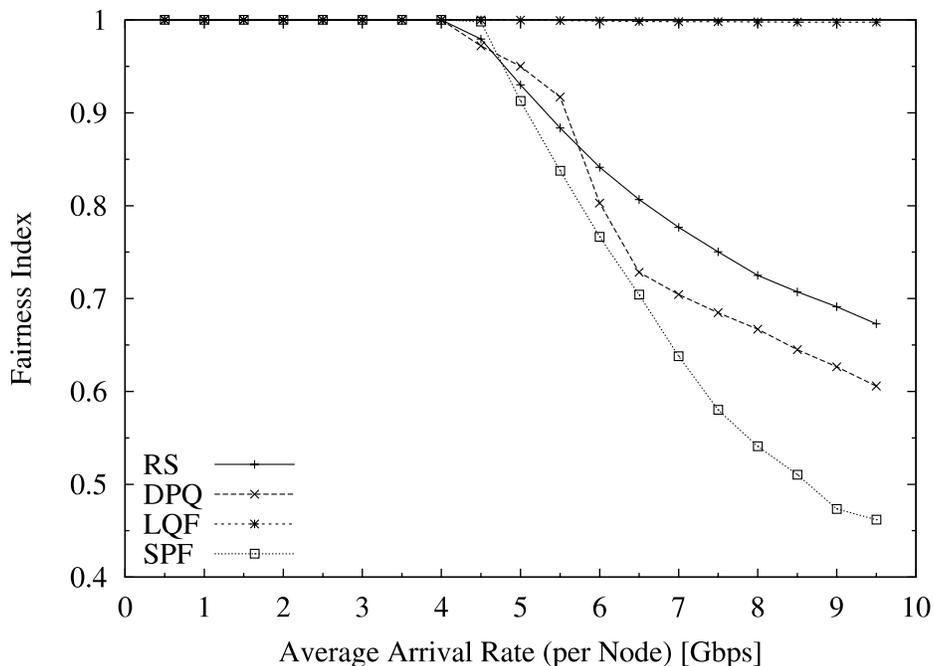


Fig. 3. Fairness index of designed scheduling algorithms for optical buffer size of 13 octets.

average frame access delay. The frame access delay is defined as the time spent from the moment a scheduler selects a channel and moves the first frame from

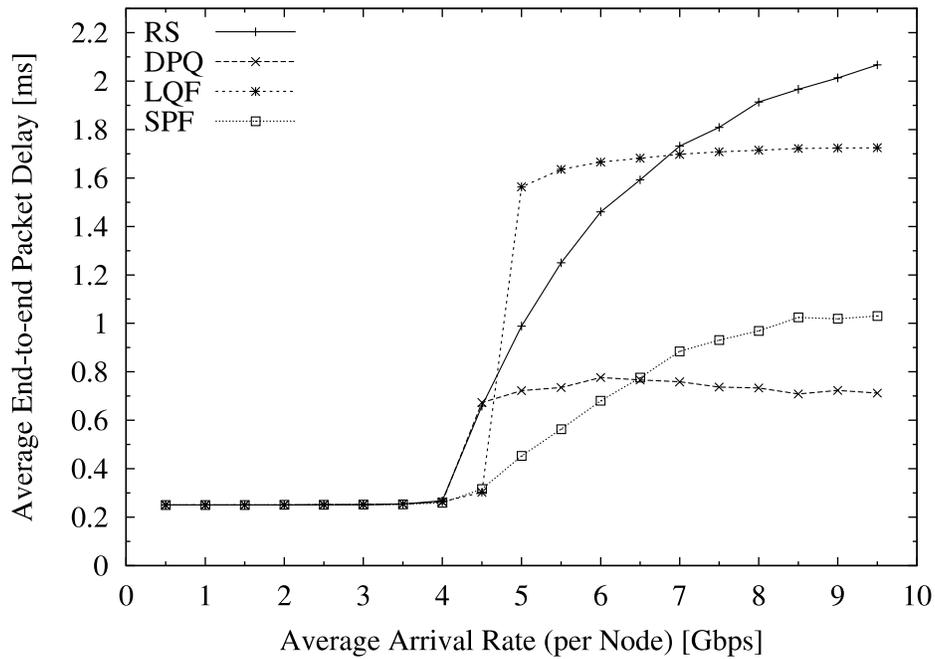


Fig. 4. Packet end-to-end delay of designed scheduling algorithms for optical buffer size of 13 octets.

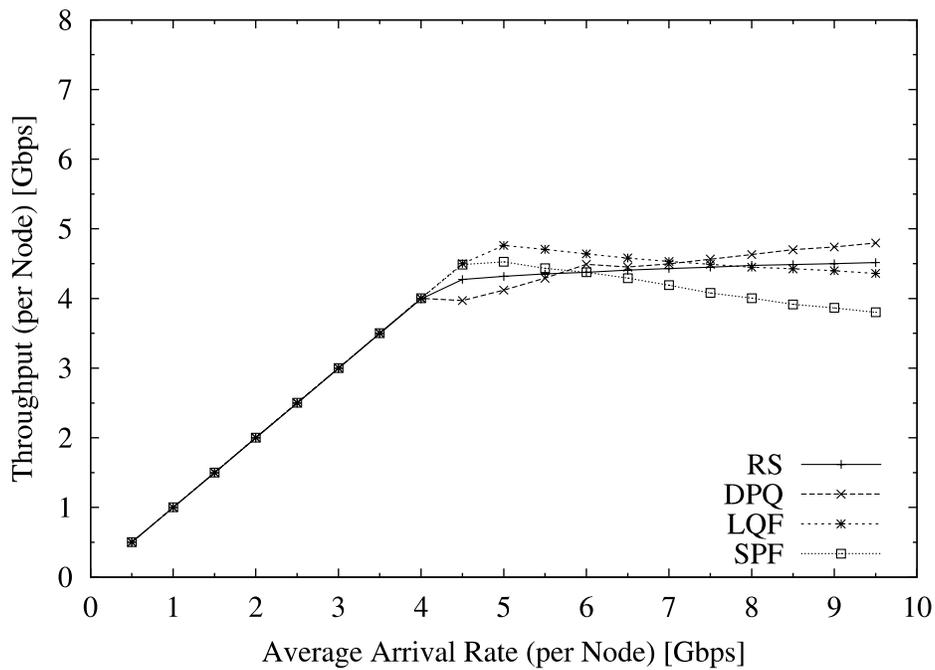


Fig. 5. Throughput per node of designed scheduling algorithms for optical buffer size of 78 octets.

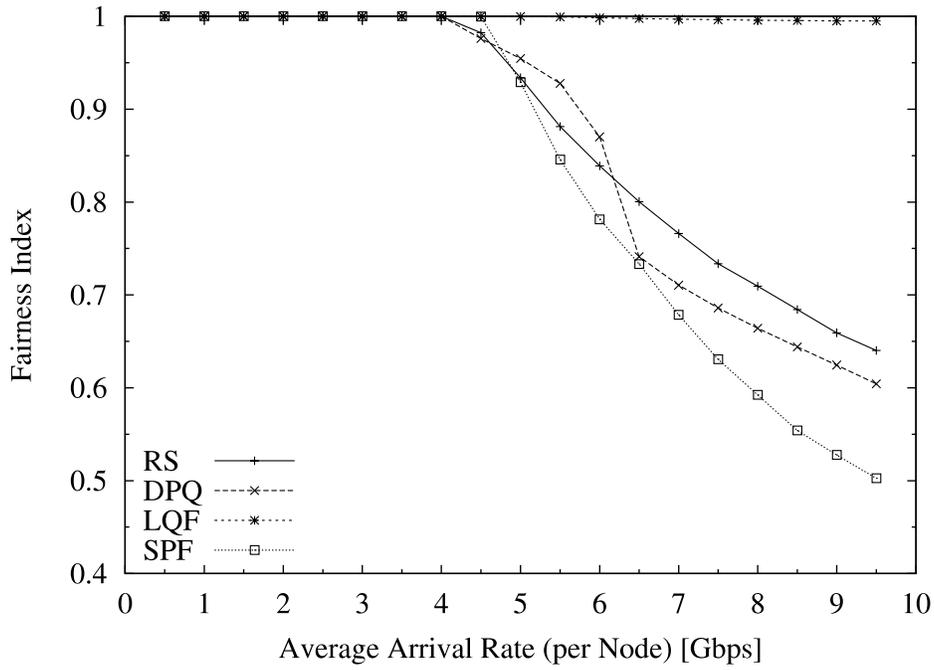


Fig. 6. Fairness index of designed scheduling algorithms for optical buffer size of 78 octets.

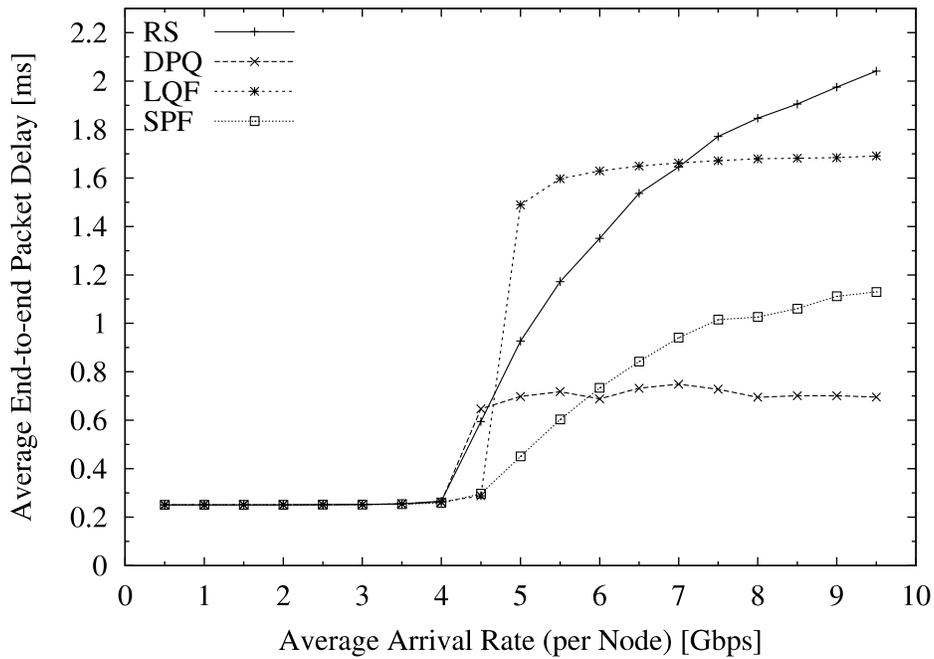


Fig. 7. Packet end-to-end delay of designed scheduling algorithms for optical buffer size of 78 octets.

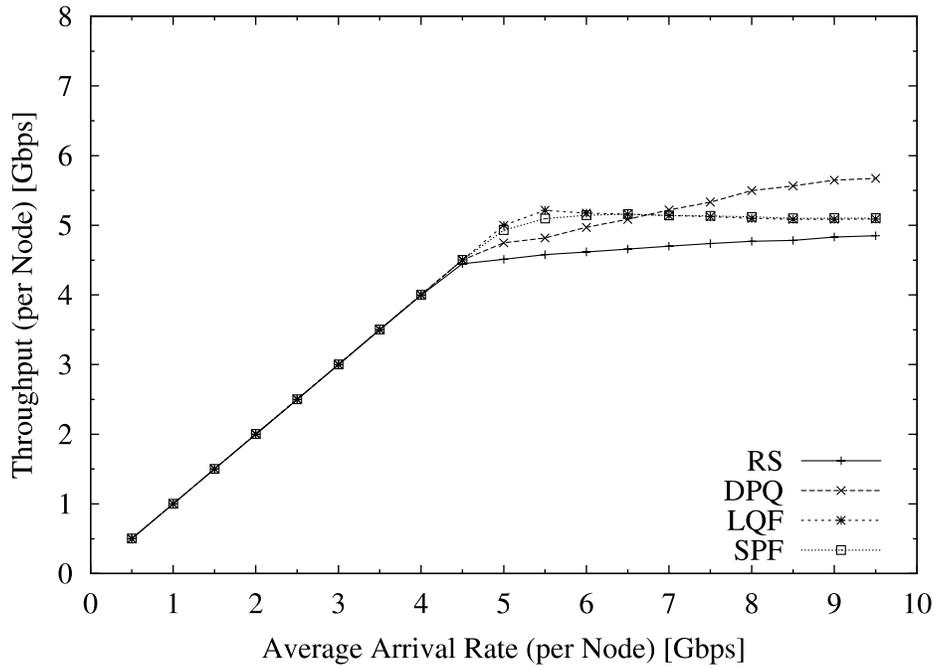


Fig. 8. Throughput per node of designed scheduling algorithms for optical buffer size of 590 octets.

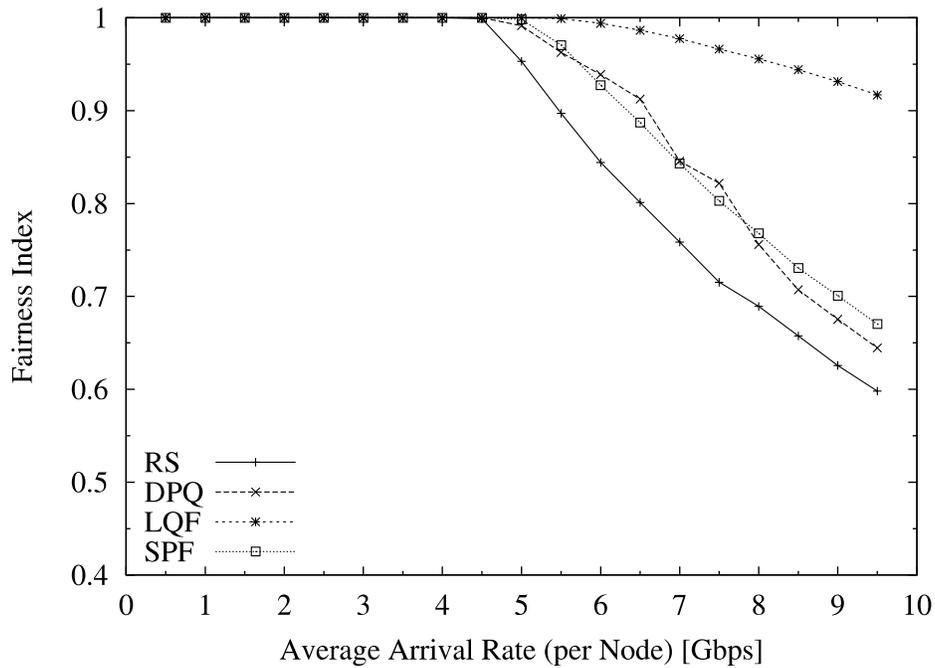


Fig. 9. Fairness index of designed scheduling algorithms for optical buffer size of 590 octets.

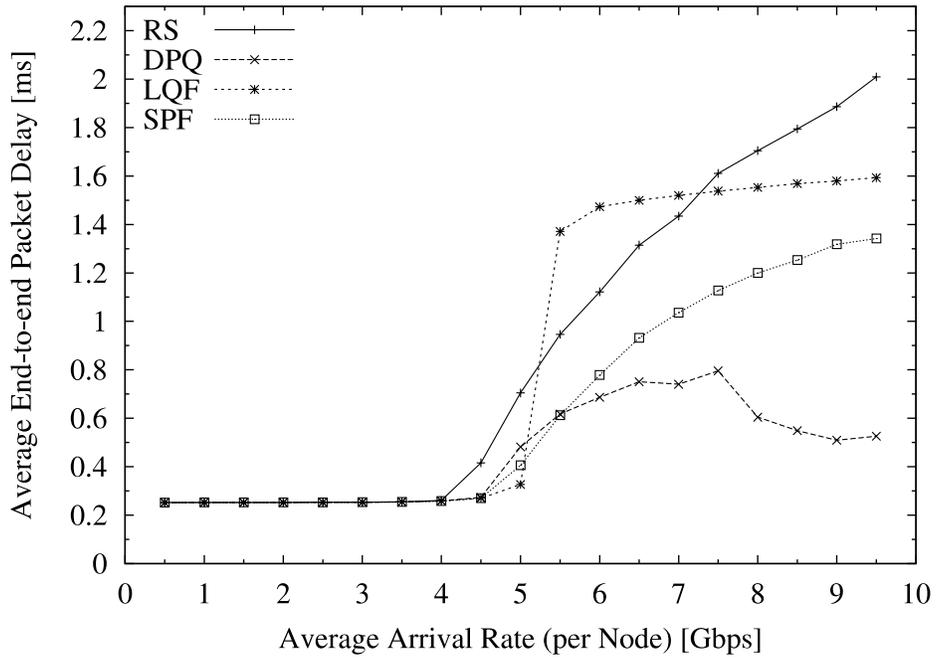


Fig. 10. Packet end-to-end delay of designed scheduling algorithms for optical buffer size of 590 octets.

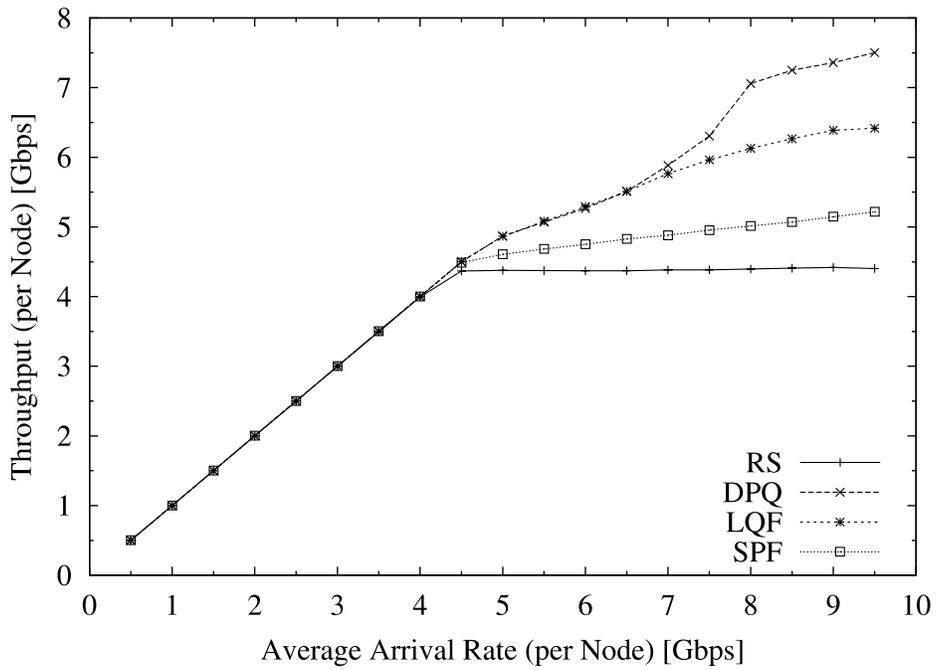


Fig. 11. Throughput per node of designed scheduling algorithms for optical buffer size of 1538 octets.

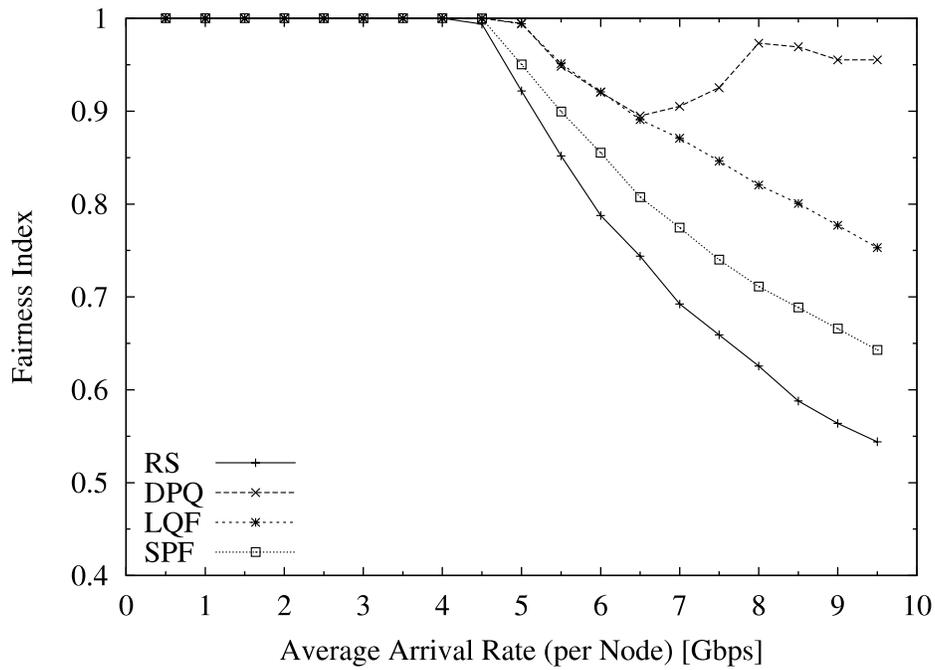


Fig. 12. Fairness index of designed scheduling algorithms for optical buffer size of 1538 octets.

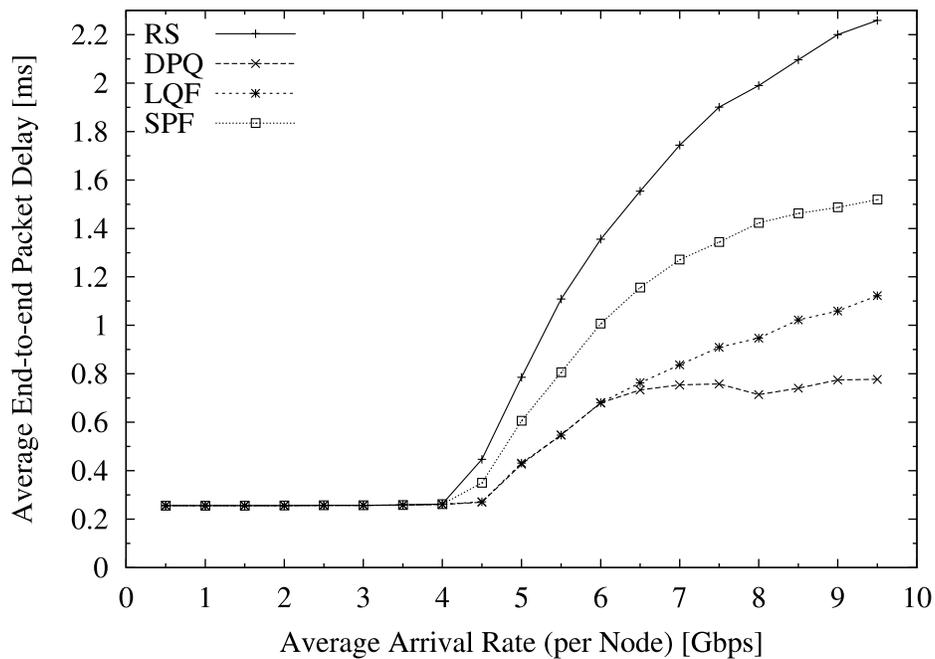


Fig. 13. Packet end-to-end delay of designed scheduling algorithms for optical buffer size of 1538 octets.

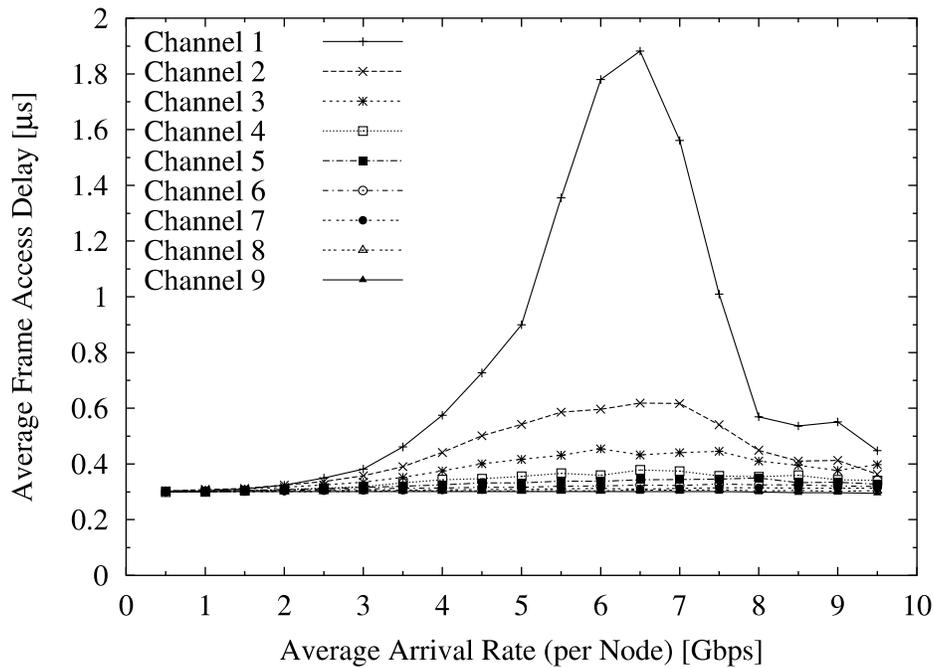


Fig. 14. Average frame access delay at node 0 for DPQ scheduling with optical buffer size of 1538 octets.

the channel VOQ to a transmission buffer until the last bit of the frame leaves the transmission buffer as a result of successful transmission. Fig. 14 shows the average frame access delay for each channel at node 0. It is clear from the figure that the access delay for channels with lower positional priorities increases as the arrival rate goes up until it reaches 6.5 Gbps.³ Further increasing of the arrival rate, however, results in decrease of the access delay for channels with lower positional priorities. It can be explained that when the time node 0 stays on those channels with lower positional priorities, especially on channel 0, reaches a certain value such that resulting gaps on channels with higher positional priorities at node 0 (channels 9, 8, ...) can accommodate even transmission of the longest frames, downstream nodes are forced to select those channels by DPQ scheduling, which in turn gives node 0 higher chance of frame transmission on channels with lower positional priorities without interruption,

³ In simulations each node, after selecting a transmission channel, waits for a certain amount of time before starting its frame transmission, which models nonzero wavelength tuning time of actual tunable lasers, and for simplicity, we assume that it is equal to the guard band time. Due to this nonzero tuning time, even with the maximum-size optical buffer, the frame transmission can be interrupted when it detects incoming frames before it completes wavelength tuning. In this case, however, no incomplete frame is generated and the frame transmission is simply rescheduled for later time, which is the reason for the increase of access delay in Fig. 14.

resulting in lower access delay in the end. This unusual behavior of DPQ with the maximum-size optical buffer is just one example revealing highly-complicated nature of unslotted CSMA/CA with backoff MAC protocol.

Note that the results in this paper strongly depend on packet size distribution and the operational assumption we take for handling optical buffer status. For example, if we keep track of all incoming frames in the optical buffer and use openings between them for frame transmission, the performance would improve even further with a bigger optical buffer. But this highly increases the implementation complexity, which eventually eliminates the benefits of the unslotted CSMA/CA with backoff MAC protocol.

5. Summary

In this paper, we have described four scheduling algorithms designed for the unslotted CSMA/CA with backoff MAC protocol and presented the results of the performance evaluation through extensive network-level simulations. From the simulation results, we have verified that under uniform traffic condition, the LQF with optical buffer size of 13 and 78 octets shows the best performance in terms of fairness, guaranteeing full fairness (fairness index ≈ 1), while for throughput and packet delay, the DPQ with the maximum optical buffer size of 1538 octets gives the best results. We have also identified that the optical buffer size greatly affects the performance of nonrandom scheduling algorithms, which also depends on packet size distribution and the operational assumption on the optical buffer handling.

Acknowledgements

The authors are greatly indebted to Ian M. White, Kapil Shrikhande, and other members of HORNET team at OCRL, Stanford, for their valuable discussions and suggestions for this work.

References

- [1] K. Shrikhande, I.M. White, D. Wonglumsom, S.M. Gemelos, M.S. Rogge, Y. Fukashiro, M. Avenarius, L.G. Kazovsky, HORNET: a packet-over-WDM multiple access metropolitan area ring network, *IEEE J. Select. Areas Commun.* 18 (10) (2000) 2004–2016.
- [2] K. Shrikhande, A. Srivatsa, I.M. White, M.S. Rogge, D. Wonglumsom, S.M. Gemelos, L.G. Kazovsky, CSMA/CA MAC protocols for IP-HORNET: an IP over WDM metropolitan area ring network, in: *Proceedings of GLOBECOM'00*, vol. 2, 2000, pp. 1303–1307.

- [3] E. Wong, S.K. Marks, M.A. Summerfield, R.D.T. Lauder, Baseband optical carrier-sense multiple access – Demonstration and sensitivity measurements, in: OFC 2001, Technical Digest Series, Anaheim, CA, 2001, WU2.
- [4] A. Varga, OMNeT++: discrete event simulation system, Technical University of Budapest, Version 2.1, March 2001.
- [5] WAN packet size distribution. Available from <http://www.nlanr.net/NA/Learn/packet-sizes.html>.
- [6] R. Jain, D. Chiu, W. Hawe, A quantitative measure of fairness and discrimination for resource allocation in shared computer systems, Technical Report DEC-TR-301, Digital Equipment Corporation, September 1984.

A MAC Protocol with Fairness Control for the HORNET Metro Network Architecture^{*}

Ian M. White^{*}, Kapil Shrikhande, Matthew S. Rogge, and Leonid G. Kazovsky

Stanford University Optical Communications Laboratory, 350 Serra Mall, MC9515, Stanford, CA 94305, USA

Abstract

The *HORNET* (Hybrid Opto-electronic Ring Network) architecture reduces infrastructure costs for next-generation high-capacity metro networks by utilizing fast-tunable packet transmitters, wavelength routing, and a novel MAC protocol. The MAC protocol is designed to accommodate variable-sized IP packets and to provide fairness control. In this work, the design of the MAC protocol is presented and then analyzed with a computer simulator developed for *HORNET*.

Key words: Metropolitan Area Network, Media Access Control, Packet-over-WDM, Optical Networking, Fairness, DQDB

1 Introduction

An evolution in metropolitan area networking has begun, and it will continue to change the Internet for many years to come. The current solution for delivering Internet content throughout the metro area evolved from telephony systems designed to transmit digitized voice circuits across an optical link. Thus, the technology is sub-optimal for the transport of bursty, packet-based Internetworking Protocol (IP) data traffic. New solutions for metro networks that implement a data optimized protocol stack over the ring-based fiber plant are now under development.

^{*} Funded by The Defense Advanced Research Projects Agency under agreement number F30602-00-2-0544, and by Sprint Advanced Technology Laboratories.

^{*} Corresponding author.

Email address: ianwhite@stanfordalumni.org (Ian M. White).

Looking into the not-so-distant future, it is clear that IP data traffic will continue to scale at a quick pace. To continue delivering the capacity demanded by Internet consumers, networks will have to scale using wavelength division multiplexing (WDM). Ultimately, metro networks will be forced to scale to capacities beyond 1 Tb/s . At capacities of this magnitude, the conventional architectures and the incoming generation of architectures require an excessive amount of optical-to-electrical and electrical-to-optical converters, high-speed line cards, and enormous packet switching capacity. Thus, a new architecture will be necessary to deliver greater than 1 Tb/s capacity while still allowing network operators to compete in the cost-sensitive market.

We have created a new IP-optimized metro networking architecture named *HORNET* that scales inexpensively beyond 1 Tb/s [1–4]. The architecture uses fast-tunable packet transmitters and wavelength routing to eliminate the need for excessive amounts of equipment. Along with a new architecture, the new protocols necessary for the architecture, such as a media access control (MAC) protocol, a fairness protocol, and a survivability protocol have been developed. The performance of the architecture and its protocols has been evaluated with custom-designed computer simulations. Additionally, the architecture and protocols have been implemented in a laboratory experimental testbed featuring fast-tunable packet transmitters [5,6].

This paper is organized as follows. Section 2 presents the motivation behind the development of the *HORNET* architecture. Section 3 describes the architecture of *HORNET*, and Section 4 presents the design of *HORNET*'s MAC protocol. The simulation results obtained with the custom-designed simulator are then given in Section 5. Section 6 presents the summary and the conclusions of the work presented in this paper.

2 Motivation

In the early days of photonics research for the Internet, the metropolitan area networks did not attract a lot of attention. Most companies and research institutes were focused on pushing the capacity of photonic links into the terabit per second (Tbps) realm. However, a noticeable shift occurred just before the turn of the century, as it became apparent that the ultra-high capacity backbone links would not necessarily be useful if a bottleneck existed in the metropolitan area between the Internet backbone and the user. The last few years of investment in metropolitan area networking has resulted in a few competing architectures aimed at cost-effective solutions that deliver moderate capacity. However, metropolitan area networks are only at the beginning of a major evolution towards a new age of end users and applications.

2.1 Next Generation Metropolitan Area Networks

A metropolitan area network of the near future will be characterized by the quantity and diversity of its end users, by the high percentage of randomly fluctuating packet-based data traffic, and by the incredible load placed on the network at peak usage times. End users may range from today's typical users, such as home and business users, to futuristic users such as automobiles, appliances, hand-held devices, and other things not yet imagined. It is no longer unthinkable for over a million users to simultaneously access the same metro network in the near future. With this many users, it is reasonable to believe that metro networks will be forced to support capacities of up to and beyond 1 Tbps . Additionally, it is safe to assume that a large portion of this traffic will be bursty, packet-based data traffic, as is common with the Internet.

Next-generation metro networks will largely be affected by new Internet trends. Two new trends emerging today will change the distribution of Internet traffic in the metropolitan area. A new technology called Web caching [7,8] is being used to place commonly accessed content closer to the end users, potentially in the metropolitan area network nodes. It helps to keep the load in the Internet more balanced and reduces download times for end users. Protocols have already been developed that allow networks, such as a metro network, to be aware of the content that all nodes in the network are caching, thus allowing the entire metro network to serve as a distributed cache [8]. The use of Web caching in a metropolitan network results in an increase in *intra-network* traffic. Adding to this effect is the new trend of distributed file and processor sharing. This Internet technology is most famous for the controversial exchange of music and video files, but has many other practical extensions as well. It is clear that this will also increase the amount of intra-network traffic. It is conceivable that new trends such as these and others will boost the level of intra-network traffic to the point where it is even a majority of the total network traffic.

In summary, next-generation metro networks will likely have the following characteristics. There will be millions of end users simultaneously accessing the network, resulting in more than 1 Tbps of load on the network. Traffic will be composed primarily of randomly fluctuating, bursty, packet-based data traffic, much of which may be intra-network traffic. Additionally, the market for metro network operators is much more competitive than that of Internet backbone operators, and hence the cost-effectiveness and efficiency of a network is crucial. Thus, a network architecture for next generation metropolitan area networks should *cost-effectively* support more than 1 Tbps of *bursty, packet-based data* traffic with *randomly distributed* source and destination node pairs.

2.2 Current Metro Solutions

Currently, SONET ring networks are the most popular solution for metropolitan networks. Despite this, however, there are several drawbacks to using SONET-based solutions for next-generation Internet traffic. First, the time-division-multiplexing (TDM) operation of SONET is wasteful of bandwidth in a networking environment featuring randomly fluctuating, bursty, packet-based data traffic. Second, provisioning new circuits in SONET requires far too much time because an excessive amount of planning and network management is necessary. This is unacceptable for today's and tomorrow's dynamic Internet world.

A third disadvantage is the wasted bandwidth and equipment that is used to maintain survivability. To protect *all* TDM time slots in the network (including unused time slots and best effort traffic), only half of the bandwidth and equipment in the network is utilized for working traffic. Another critical disadvantage in SONET is the high price of SONET ADMs as compared the new generation of high-speed data networking equipment (e.g. routers and Ethernet switches).

It seems logical to replace SONET networks with an architecture and protocols developed specifically for Internet data traffic. A perfect example of such an architecture and set of protocols is Ethernet. Ethernet has several primary advantages over the SONET architecture presented above. First of all, bandwidth utilization for statistically fluctuating traffic is much better in Ethernet. Secondly, provisioning new circuits is much simpler. Also, it is commonly accepted that Ethernet switches are much less expensive than SONET multiplexing equipment of the same capacity.

Given these advantages, it appears that Ethernet should be the new solution for the metro area. However, a few very important points have been neglected in this argument. Although Ethernet can be implemented over the currently deployed fiber ring infrastructure, it is not optimized for it. For example, Ethernet would not take advantage of the ring's survivable nature. Another drawback to using Ethernet for a next generation metro network is that Ethernet is not designed to handle quality of service or global network fairness issues [9]. However, these will be very important issues in metropolitan area networks.

Thus, it is fair to say that Ethernet is close to being an optimal solution for today's metro networks, but falls short because of its inability to take advantage of the ring topology and because of its disregard for fairness and quality of service. Thus, a modified version of Ethernet that is designed to compensate for these two shortcomings may be the best solution. It is this line

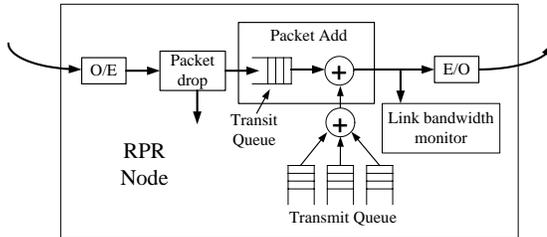


Fig. 1. Packet add/drop multiplexer in the RPR node. O/E = optical to electrical converter; E/O = electrical to optical converter.

of thinking that is behind the formation of the *IEEE Resilient Packet Ring (RPR) Working Group* and the *Resilient Packet Ring Alliance* [9]. RPR is a new data link layer protocol designed for metro area photonic ring networks. The RPR Working Group is attempting to copy the virtues of Ethernet and to utilize them in a metro area ring architecture with survivability, fairness, and quality of service (QoS).

RPR operates on a bi-directional optical ring network with a *packet add/drop multiplexer* (ADM) in each node. The packet ADM is illustrated in Figure 1. Packets that enter a particular node's input from the ring are either destined for that node or for a node further downstream. If the ADM determines that the packet is destined for its node, it drops the packet into the node. If the packet is not dropped, it is sent into the *transit* queue, which is a first-come-first-serve (FCFS) queue that holds the packets until they can be sent to the transmitter. Between the transit queue and the output transmitter is the add component of the packet ADM. Packets that are to be transmitted onto the network by a particular node are queued in the *transmit* queue waiting to be sent to the transmitter. An arbitrator uses the RPR fairness algorithm to determine when to send packets from the *transit* queue and when to send packets from the *transmit* queue. Note that the packet ADM has an advantage over a traditional Ethernet switch because the Ethernet switch does not distinguish between packets passing through the node and packets being inserted onto the network. Also, the RPR architecture is designed to be survivable without wasting bandwidth [9].

2.3 RPR-over-WDM

The initial deployment of RPR will likely use only one wavelength in each of the two fiber rings. However, it is clear that to support the quickly increasing demand for bandwidth in the metropolitan area, RPR will be forced to scale its capacity using WDM. Such an architecture is referred to in this work as *RPR-over-WDM*. As mentioned in Section 2.1, it is expected that in the near future metro networks will be forced to support capacities of up to or even beyond 1 Tbps. Obviously, at such high capacities, a large number of wavelengths will

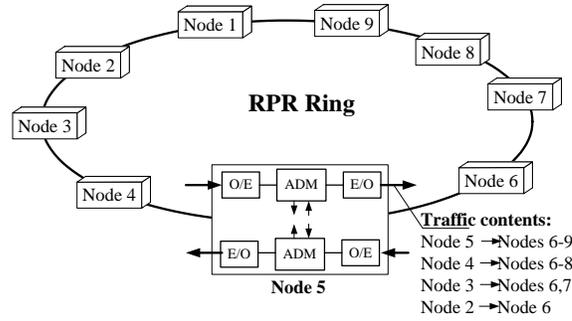


Fig. 2. In addition to receiving and transmitting their own traffic, RPR nodes must receive, switch, and re-transmit the traffic coming from upstream nodes and going to downstream nodes. O/E = optical to electrical conversion; E/O = electrical to optical conversion.

be required. This can be troublesome for *RPR-over-WDM* because if there are W wavelengths in each of the two rings, then every node will contain $2W$ receivers and $2W$ transmitters. Also, the packet ADMs must be designed to drop packets from W wavelength paths in each of the two directions, while the *transmit queue* should be designed to add packets on any of the W paths in each of the two directions. This is clearly expensive to design, especially considering the data path will likely be operating at 10 Gbps.

The cost of the equipment in a node becomes quite high when the capacity of the network must scale to the capacities of the near future because of the excessive amount of photonic transmitters and receivers, and because of the complexity of the packet ADM. However, when looking at the operation of the node, it becomes apparent that adding more intelligence into the network architecture design brings about a much more cost-effective solution. Notice that so much photonic equipment and electronic complexity is required within each node because the node is receiving, switching, and re-transmitting a lot of traffic that comes *from an upstream source* and is going *to a downstream destination*.

Consider the example of a 9-node bi-directional ring shown in Figure 2. Node 5 transmits traffic to Nodes 6, 7, 8, and 9 in the counter-clockwise direction. However, it also has to transmit traffic *from* Node 4 to Nodes 6, 7, and 8, and *from* Node 3 to Nodes 6 and 7, as well as traffic *from* Node 2 to Node 6. Thus, under uniform traffic conditions, only 40% of the traffic being transmitted by the node's transmitters came from this node. Additionally, only 40% of the packets coming through the packet drop stage were destined for this node. In fact, it can be shown that for a network with N nodes under uniform traffic conditions, only $\frac{4}{N+1}$ of the traffic transmitted by a node is originated by the node. Thus, for 25 nodes, only 15% of the traffic transmitted by the node was originated by the node. The other 85% of the traffic is only passing through.

Network inefficiency such as this is typically eliminated by utilizing *wavelength*

routing in the architecture. The cost of the node could be decreased significantly if traffic that originated upstream and is destined downstream would pass through all intermediate nodes optically. A node's photonic components would be required to operate on far less traffic, and thus could be reduced. The packet ADM would not process nearly as much traffic, and thus the complexity could be significantly reduced.

However, RPR is not designed to utilize wavelength routing. A typical wavelength routing implementation would have each node receive a uniquely assigned wavelength. When a node wants to transmit a packet to a particular destination node, the transmitting node inserts the packet using a transmitter that emits on the *wavelength assigned to the destination node*. This implies that the node has transmitters for every wavelength in the network, even though the node is only terminating traffic on one (or maybe a few) wavelengths. However, the RPR MAC protocol is only designed for the electronic packet ADM. The wavelength routing design requires a new MAC that controls an *optical packet ADM*. Unfortunately, an optical packet ADM similar to RPR's electronic packet ADM cannot be constructed because there is currently no practical optical queue, and thus there can be no *transit queue* for the optical signals passing *through* the node. As a result, the new MAC protocol would likely need to be more complex because of the shortcomings of the optical packet ADM. Ultimately, however, if a MAC protocol and an optical packet ADM can be designed that utilize wavelength routing advantageously, it is clear that the cost of the network would decrease tremendously.

3 ***HORNET***: The Next-Generation Solution

A new solution for metro networks that utilizes the advantages of wavelength routing can tremendously *decrease* the cost of a next-generation metro network. The solution requires a new method of transmitting packets that incorporates an *optical packet ADM*, as opposed to the electronic packet ADM proposed in RPR. A new MAC protocol also needs to be developed to control the optical packet ADM, as it differs significantly from the electronic packet ADM. The MAC protocol must efficiently support variable-sized packets and must be fair to all source-destination pairs.

These architectural requirements form the basis of the *HORNET* architecture. *HORNET*, which stands for Hybrid Opto-electronic Ring Network, utilizes fast-tunable packet transmitters, wavelength routing, and a novel MAC protocol to form an architecture that is more *cost-effective at high capacities* than any of its commercial predecessors. The generic design of the *HORNET* architecture is shown in Figure 3. Like its predecessors, *HORNET* is a bi-directional ring topology, meaning that it can use the currently deployed fiber infrastruc-

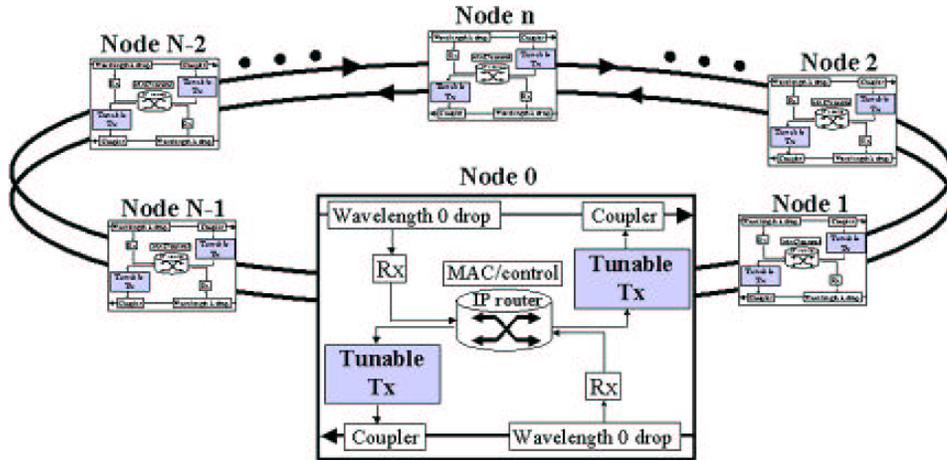


Fig. 3. The *HORNET* architecture is a bi-directional wavelength routing ring network with tunable transmitters in each node.

ture. Unlike SONET networks, however, *HORNET* uses all available bandwidth and equipment for working traffic without compromising survivability. Despite this, *HORNET* is still survivable, as described and demonstrated in [5].

As Figure 3 shows, nodes use fast-tunable packet transmitters to insert packets onto the ring. The packets are coupled optically onto the ring using a wideband coupler (currently, a fast-tunable wavelength-selective multiplexer is not available). A packet is transmitted on the wavelength that is received by the packet's destination node. A wavelength drop is used to drop one or more assigned wavelengths into each node. Thus, only the packets destined for a particular node are dropped into the node. All of the packets carried by the other wavelengths pass through optically, such that the node does not receive or process them. The *RPR-over-WDM* nodes require significantly more equipment because they have to receive, process, and re-transmit all packets that pass through. In *HORNET*, a node only needs enough equipment to process the packets to and from its local users.

HORNET is not the only project investigating next-generation metropolitan area ring networks. Several other projects [10–16] have also in recent years investigated WDM ring architectures for the metro area. Some of these projects use the same wavelength routing concepts that are used in the *HORNET* architecture. However, the survivability scheme and the MAC protocol developed for *HORNET* are unique. The novel MAC protocol developed for *HORNET*, which is optimized for variable-sized packets and provides fairness control, is described in detail in the following section.

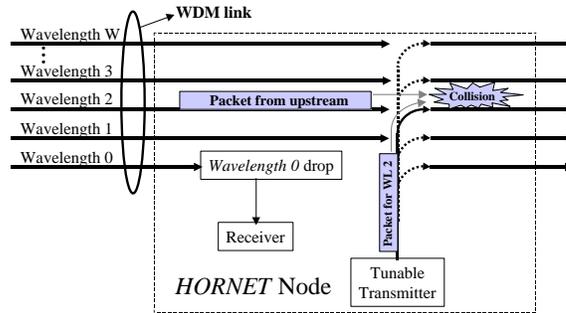


Fig. 4. A collision occurs when a transmitter inserts a packet on a wavelength that is currently carrying a packet through the node.

4 HORNET Media Access Control Protocol

Since the packet ADM process in the *HORNET* architecture is completely different from the ADM process of any preceding commercial network, a new MAC protocol must be developed. The primary function of the MAC protocol in *HORNET* is to prevent collisions at the point in the node where the transmitter inserts packets. Since the transmitter can insert a packet on any wavelength, and since most of the wavelengths are passing through the node without being terminated, a transmitter could insert a packet onto a particular wavelength that collides with another packet that is passing through the node on that wavelength. Figure 4 shows the occurrence of a collision. To prevent collisions, the MAC protocol should monitor the WDM traffic passing through the node, locate the wavelengths that are available, and inform the transmitter of which wavelengths it is allowed to use at a particular moment. As a result, the transmitter will not insert a packet on a wavelength that is currently carrying another packet through the node.

4.1 Potential Solutions

There are a few different design options for a MAC protocol that coordinates packet transmissions in order to avoid collisions. Two interesting possible designs involve treating the ring network as a packet switch, where the transmitter queues in each node are the *inputs* and coordinated time slots on the ring are outputs. The first packet switch emulator is a centralized design, while the second uses distributed control. In the centralized design, the nodes place requests to a scheduler for transmissions, where the scheduler is located in a master node. The scheduler determines the best schedule for all of the requests, and then the scheduler informs the nodes of when to transmit their packets. This appears to be a logical way to accomplish the MAC goals, but it is not practical because of the ultra-high-capacity and the large geographic area of the emulated switch.

The packet switch emulation design that uses distributed control eliminates the complicated scheduler. The nodes send requests for a transmission slot to the destination nodes and then wait for an acceptance. Only if they receive a positive acceptance will they send a packet. This MAC protocol attempts to copy the operation of a packet switch like the one discussed in [17]. In theory, this ensures that collisions do not occur, and that the network is fair to all users. This is certainly more attractive than the centralized scheduler, but the problem of the geographic size of the network remains. In general, this scheme requires a time-slotted environment in which the slot duration is equal to the propagation time of light around the optical ring (on the order of a *millisecond*). Thus, the time necessary for requests and acceptances adds at least a few milliseconds to the queuing delay. An excellent example of this competitive approach is thoroughly described in [11].

To avoid these problems, a MAC should be developed that uses only local information to make the decision about when to transmit. The most obvious solution for this method is for the node to monitor the optical power on each wavelength as the wavelengths pass through the node. If the node measures no power on a wavelength for the duration of a packet, then it concludes that the transmitter can use that wavelength without causing a collision. In this design however, a problem arises because of the difficulty in optically monitoring the power on several tens of WDM wavelengths. One option is to tap a small percentage of power from the ring within the node and to send that WDM stream to a WDM channel monitor, which is composed of a scanning optical filter and a detector. However, because IP packets on the optical ring can be as short as 50 ns, the filter would have to scan the entire WDM transmission bandwidth at a rate of greater than 20 MHz. This is difficult to achieve today. A second option is to send the tapped WDM stream to a WDM demultiplexer, which has a photodetector at each of the outputs of the demultiplexer. This option is far more expensive than desired, however, because of the high cost of WDM demultiplexers and the large number of photodetectors and receiver circuits that are required for a network with a high number of wavelengths. Examples of this approach can be found in [13,18].

The first design of the MAC protocol for *HORNET*, which is called *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) [19], accomplishes the same result as the above scheme but with much less equipment cost. In the CSMA/CA protocol, each network wavelength is assigned a corresponding unique RF frequency that has a higher value than the baud rate of the payload data stream. For example, if the payload data rate is 10 Gbps, the lowest possible RF frequency must be significantly greater than 10 GHz (e.g. 15 GHz). When a node transmits a packet, it frequency-multiplexes a subcarrier tone, where the subcarrier uses the frequency that corresponds to the wavelength carrying the packet. A node determines what wavelengths are occupied with packets in the WDM traffic passing through the node by tap-

ping a small amount of optical power and receiving it with a photodetector. The resulting instantaneous power spectrum contains power at the subcarrier frequencies corresponding to the wavelengths carrying packets at the moment. An experimental demonstration is reported in [19]. Clearly, by only using one photodetector and by using RF demultiplexing instead of optical demultiplexing, costs can be significantly reduced as compared to the alternative methods of wavelength monitoring presented above.

Despite the apparent advantages of the CSMA/CA scheme, it was ultimately determined that the scheme was not the best. The main concern is the fact that the subcarrier frequencies lie well beyond the payload data baud rate. This is necessary for proper demultiplexing of the subcarrier tones and the payload data in both the subcarrier receiver and the payload data receiver. Thus, if the data rate is 10 Gbps, the subcarrier tones may be required to be higher than 15 GHz. Because of the difficulty of building narrow-band filters at such high frequencies, and because of the large number of subcarrier frequencies used in a high capacity network, the band for the subcarriers may stretch over several GHz. As a result, for a bit rate of 10 Gbps, the network nodes would likely be forced to use transmitters and receivers with a total bandwidth of 20 to 25 GHz, significantly increasing the cost of the network.

4.2 The HORNET MAC Protocol

Possible replacements for the CSMA/CA protocol were investigated, some of which are described in [4]. Ultimately, the current approach for the HORNET MAC protocol evolved from these proposed designs. HORNET uses a control channel to convey the *wavelength availability information*. The control channel is carried on its own wavelength in the WDM network. That control wavelength is dropped and added in every node so that all nodes can process and modify the control channel. The implementation of the control channel is inexpensive if a wavelength of approximately 1310 nm is used to transport the control channel.

Figure 5 illustrates the operation of the control channel for the MAC protocol. The control channel is time-slotted into frames, much like any typical point-to-point high-speed data stream. The frame boundaries are demarcated with a *start-of-frame* (SOF) indicator byte. Within each frame is a bit-stream that conveys the wavelength availability information for the time period during the following frame. This allows the node to see one frame into the future. Potentially, the design could be modified to allow more look-ahead if it is determined to be beneficial.

The wavelength availability bit-stream is a sequence of bits of length W , where

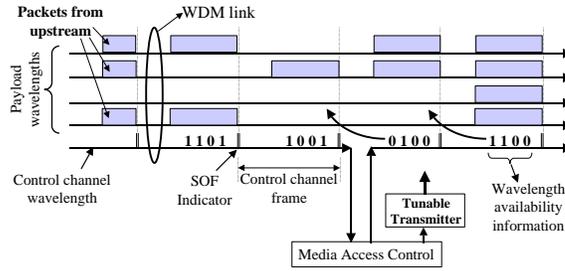


Fig. 5. The control channel conveys the availability of the wavelengths during a framed time period.

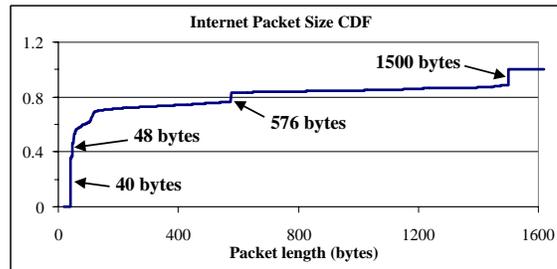


Fig. 6. This cumulative distribution function of IP packet sizes on a particular link measured by NLANR shows that packets range from 40 bytes to 1500 bytes.

W is the number of wavelengths in the network. If bit w equals a '1,' then wavelength w is carrying a packet during the time period of the next control channel frame. A '0' bit indicates that the wavelength is available during the next frame. A node sorts its queued packets into virtually separated queues called virtual output queues (VOQs) [20], the classic technique to avoid the head-of-line (HOL) blocking problem [21]. Each VOQ corresponds to a wavelength in the network. When a node reads the bit stream, it determines the set of VOQs with a packet to transmit that overlaps with the set of available wavelengths. The node then determines which packet in the overlapping set it will transmit during the next frame. If the node decides to send a packet on wavelength w , it modifies bit w in the wavelength availability bit-stream to a '1.' All nodes clear the wavelength availability bit(s) corresponding to the wavelength(s) that they receive.

The framed format of the control channel makes the MAC protocol ideal for small, fixed-sized packets. However, Internetworking Protocol (IP) packets are inherently variable in size. Figure 6 shows a cumulative distribution function (CDF) of packet sizes measured on a typical IP link. This data is measured and reported by the National Laboratory for Applied Network Research (NLANR) [22]. As shown in the figure, IP packets have a very wide range of typical sizes, from 40 bytes to 1500 bytes.

Such a wide range of packet sizes is not compatible with a framed control channel with inflexible frame sizes. A simple solution exists for this problem

that avoids any changes to the MAC protocol. As is done in IP-over-ATM, the variable-sized IP packets can be segmented into small, fixed-sized cells. The size of the segmented cell and the size of the control channel frame can be designed to match each other. Although the solution is simple, there is a significant drawback to the segmentation. Whenever a packet or a segment of a packet is transmitted, a header must be applied. The *HORNET* header includes information about the source and destination, allowance for transmitter tuning time and clock recovery time, and a few other items as well. Thus, a long packet, such as a 1500-byte packet, will have the *HORNET* packet header applied to it a large number of times. This will result in an excessive amount of overhead.

Adding only a small amount of intelligence into the MAC protocol can significantly reduce the overhead. Instead of automatically segmenting the packets such that each packet fits in one frame, the *HORNET* MAC protocol segments packets only when necessary. This modification to the MAC protocol is called *segmentation and re-assembly on demand* (SAR-OD). In this protocol, a node must begin to insert a packet in alignment with the beginning of the control frame. If the packet is longer than the control frame duration, the node *continues to transmit* the packet (without segmenting the packet and re-applying the header) until either the packet is complete or until the MAC protocol informs the transmitter that another packet is coming from upstream on the transmission wavelength. If an upstream packet on the node's current transmission wavelength passes through the node while the node is transmitting a packet, the node *ceases the transmission* of its packet at the end of the last available frame (i.e. the one before the frame that is carrying the oncoming packet). At the end of the packet segment, the transmitter applies a byte that indicates that the segment is an incomplete packet. The node is now free to send packets on different wavelengths while it waits for an opportunity to finish the packet it had begun. At the next opportunity, the node begins transmitting the segmented packet beginning with the location in the packet at which it was segmented. When the final segment of a packet is completely transmitted, the node finishes the packet with a byte that indicates that the packet is complete.

The receiver in a *HORNET* node has a slight amount of extra intelligence built into it to work with the SAR-OD protocol. The receiving process is illustrated in Figure 7. The receiver in a node maintains separate virtual queues for each node on the ring. When a packet arrives at the receiver, the receiver reads the packet header to determine the source node and then begins to write the payload of the arriving packet into the virtual queue corresponding to the source node. If the last byte of the segment indicates that the packet is incomplete, the segment remains in the queue. The next segment arriving at the receiver from the same source node will belong to the same packet, and thus the receiver will store this segment at the queue location immediately following the

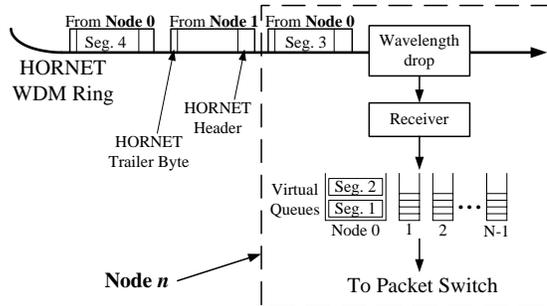


Fig. 7. After receiving the packet segments, the node queues them in separate queues sorted according to the source node. After the entire packet is received, it is passed onto the packet switch.

previously received segment, just like a first-come-first-serve (FCFS) queue. When the packet is fully received, it will be sent to the node's packet switch with the integrity of the IP packet completely preserved.

In the example shown in Figure 7, Node 0 is sending a long packet to Node n . Two of the segments already arrived to Node n and are stored in the queue waiting for the rest of the packet. After beginning the third segment, a packet from Node 1 to Node n passed through Node 0, forcing it to segment the packet again. After the packet from Node 1 has passed, Node 0 can begin the fourth segment of the packet for Node n . When the third and fourth segments arrive to Node n they will be stacked in the queue on top of the first two segments. If the fourth segment is the last, the final byte will indicate so, and Node n will pass the re-assembled packet on to the packet switch.

4.3 Fairness Control for the HORNET MAC Protocol

4.3.1 Unfairness of the HORNET Architecture

Although there are many advantages to using the bi-directional ring architecture for *HORNET*, there is one problem that arises because of it. Multiple-access ring networks are inherently unfair. The unfairness problem is most easily seen by considering only one of the network wavelengths and then unwrapping it, as is done in Figure 8. Consider the wavelength that is received by Node $N-1$ in Figure 8. When Node 0 wants to send packets to Node $N-1$, it is never blocked on the wavelength received by Node $N-1$. When Node 1 wants to send packets to Node $N-1$, it has to contend with (can occasionally be blocked by) the packets transmitted by Node 0 on the wavelength of Node $N-1$. Node 2 has to contend with Nodes 0 and 1, while Node 3 has to contend with Nodes 0, 1, and 2. This pattern continues around the ring to Node $N-2$, which has to contend with all of the nodes except Node $N-1$, making it more difficult for Node $N-2$ to transmit packets to Node $N-1$ than for the

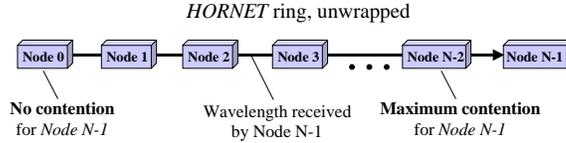


Fig. 8. The *HORNET* ring unwrapped, while focusing on the wavelength received by Node $N-1$. Nodes closer to Node $N-1$ experience more difficulty sending packets to Node $N-1$ than do the nodes further upstream.

nodes further upstream. Thus, the network is *biased against* nodes closer to the destination. As a result, the VOQs that are queuing packets for *unfortunate* source-destination pairs will experience lower throughput, resulting in *higher latency* for packets in the VOQs. Clearly, fairness control is necessary for the *HORNET* MAC to avoid this negative result.

Before designing a fairness control protocol, it is imperative to determine the goal of the protocol. Often when defining fairness, the network nodes are treated as users, and the fairness scheme is designed to give all nodes the same amount of bandwidth. In such a control scheme, if there are ten nodes on the network and they all want a percentage of a wavelength's bandwidth arbitrarily greater than 10%, then the network would allocate 10% of the available bandwidth for each node. Similarly, if three nodes are attempting to access the same wavelength, and one node wants 99% of the wavelength's bandwidth while the other two want 10% each, then the first node receives 80% and the other two receive their 10%. In this case, only one node suffers for the over-subscription of the wavelength. The other two get exactly what they desire.

However, nodes are not end users. In this work, it is argued that basing the fairness control on the principle of allocating bandwidth to *nodes* does not eliminate positional priority. Imagine the following hypothetical example. A Web server attached to a node in the network hosts an Internet contest. Contestants are required to make a Web connection with the server that requires a significant amount of bandwidth. If many more users in one geographic area of the ring are intelligent enough to participate than in another area, then the contestants in the intelligent region are penalized. Consider the case where the wavelength received by the node hosting the contest's Web server can only support enough bandwidth for 1000 connections. Attached to one node in the network are 999 contestants who desire to participate, while two other nodes are hosting only 100 contestants each. If bandwidth is allocated to nodes as explained above, the node with 999 contestants will only be allowed to utilize the bandwidth to support 800 users, while the other nodes serve all 100 of their users. As a result, the users on the popular node are penalized because of their location (i.e. because of their close proximity to other users).

In this work, fairness is considered on the basis of the *end user*, not the *node*.

The fairness control protocol designed for this work attempts to transform the ring into one *large distributed FCFS queue*. If a wavelength becomes oversubscribed, as in the previous example, then all nodes will suffer the same average packet latency. Thus, a *user's* position on the ring becomes irrelevant. There is no disadvantage to being located closer to the destination, and there is no disadvantage to living in an area densely populated with similar users.

4.3.2 *HORNET* Fairness Control Protocol: DQBR

The solution for the fairness control protocol developed in this work is a novel protocol established specifically for incorporation into the *HORNET* MAC protocol. It is called *Distributed Queue Bi-directional Ring* (DQBR) because the protocol attempts to transform *HORNET's* bi-directional ring architecture into a distributed FCFS queue. The protocol is an adaptation of an older protocol called *Distributed Queue Dual Bus* (DQDB) [23–26], which was created for single channel dual-bus metro networks of the 1980's. It is also known as IEEE 802.6.

In *IEEE 802.6*, when a packet arrives to the front of a transmitter's queue, the node sends a request in the direction opposite to which the packet must be transmitted (upstream). The request consists of setting the *request bit* in the control information field of the current frame. The request passes through all nodes that are upstream of the requesting node with respect to the direction that the packet will travel. The nodes count the requests they see. According to the protocol, a node must allow enough *unused* frames to pass through to satisfy all the requests it has seen *before* a packet came to the front of its queue. To an approximation, this causes the network to emulate a distributed FCFS queue. For example, if packets arrive at Nodes 2 and 3 before a packet arrives at Node 1 (where Node 1 is further upstream), Node 1 must allow two empty frames to pass by before sending its packet so that Nodes 2 and 3 can send their packets first.

DQBR, the *HORNET* fairness control protocol, is adapted from IEEE 802.6 to accommodate *HORNET's* WDM ring by allowing one request *for each wavelength* in each control channel frame, and by maintaining request counters *for each wavelength*. It is implemented using the *HORNET* control channel. The control channel frame carries two bit streams, each of length W , where W is the number of wavelengths. The first bit stream indicates wavelength availability information (as explained in Section 4.2) and the second indicates *requests* (note that if $2W$ bits are used for the requests, four levels of priority can be requested, but that extension is not covered in this work). When a node receives a packet in its transmitter's VOQ, it sets the bit in the *request bit stream* corresponding to the wavelength the packet will use for transmission. All nodes clear the request bit(s) from the control channel corresponding to

the wavelength(s) that they receive.

The original version of IEEE 802.6 contains a well-known unfairness problem that is thoroughly described in [25,26]. When an upstream node is saturating the transmission bandwidth of the multiple-access channel, a downstream node can be nearly locked out of the network. This issue occurs because when the downstream node places a request, there is significant propagation time for the request to get to the upstream node and then for the available slot to reach the downstream node. During the time between the two events, the upstream node can fill all available slots with packets. Only after the downstream node transmits its packet can it send another request, because the request is sent when a packet reaches the *front* of the transmission queue. The result is that the upstream node is allowed to use almost the entirety of the channel bandwidth.

A correction was developed for this unfairness problem named *bandwidth balancing* [25,26]. Bandwidth balancing allocates transmission bandwidth evenly among the nodes. However, this solution is contrary to the definition of fairness presented in Section 4.3.1. Therefore, a different solution was developed for *DQBR*. With DQBR fairness control, the node places the upstream requests as soon as a packet arrives to the *back* of the transmitter's queue. The result is that upstream nodes are made aware of the downstream nodes' need for bandwidth and as a result they allow the nodes the opportunity to transmit.

The request counting system, which is diagrammed in Figure 9, works as follows. A node maintains a *request counter* (RC) for each wavelength. Every time the node sees a request bit on the control channel for any particular wavelength, it increments the RC for the corresponding wavelength, as shown in Figure 9 (a) and (b). Whenever the transmitter's VOQ in a node receives a packet that desires to use a particular wavelength, the value in the RC for that wavelength is transferred to a *wait counter* (WC), which is *stamped* onto the arriving packet, as shown in Figure 9 (c). The RC is then cleared. After the packet makes its way to the front of the VOQ, the node decrements its WC value for each available frame it sees on the desired wavelength, as illustrated by Figure 9 (c) and (d) ('available frame' refers to a '0' bit in the downstream control channel wavelength availability bit stream). Only when the WC value has been decremented to zero can the packet be transmitted. If the WC value equals zero, or if there is not a packet in the front of the queue, the RC value is decremented each time an available time frame passes by on the corresponding wavelength.

The DQBR request-counting system attempts to ensure that if two packets arrive at two different nodes and desire the same wavelength, the one that arrived first will be transmitted first, as if the network is one large distributed FCFS queue. According to the definition of fairness presented in Section 4.3.1,

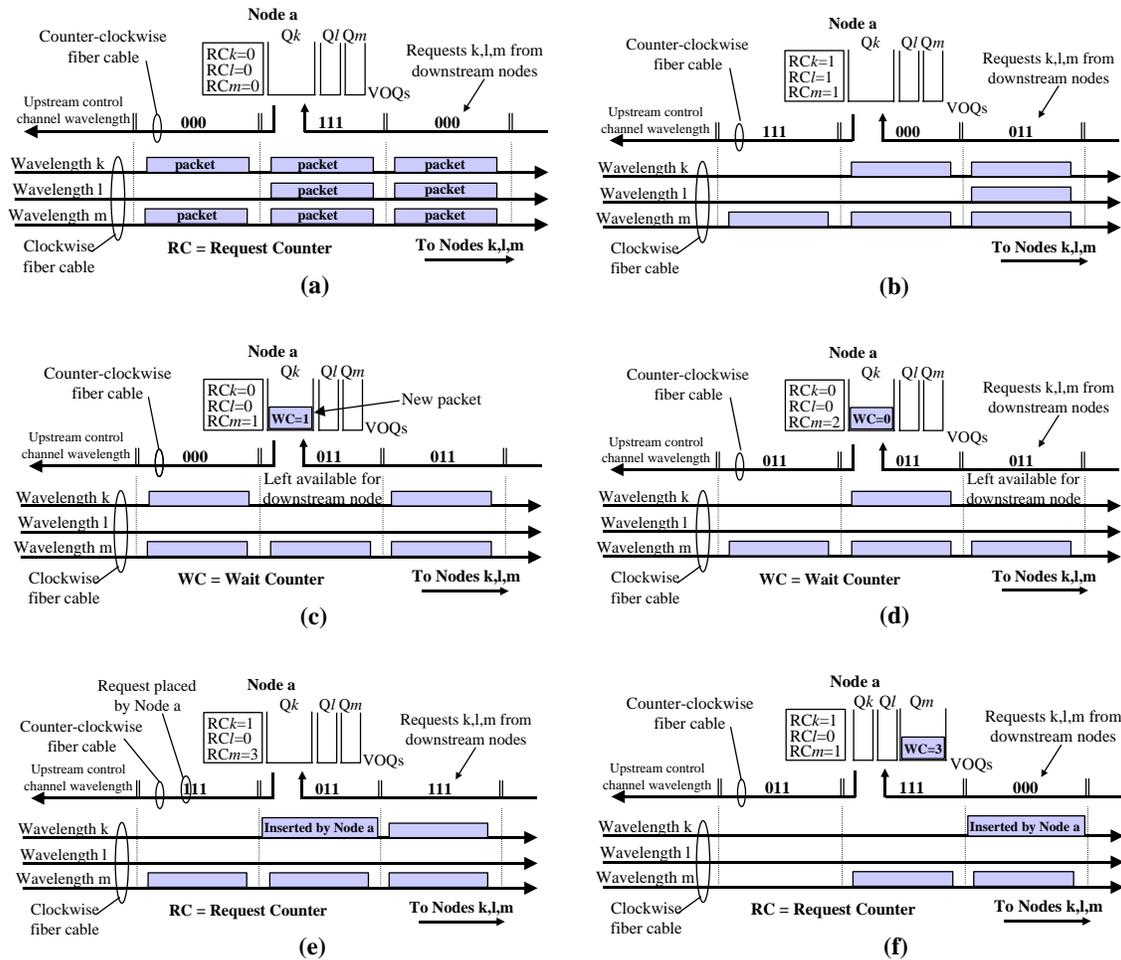


Fig. 9. DQBR operation: (a) A node monitors the requests on the upstream control channel coming from the downstream nodes. (b) The node increments the RC counters for any requests it sees. (c) When a packet arrives in a VOQ, the value in the corresponding RC counter is stamped onto the packet as the WC. The packet cannot be inserted into the availability because the WC value is nonzero. (d) The WC counter is decremented for every availability that passes by on the corresponding wavelength. (e) The packet can now be transmitted. (f) When a packet arrives to VOQ m , the value from RC_m is moved into the WC stamped onto the packet. The packet will have to allow three empty frames on Wavelength m to pass before it can be transmitted.

the distributed FCFS operation is in fact fair to all users in the network. The ability of the protocol to guarantee equal opportunity for all users of any location is investigated later in Section 5.3.

4.4 *HORNET* MAC Summary

The simple and inexpensive control-channel-based MAC protocol designed for *HORNET* coordinates transmissions within the nodes to avoid collisions. The protocol is designed to accommodate variable-sized packets using *segmentation and re-assembly on demand*. The DQBR fairness control component of the MAC protocol provides equal opportunity for all users on the network. In Section 5, simulations are used to measure the performance of the MAC, with an emphasis on its ability to provide fairness control.

5 Network Simulations for the *HORNET* MAC Protocol

A custom designed simulator was developed to model the *HORNET* lower-layer protocols. As presented in this section, the simulator is used to analyze the performance of the SAR-OD protocol and the performance penalty due to the *HORNET* overhead. Also, the simulator is used to verify that DQBR provides fairness control and to measure any penalty resulting from DQBR.

5.1 *HORNET* Simulator

The simulation iterates over time steps while iterating over all nodes during each time step. The time duration of a time step iteration is equal to the time duration of a control channel frame. While operating on a node, the simulator performs statistical arrivals at each VOQ in the node. The packets arriving at the queues in a particular node are the statistical sum of packets being generated by hundreds of users that are accessing the Internet through that node.

Variable-sized packets are used in the simulations. The distribution of IP packet sizes is discussed in Section 4.2. Estimates based on all of the distributions found in [22] are used for packet size distributions throughout the rest of this work. The variable-sized packet arrivals are modelled in these simulations as follows. After a packet arrives, the simulator determines its length. The simulation uses a probability density function (PDF) specified by the simulation user to randomly determine the packet size. Figure 10 shows the corresponding cumulative distribution (CDF) of packet sizes. The distribution shown in this figure is used in the simulations presented in this work.

In each time step of the simulation each node attempts to transmit payload data and to place requests for the DQBR fairness control. The simulator main-

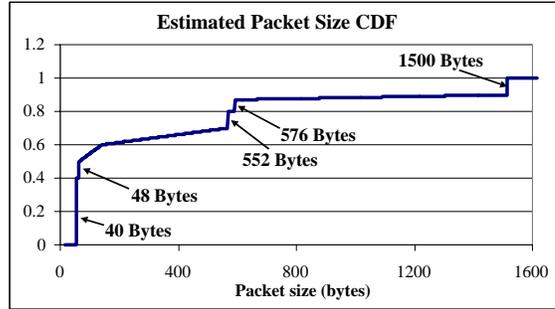


Fig. 10. A cumulative distribution function of packet sizes modelled by the simulator.

tains control channel frames for each direction of transmission. The control channel frames carry the wavelength availability information and the DQBR requests. In each time step the control channel frames rotate one slot around the ring in the appropriate direction. The simulator also maintains the RC and WC counters for each node, as specified by DQBR.

After a packet is completely transmitted, the simulator records the age of the packet. At the end of the simulation, the average latency of packets transmitted from each VOQ is calculated. The maximum capacity of the network can be determined by locating the network load at which the average packet latency asymptotically approaches infinity.

To make the simulation realistic, extra overhead must be added to the packet. The overhead of a *HORNET* packet includes guard band for synchronization errors and tuning time, source and destination node numbers, a cyclic redundancy check sequence, and a few other items. It is expected that the header in a commercial implementation of *HORNET* will be 16 bytes in duration. Thus, 16 bytes of overhead are used in the simulations in this work.

In the *HORNET* simulation, whenever the node begins transmitting a packet, it must first insert the overhead. Thus, if the frame length is 64 bytes and the header is 16 bytes, during the first frame the node can only send (and thus subtract) 48 bytes from the packet that is currently being transmitted. If the packet continues into the next frame, then the simulator can subtract 64 bytes from the remaining length, assuming that there are at least 64 bytes remaining in the packet. If the packet is segmented, then the node must insert header bytes again when it resumes transmission of the packet. Thus, segmentation of packets adds extra overhead that detracts from the performance of the network. The simulator tracks the number of overhead bytes transmitted as one of its important statistics.

Figure 11 analyzes the performance of *HORNET* when overhead and packet segmentation and re-assembly are included in the simulation. The graph illustrates the performance penalty due to the overhead in a *HORNET* network. The curve on the far right is the result of simulations with small fixed-sized

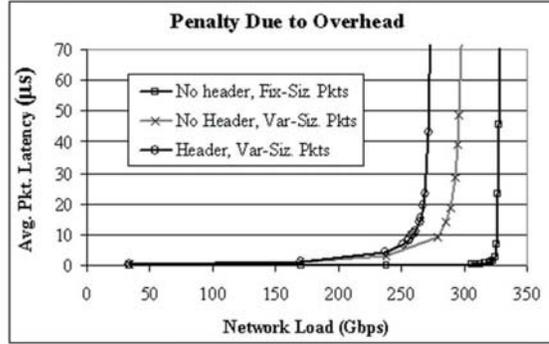


Fig. 11. This graph shows the penalty incurred for the use of variable-sized packets and 16-byte packet headers. The transmission rate is 10 Gbps.

packets that match the control channel frame size and with no headers applied to the packets. For this case, the maximum capacity for a 17-node network is shown to be 340 Gbps (assuming a transmission rate of 10 Gbps). This is an intuitive result. For each of the two directions, there are 17 transmitters sending packets on 17 wavelengths that can support 10 Gbps data. Under uniform traffic conditions, each of the 34 transmitters sends a maximum of 10 Gbps, resulting in a maximum capacity of 340 Gbps.

The curve to the left of it uses variable-sized packets but still no packet headers. The average latency increases because longer packets take longer to transmit and because there is now a small amount of overhead due to packet size mismatch with the control frame size. This happens because variable-sized IP packets will in general not have a duration that is equal to an integer number of control channel frames. Thus, a packet will end at a random location along the control channel frame. The next packet on that wavelength cannot begin until the next control channel frame begins, and thus there is a period of time on the wavelength that payload data cannot be carried. This unusable bandwidth is overhead, and slightly reduces the performance of the network. The control channel size is optimized in these simulations to keep the penalty to significantly less than 10%.

The third curve in Figure 11 is the simulation result when 16-byte packet headers are applied to all packets in the network. As the figure shows, a penalty is incurred due to the additional overhead of the packet headers. The SAR-OD protocol is used in both of the simulations that considered variable-sized packets. The optimized control channel frame length of 64 bytes is used in this simulation, as well as all other simulations in this work with variable-sized packets and packet headers.

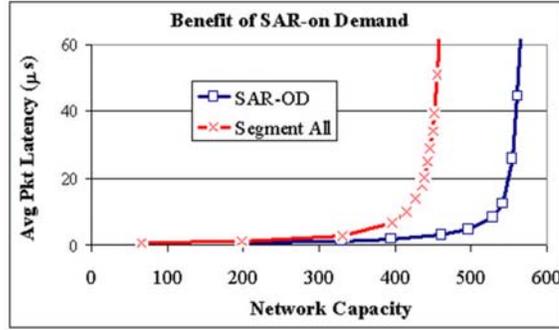


Fig. 12. This graph shows the advantage of using SAR-OD instead of automatically segmenting all packets into small, fixed-sized cells. The transmission rate is 10 Gbps.

5.2 Performance Advantage of SAR-OD

The SAR-OD protocol was developed for *HORNET* to avoid the excessive overhead that can result from segmenting variable-sized packets into fixed-sized cells to fit the transmission frame (e.g. IP-over-ATM). However, SAR-OD adds slightly more complexity to the node design than does the alternative. Thus, it is important to measure the performance benefit provided by SAR-OD to determine whether the extra complexity results in a meaningful performance advantage. The performance advantage measured by the simulator is shown in Figure 12. The graph shows a performance advantage of approximately 15%. Intuitively, this makes sense. The overhead measured by the simulator at the maximum load for the curve shown in shown in Figure 12 is 10.5%. The average overhead for a network that segments all packets can easily be calculated to be more than 25% (16 bytes of overhead in every 64-byte slot, plus unused bytes at the end of the packet). As a result, a performance advantage of at least 15% is expected.

5.3 DQBR Performance Simulations

In Section 4.3.1, the unfairness of the *HORNET* architecture was described. It was suggested that lower throughput occurs for VOQs buffering packets for unfortunate source-destination pairs. The simulator verifies that without fairness control, this result in fact occurs. Figures 13 and 14 show the throughput in several VOQs in a *HORNET* network when DQBR is *not used*. Figure 13 shows VOQ number 18 in each node on a 25-node bi-directional *HORNET* ring when Wavelength 18 is heavily saturated. VOQ 18 in each node is queuing packets that are destined for Node 18, which in this simulation receives Wavelength 18. As the figure shows, nodes closer to their destination are unable to transmit to the destination node when the transmission wavelength is saturated.

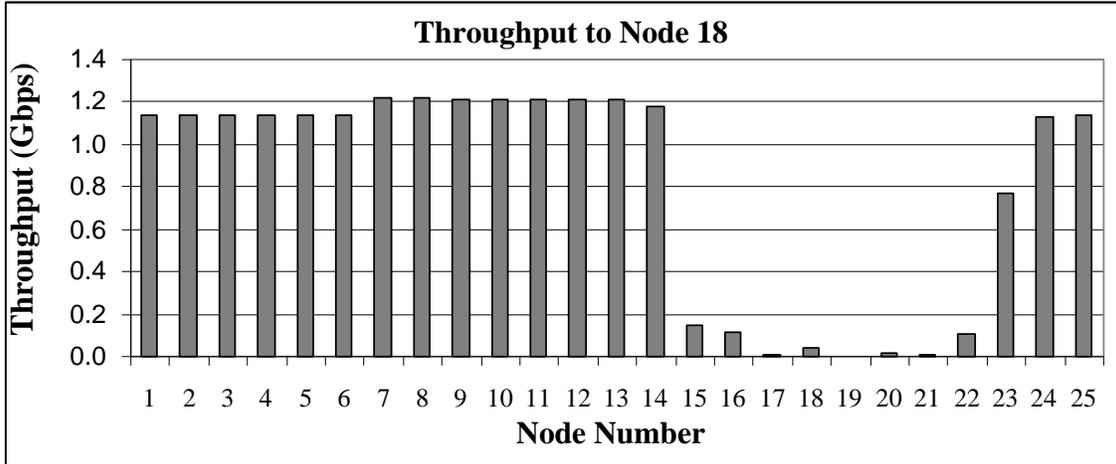


Fig. 13. Throughput in the nodes' VOQs that use Wavelength 18 for all nodes on the network. The nodes transmitting on the wavelength that their neighbor receives have very high latency because of the unfairness problem.

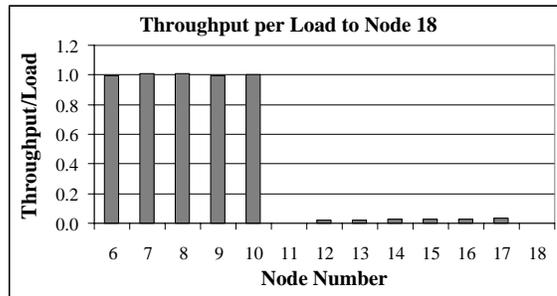


Fig. 14. Throughput divided by load on the nodes' VOQs that use Wavelength 18. Nodes 10 and 11 are sending a large amount of traffic to Node 18, while the other nodes are only sending light amounts of traffic.

The simulation that generated the results shown in Figure 14 models a network traffic scenario that is very likely to cause unfairness problems in a *HORNET* network. For this plot, Nodes 10 and 11 are sending very heavy amounts of traffic to Node 18, while all other nodes are sending a very light amount. The traffic from Nodes 10 and 11 is saturating the wavelength received by Node 18 (Wavelength 18). The Figure shows each node's *throughput* divided by the *load on VOQ 18*. According to the definition of fairness presented in Section 4.3.1, all nodes would have the same ratio of *throughput to load* if the network were fair. However, because of the unfairness of the architecture, the nodes between Nodes 10 and 18 are unable to use Wavelength 18 to send packets to Node 18. Clearly, the simulations verify that there is a need for a fairness control protocol in *HORNET*.

5.3.1 DQBR Measured Fairness Performance

To demonstrate the fairness control, the throughput of each node is measured when the network is saturated. To do this, the traffic conditions of the simulation are such that the total network load on the observed wavelength is significantly greater than the capacity of the wavelength. Because the wavelength is oversubscribed, a queue management protocol is necessary because otherwise the queue depths will grow uncontrollable. To ensure that the simulations are realistic under such conditions, the *Random Early Detection* (RED) protocol for congestion control [27] is implemented in the simulator because it is expected that a similar protocol would be used in a commercial *HORNET* network. In reality, it is preferred that the congestion control protocol presented in [28] is used because it penalizes users that do not properly respond to the congestion protocol. It is assumed in this work that all users will behave properly, and thus the RED protocol is used.

Figures 15 and 16 show that DQBR resolves the unfairness problem in the *HORNET* architecture. Initially, fixed-sized packets are used because DQBR was explained under that assumption. Variable-sized packets are addressed in Section 5.3.3. Figure 15 shows the throughput for nodes sending packets to Node 18 on a 25-Node *HORNET* network. With DQBR, the throughput is equal for all nodes, whereas *without* DQBR, the nodes close to Node 18 have a very difficult time sending packets to Node 18. Also, recall from Section 4.3.2 that DQBR is designed to eliminate the unfairness condition that occurs in IEEE 802.6 due to propagation distance between nodes [26]. In this simulation, there is enough propagation distance between nodes to hold 50 control frames, yet the throughput is still equal for all nodes when DQBR is used. Thus, it is clear that propagation distance does not affect the fairness of DQBR.

Figure 16 shows a simulation with the same unbalanced traffic as in Figure 14. In this traffic case, Node 10 has 9.33 Gbps of traffic arriving to its queue destined for Node 18, Node 11 has 4.67 Gbps destined for Node 18, and all other nodes have very little traffic. The wavelength can only support 10 Gbps, so it is heavily oversubscribed. As the figure shows, without DQBR controlling the fairness, the nodes close to Node 18 are unable to transmit packets on Wavelength 18, while in the DQBR network, all nodes have an equal ratio of *throughput to load* for Wavelength 18.

To justify the fairness of this situation, imagine that the simulation results of Figure 16 were generated by the following network conditions. There are 250 users of a *HORNET* network. All are sending 58.3 Mbps of traffic to Node 18. Attached to Node 10 are 160 of those users, while 80 are accessing the network through Node 11. The other ten users are each using one of the other nodes shown in the plot of Figure 16. Under a scheme that equalizes bandwidth to the nodes, such as DQDB's bandwidth balancing [25,26], the users attached

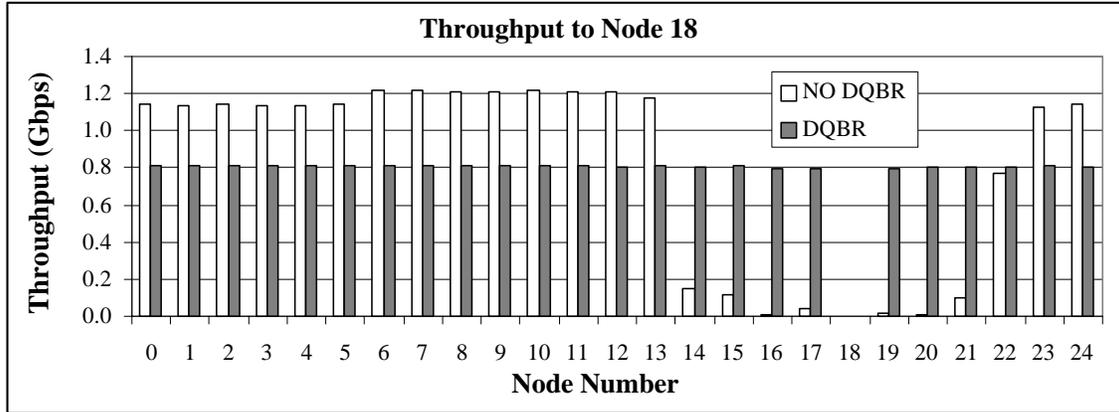


Fig. 15. Throughput for VOQ number 18 for the 25 nodes on a *HORNET* network. VOQ number 18 corresponds to Wavelength 18, which is received by Node 18. The total network load for Wavelength 18 is 1.5 times its capacity. There is enough propagation delay between nodes to hold 50 control frames.

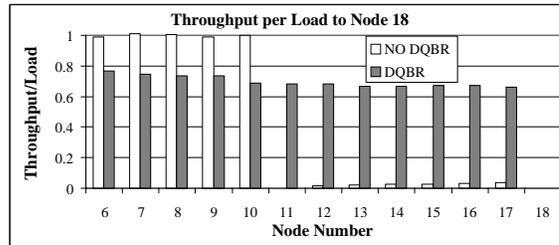


Fig. 16. *Throughput divided by load* for VOQ number 18 for several nodes. VOQ number 18 corresponds to Wavelength 18, which is received by Node 18. In this simulation, the load on VOQ 18 in Node 10 is 9.33 Gbps, and the load on VOQ 18 in Node 11 is 4.67 Gbps. All other nodes have only a small load.

to Node 10 would be required to reduce their throughput to 29.7 Mbps each, while all other uses continue to transmit at 58.3 Mbps. This is because Node 10 would be allocated 4.753 Gbps, allowing Node 11 to transmit at 4.664 Gbps, and all other nodes to transmit at 58.3 Mbps. This might be fair if nodes were users, but instead the users of Node 10 are penalized because they happen to be grouped in the same location. In contrast, DQBR allocates each node a *throughput to load ratio* of approximately 0.7, as shown in Figure 16, and thus each user receives 40 Mbps.

To verify this result further, the average packet latency and the packet drop probability can be analyzed. The average delay suffered by packets in the VOQs for Node 18 is plotted in Figure 17. The results are generated using the same unbalanced traffic case described above. As the figure shows, with DQBR packets suffer the same latency in all nodes. The packet drop probability at each node is shown in Figure 18 for the unbalanced traffic case. The packet loss probability in a *HORNET* network at all nodes is nearly equal when DQBR is used. Thus, all users of the network will experience the same packet loss

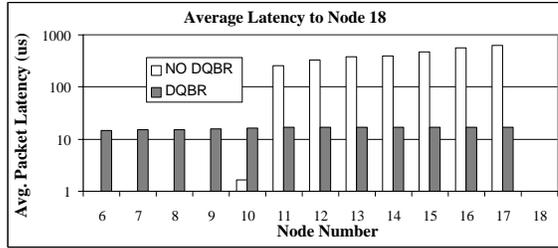


Fig. 17. Average packet latency in each *HORNET* node for the unbalanced traffic case (Nodes 10 and 11 have a heavy traffic load for Node 18 while all others have light traffic).

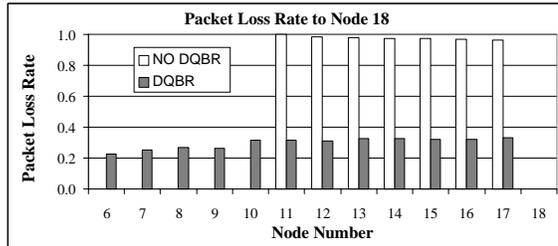


Fig. 18. Packet loss probability in each *HORNET* node for the unbalanced traffic case (Nodes 10 and 11 have a heavy traffic load for Node 18 while all others have light traffic).

probability, and as a result the transport control protocol will regulate the users' load in the same way.

5.3.2 DQBR Performance Penalty

It should be obvious by inspection that the *HORNET* network without fairness control is work-conserving (i.e. if an input has at least one packet for at least one currently available output, then the input will transmit a packet with a probability of 1). If a node has a packet to transmit and there is an opening for the packet, the only event that would prevent the node from sending that packet is if the node sends another packet. Thus, 100% throughput is achievable (when overhead is not considered). However, when DQBR control is applied to the *HORNET* network, it is no longer perfectly work conserving. This is because the DQBR fairness control occasionally forces nodes *not* to transmit any packets, even though there are packets in the queues and available wavelengths to carry those packets. The reason a node does this is because it may be forced to allow an availability on a wavelength to go by for downstream nodes to use. In most cases, these wavelength availabilities would be utilized by nodes downstream. However, there is a non-zero probability that the downstream node that generated the corresponding request may decide to transmit a packet on a different wavelength, and thus leave the wavelength availability unused. The simulator computed the total throughput

for the simulations presented in Figure 15. Without DQBR, the throughput is 0.999, while with DQBR the throughput is 0.965. Thus, the penalty of DQBR is only 3.5%. This is a very minor penalty, considering the tremendous benefit it provides.

5.3.3 DQBR with Variable-Sized Packets

Thus far in this section, the simulations analyzing the performance of *HORNET* with DQBR fairness control have only used fixed-sized packets that are the same size as the time step (and thus the control channel frame), and that have no overhead. In that case, when a packet arrives, the node attempts to insert one request into the upstream control channel. However, the situation is more complicated when variable-sized packets are transmitted using the SAR-OD protocol. When a packet arrives to the node, the node should place a number of control channel requests equal to the packet's length measured in control channel frames. If the packet is not going to be segmented, then the calculation is as simple as dividing the sum of the packet length and header by the control channel frame size (in bytes). However, in the segmentation and re-assembly protocol, the node must reapply the header each time a packet is segmented, making the total number of bytes transmitted a random variable because the number of packet segmentations is random. The random variable depends on the traffic with which the node must contest, and thus is different for each wavelength as well as time variant.

The ideal solution is for the node to correctly estimate how many times a packet will be segmented to determine the amount of bytes that will be transmitted (payload plus overhead), and to place the necessary amount of requests to carry this amount of bytes. For example, if a node must segment a 560-byte packet five times (i.e. into 6 segments), and the header is 16 bytes, then it will send a total of $560 + (6 \times 16) = 656$ bytes. If the frame size is 64 bytes, then the node should place $\lceil \frac{656}{64} \rceil = 11$ upstream requests, where $\lceil \dots \rceil$ is the *ceiling* operator. If the node had not considered the extra overhead due to segmentation and re-assembly, it would have only placed $\lceil \frac{560+16}{64} \rceil = 9$ upstream requests.

Ultimately, however, determining the correct expected value for the number of times a packet will be segmented is very complex. It depends not only on the upstream traffic rate, but also on the burstiness and self-similarity of the traffic. In practice, this may be very difficult to measure. In this work, it is assumed that the packet segmentation probability is solely dependent upon the upstream traffic rate, which can easily be measured in practice by monitoring the control channel. To determine the number of slots to request

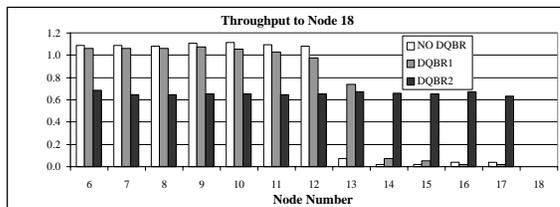


Fig. 19. Throughput for VOQ number 18 for several nodes for the following cases: no fairness control; DQBR without considering SAR-OD (DQBR1); and DQBR while considering overhead due to SAR-OD (DQBR2). VOQ number 18 corresponds to Wavelength 18, which is received by Node 18.

for a packet, the node uses the following expression:

$$Rq = \lceil \frac{PB}{(CCF - R_u \times HB)} \rceil.$$

where Rq is the number of requests, PB is the number of payload bytes transmitted, HB is the number of bytes in the *HORNET* header, CCF is the control channel frame length, R_u is the upstream traffic rate (normalized to 1), and $\lceil \dots \rceil$ is the *ceiling* operator. Thus, the number of requests varies linearly with the traffic rate between the minimum and maximum number of possible requests. The result of using this expression is shown in Figure 19. This figure shows the throughput in the VOQs on Wavelength 18 when no fairness control is used, when DQBR *without* considering segmentation is used (DQBR1 in legend), and when DQBR while considering segmentation is used (DQBR2 in legend). As the figure shows, DQBR is not perfectly effective unless overhead due to packet headers is properly considered. In fact it only has a small effect on fairness control when the extra overhead due to SAR-OD is not considered.

6 Summary and Conclusions

HORNET reduces the infrastructure cost of a high-capacity next-generation metro network by utilizing fast-tunable packet transmitters and wavelength routing in its architecture. A novel control-channel-based MAC protocol was developed for the architecture. The protocol has a simple, distributed design with an inexpensive implementation. It is designed to transport variable-sized IP packets and to provide fairness control to the network users through the use of the novel DQBR protocol.

A custom-designed simulator was assembled to analyze the performance of *HORNET* and its protocols. In this work the simulator was used to quantify the benefit of the SAR-OD protocol and to analyze the performance of the

DQBR fairness control protocol. The simulation results showed that the SAR-OD protocol has better than a 15% advantage over the conventional method of handling variable-sized packets. Also, the simulator proved that DQBR provides fairness control in the *HORNET* architecture. Fair operation was verified for a network with sufficient propagation distance between nodes and for the case of variable-sized IP packets. Additionally, the simulator measured a performance penalty of only 3.5% due to the implementation of DQBR, which is excellent considering the benefit DQBR provides.

From the results summarized above, we can conclude that the MAC protocol developed for the *HORNET* architecture is practical and cost-effective. The SAR-OD protocol enables *HORNET* to be efficient, even when dealing with the difficult task of transporting variable-sized packets in a multiple-access environment. The DQBR fairness control mechanism transforms the inherently unfair *HORNET* architecture into a fair architecture without sacrificing performance or dramatically increasing the complexity. Thus, the MAC protocol presented in this work is efficient, fair, cost-effective, and practical. It is a critical *enabling technology* for *HORNET*, and thus the completion and verification of the MAC protocol brings the deployment of a commercial *HORNET* network closer to a reality.

7 Acknowledgements

The authors would like to thank Eric Hu, Yu-Li Hsueh, and Hiroshi Okagawa for their helpful contributions on this publication.

References

- [1] I. M. White. *A New Architecture and Technologies for High-Capacity Next Generation Metropolitan Networks*. PhD dissertation, Stanford University, Department of Electrical Engineering, August 2002.
- [2] I. M. White, K. Shrikhande, M. S. Rogge, S. M. Gemelos, D. Wonglumsom, G. Desa, Y. Fukashiro, and L. G. Kazovsky. Architecture and protocols for HORNET: A novel packet-over-WDM multiple-access MAN. In *GLOBECOM 2000 Conference Record*, pages 1298–1302, November 2000.
- [3] K. Shrikhande, A. Srivatsa, I. M. White, M. S. Rogge, D. Wonglumsom, S. M. Gemelos, and L. G. Kazovsky. CSMA/CA MAC protocols for IP-HORNET: An IP over WDM metropolitan area ring network. In *GLOBECOM 2000 Conference Record*, pages 1303–1307, November 2000.

- [4] K. Shrikhande, I. M. White, D. Wonglumsom, S. M. Gemelos, M. S. Rogge, Y. Fukashiro, M. Avenarius, and L. G. Kazovsky. HORNET: A packet-over-WDM multiple access metropolitan area ring network. *IEEE Journal on Selected Areas in Communications*, 18(10):2004–2016, October 2000.
- [5] I. M. White, M. S. Rogge, Y-L. Hsueh, K. Shrikhande, and L. G. Kazovsky. Experimental demonstration of the HORNET survivable bi-directional ring architecture. In *Optical Fiber Communications Technical Digest*, pages 346–349, March 2002.
- [6] K. Shrikhande, I. M. White, M. S. Rogge, F-T. An, E. S. Hu, S. S-H. Yam, and L. G. Kazovsky. Performance demonstration of a fast-tunable transmitter and burst-mode packet receiver for HORNET. In *Optical Fiber Communications Technical Digest*, pages ThG2:1–ThG2:3, March 2001.
- [7] G. Barish and K. Obraczka. World Wide Web caching: Trends and techniques. *IEEE Communications Magazine*, pages 178–185, May 2000.
- [8] T. T. Tay, Y. Feng, and M. N. Wijesundera. A distributed Internet caching system. In *Proceedings of the 25th Annual IEEE Conference on Local Computer Networks*, pages 624–633, November 2000.
- [9] Resilient Packet Ring Alliance. An introduction to resilient packet ring technology. White Paper, available at <http://www.rpralliance.org>, October 2001.
- [10] N. M. Froberg, S. R. Henion, H. G. Rao, B. K. Hazzard, S. Parikh, B. R. Romkey, and M. Kuznetsov. The NGI ONRAMP test bed: Reconfigurable WDM technology for next generation regional access networks. *Journal of Lightwave Technology*, 18(12):1697–1708, December 2000.
- [11] M. J. Spencer and M. A. Summerfield. WRAP: A medium access control protocol for wavelength-routed passive optical networks. *Journal of Lightwave Technology*, 18(12):1657–1676, December 2000.
- [12] M. A. Marsan, A. Bianco, E. Leonardi, M. Meo, and F. Neri. MAC protocols and fairness control in WDM multirings with tunable transmitters and fixed receivers. *Journal of Lightwave Technology*, 14(6):1230–1244, June 1996.
- [13] R. Gaudino, A. Carena, V. Ferrero, A. Pozzi, V. De Feo, P. Gigante, F. Neri, and P. Poggiolini. RINGO: A WDM ring optical packet network demonstrator. In *Proceedings of the 27th European Conference on Optical Communications*, September 2001.
- [14] A. Smiljanic and B. Loehfelm. Performance evaluation of optical ring network based on composite packet switching. In *Optical Fiber Communications Technical Digest*, pages 286–287, March 2002.
- [15] J-P. Faure, L. Noirie, A. Bisson, V. Sabouret, G. Leveau, M. Vigoureux, and E. Dotaro. A scalable transparent waveband-based optical metropolitan network. In *Proceedings of the 27th European Conference on Optical Communications*, September 2001.

- [16] N. LeSauze, A. Dupas, E. Dotaro, L. Ciavaglia, M. H. M. Nizam, A. Ge, and L. Dembeck. A novel, low cost optical packet metropolitan ring architecture. In *Proceedings of the 27th European Conference on Optical Communications*, September 2001.
- [17] T. Anderson, S. Owicki, J. Saxe, and C. Thacker. High speed switch scheduling for local area networks. *ACM Transactions on Computer Systems*, 11(4):319–352, November 1993.
- [18] E. Wong, S. K. Marks, M. A. Summerfield, and R. D. T. Lauder. Baseband optical carrier-sense multiple access-demonstration and sensitivity measurements. In *Optical Fiber Communications Technical Digest*, pages WU2:1–WU2:3, March 2001.
- [19] I. M. White, M. S. Rogge, K. Shrikhande, Y. Fukashiro, D. Wonglumsom, F-T. An, and L. G. Kazovsky. Experimental demonstration of a novel media access protocol for HORNET: A packet-over-WDM multiple-access MAN ring. *IEEE Photonics Technology Letters*, 12(9):1264–1266, September 2000.
- [20] D. Wonglumsom, I. M. White, S. M. Gemelos, K. Shrikhande, and L. G. Kazovsky. HORNET - a packet-switched WDM metropolitan area ring network: Optical packet transmission and recovery, queue depth, and packet latency. In *1999 IEEE LEOS Annual Meeting Conference Proceedings*, pages 653–654, November 1999.
- [21] Y. Tamir and G. Frazier. High performance multi-queue buffers for VLSI communication switches. In *Proceedings of the 15th Annual Symposium on Computer Architecture*, pages 343–354, June 1988.
- [22] National Laboratory for Applied Network Research, Measurement Operations and Analysis Team. <http://pma.nlanr.net/Datacube/>.
- [23] IEEE Standard 802.6. Distributed queue dual bus (DQDB) subnetwork of a metropolitan area network (MAN), December 1990.
- [24] R. M. Newman, Z. L. Budrikis, and J. L. Hullett. The QPSX MAN. *IEEE Communications Magazine*, 26(4):20–28, April 1988.
- [25] E. L. Hahne, A. K. Choudhury, and N. F. Maxemchuk. Improving the fairness of Distributed-Queue-Dual-Bus networks. In *Infocom '90*, pages 175–184, 1990.
- [26] E. L. Hahne, A. K. Choudhury, and N. F. Maxemchuk. DQDB networks with and without bandwidth balancing. *IEEE Transactions on Communications*, 40(7):1192–1204, July 1992.
- [27] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.
- [28] R. Pan, B. Prabhakar, and K. Psounis. CHOKe: A stateless active queue management scheme for approximating fair bandwidth allocation. In *Infocom 2000*, pages 942–951, 2000.

Summary of the HORNET Project

Ian M. White (*student member, IEEE*), Kapil Shrikhande,
Matthew S. Rogge (*student member, IEEE*), and Leonid G. Kazovsky *Fellow, IEEE*

Abstract

Metropolitan area networks are currently undergoing an evolution aimed at more efficiently transporting data-oriented traffic. However, the incoming generation of metro networks is based on conventional technology, which prevents them from cost-effectively scaling to ultra-high capacities. We have developed a new architecture and set of protocols for the next generation of metro networks. The architecture, named HORNET, is a packet-over-WDM ring network that utilizes fast-tunable packet transmitters and wavelength routing to enable it to scale cost-effectively to ultra-high capacities. A control-channel-based MAC protocol enables the network nodes to share the bandwidth of the network while preventing collisions. The MAC protocol is designed to transport variable-sized packets and to provide fairness control to all network end users. The efficiency and the fairness of the MAC protocol is proven with custom-designed simulations. The implementation of the MAC protocol and the survivability of the network have been demonstrated in a laboratory experimental testbed.

This article summarizes the accomplishments of the HORNET project, including the design, analysis, and demonstration of a data-optimized metro architecture and a set of protocols. As this work shows, the HORNET architecture and protocols are an excellent candidate for next-generation high-capacity metro networks.

I. INTRODUCTION

OPTICAL communications networks in the metropolitan area must evolve to adapt to the new environment of data-dominated traffic. One popular approach currently under development is to adapt Ethernet to a ring topology, as is suggested by the IEEE Resilient Packet Ring (RPR) Working Group (IEEE 802.17) [?]. Another somewhat related approach is to use a new framing protocol named Generic Framing Procedure (GFP) [?] along with virtual concatenation [?] for SONET. Both of these approaches enable more efficient transport of data-oriented traffic. Both approaches are sensible for today's network traffic. However, Internet traffic in the metro area (particularly data-oriented traffic) will continue to grow to the point at which next-generation metro networks will need to support capacities on the order of 1 Tb/s. At such high capacities, architectures based on conventional technologies require excessive amounts of transmitters, receivers, and packet switching capacity. Because the metro area is a very cost-sensitive market, high capital expenses for equipment and high operating expenses for space, power, and complexity should be avoided. Thus, a new architecture must be developed that cost-effectively scales to ultra-high capacities.

We have proposed a new architecture named *HORNET* [2–4] (Hybrid Opto-electronic Ring Network), which satisfies all of the requirements of next generation metro networks. The architecture is designed to be survivable to a fiber cut or node failure. A novel control-channel-based media access control (MAC) protocol was developed for *HORNET*. The MAC protocol is designed to efficiently transport variable sized packets, and to provide *fair* access to the network for all users.

This article summarizes the accomplishments of the *HORNET* project. The architecture for *HORNET* is described in Section II. In Section III, the survivability of the architecture is presented. Section IV describes the *HORNET* MAC protocol, and Section VI details the novel design of the *HORNET* node. Then, an experimental demonstration of the survivable architecture is presented in Section VII. A system-level mathematical analysis is then described in Section VIII. Finally, the work is summarized in Section IX.

II. HORNET ARCHITECTURE

The *HORNET* architecture is designed to cost-effectively scale beyond 1 Tb/s while efficiently transporting bursty, packet-based, randomly fluctuating traffic. The architectural concept is shown in Figure 1. *HORNET* is a bi-directional ring topology designed to leverage the currently deployed fiber infrastructure. Also, the 2-fiber bi-directional ring architecture enables *HORNET* to be survivable.

As Figure 1 shows, nodes use fast-tunable packet transmitters to insert packets onto the ring. The packets are coupled optically onto the ring using a wideband coupler (currently, a fast-tunable wavelength-selective multiplexer is not available). A packet is transmitted on the wavelength that is received by the packet's destination node. A wavelength drop is used to drop one or more assigned wavelengths into each node. Thus, only the packets destined for a particular node are dropped into the node. All of the packets carried by the other wavelengths pass through optically, such that the node does not receive or process them. Conventional architectures require significantly more equipment in the nodes because the nodes must receive, process, and re-transmit all packets that pass through. In *HORNET*, a node only needs enough equipment to process the packets to and from its local users.

Funded by The Defense Advanced Research Projects Agency under agreement number F30602-00-2-0544, and by Sprint Advanced Technology Labs.

The authors are with the Optical Communications Research Laboratory at Stanford University, 350 Serra Mall, Stanford, CA 94305 (e-mail: ian-white@stanfordalumni.org).

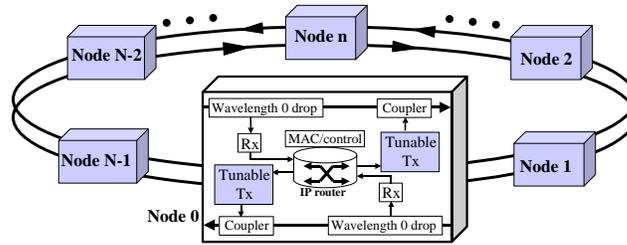


Fig. 1. The *HORNET* architecture is a bi-directional wavelength routing ring network with tunable transmitters in each node.

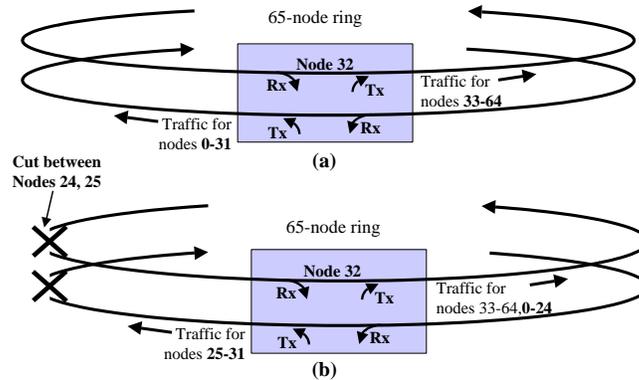


Fig. 2. (a) Under normal operating conditions, a node attempts to load-balance its traffic while using all available bandwidth in both directions. (b) When Node 32 learns of the cut, it determines that to reach Nodes 0 through 24 it must use the counter-clockwise ring.

Other projects [5–8] have also in recent years investigated next-generation optical ring architectures for the metro area. Some of these projects use the same wavelength routing concepts that are used in the *HORNET* architecture. However, the survivability scheme, the MAC protocol, and the node design developed for *HORNET* are unique. All are presented in the following three sections.

III. HORNET SURVIVABILITY

Because Internet access is *critical* to businesses and consumers, a metro network architecture must be survivable to a fiber cut or a node failure. Today's SONET networks typically use a 2-fiber unidirectional path switched ring (2FUPSR) or a 4-fiber bi-directional line switched ring (4FBLSR) architecture, both of which are described in [9]. Both of these architectures are survivable, but there is a drawback to their employment. Only one half of the equipment and available bandwidth is used for network traffic. The other half is reserved for the rare occurrence of a failure event. This straight-forward approach is necessary because of SONET's rigid TDM architecture.

HORNET does not have rigid circuits that must be reprovisioned in order to change paths between sources and destinations. This advantage over SONET is the basis for the *HORNET* survivability mechanism. The *HORNET* survivable architecture is a 2-fiber bi-directional path-switched ring (2FBPSR). In the *HORNET* 2FBPSR network, *all of a node's transmission capacity* is used for working traffic. None is reserved for protection. In the *HORNET* bi-directional architecture, two paths exist between any two nodes. Under normal conditions, when an access node has a packet to send, it chooses the transmitter that will send the packet along the better of the two paths, as determined by a simple routing algorithm. When a cut occurs, only one of the paths remains to each destination, and thus the node is forced to use that path. The path switch occurs logically inside the node's control and routing electronics. This ensures fast, reliable path switching in the event of a cut. The concept is illustrated in Figure 2.

A control channel is used for the detection of the cut and for broadcasting the necessary information about the cut (the primary use of the control channel is described in Section IV). When a cut occurs between two nodes, both of those nodes realize that they are no longer receiving optical power on the control channel or in the payload receiver(s). After the two nodes determine that a cut occurred, they each insert a message into the first control channel frame possible. The message passes through all other nodes on the ring. When a node reads the message, it determines the location of the cut based on the node address in the message. The node then uses what it knows about the topology of the network to determine for which nodes it now must use a different path, as illustrated in Figure 2.

In selecting a different path for some of the destinations, the node is performing a *switch logically*, similar to the physical switching that occurs in the SONET architectures. The logical switch takes place within the forwarding engine of the packet router within the *HORNET* node. The packet router's forwarding engine inspects the IP address of the packet and determines

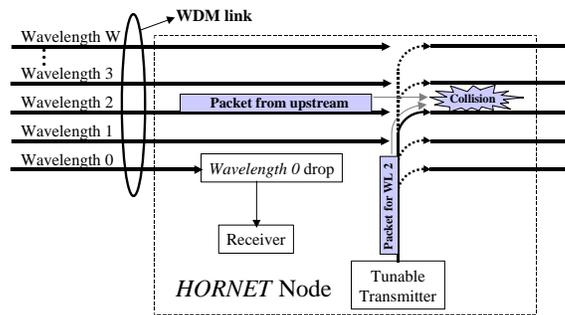


Fig. 3. A collision occurs when a transmitter inserts a packet on a wavelength that is currently carrying a packet through the node.

the destination node, similar to any typical IP router. Then, a second stage of the forwarding engine determines which path (clockwise or counter-clockwise) the packet should take to get to the destination. A table is maintained in the forwarding engine to indicate which direction packets should use for certain destination nodes. This table is updated based on network conditions, load balancing, and of course, fiber cuts. When a node learns of a cut, it determines which entries in the table must be toggled (i.e. which paths must change), and then toggles the values in the table. The amount of entries will be small, because the length of the table is equal to the number of nodes on the network. Also, the determination of which entries to modify is a simple operation, especially if the nodes are numbered sensibly.

When a cut occurs in a conventional SONET network, the transmission capacity of each node is unaffected because all links are fully protected. In *HORNET's* architecture, the effect the cut has on transmission capacity of a particular node is location-dependent. For nodes far away from the cut, the transmission capacity is generally unaffected. However, nodes closer to the cut are more affected. The extreme case is the node adjacent to the cut, which, as a result of the cut, only has the use of one of the two fiber rings for all of its transmitted data. This in general reduces its available capacity by one half, bringing it down to the same capacity as a node in a conventional network (neglecting *HORNET's* inherent advantage of being able to dynamically adapt to traffic variations). This implies that the *HORNET* architecture can guarantee to its users the maximum capacity of a conventional network, while providing up to 100% more transmission capacity for best-effort traffic, which of course is the most common traffic on the Internet today.

IV. *HORNET* MEDIA ACCESS CONTROL PROTOCOL

Since the packet ADM process in the *HORNET* architecture is completely different from the ADM process of any preceding commercial network, a new MAC protocol must be developed. The primary function of the MAC protocol in *HORNET* is to prevent collisions at the point in the node where the transmitter inserts packets. Since the transmitter can insert a packet on any wavelength, and since most of the wavelengths are passing through the node without being terminated, a transmitter could insert a packet onto a particular wavelength that collides with another packet that is passing through the node on that wavelength. Figure 3 shows the occurrence of a collision. To prevent collisions, the MAC protocol should monitor the WDM traffic passing through the node, locate the wavelengths that are available, and inform the transmitter of which wavelengths it is allowed to use at a particular moment. As a result, the transmitter will not insert a packet on a wavelength that is currently carrying another packet through the node.

A. *HORNET* MAC Design

The first design of the MAC protocol for *HORNET* is a scheme called *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) [10]. In the CSMA/CA protocol, each network wavelength is assigned a corresponding unique RF frequency that has a higher value than the baud rate of the payload data stream. For example, if the payload data rate is 10 Gbps, the lowest possible RF frequency must be significantly greater than 10 GHz (e.g. 15 GHz). When a node transmits a packet, it frequency-multiplexes a subcarrier tone, where the subcarrier uses the frequency that corresponds to the wavelength carrying the packet. A node determines what wavelengths are occupied with packets in the WDM traffic passing through the node by tapping a small amount of optical power and receiving it with a photodetector. The resulting instantaneous power spectrum contains power at the subcarrier frequencies corresponding to the wavelengths carrying packets at the moment. An experimental demonstration is reported in [10]. Clearly, by using only one photodetector and by using RF demultiplexing instead of optical demultiplexing, costs can be significantly reduced as compared to the alternative methods of wavelength monitoring presented above.

Despite the apparent advantages of the CSMA/CA scheme, it was ultimately determined that the scheme was not the best. The main concern is the fact that the subcarrier frequencies lie well beyond the payload data baud rate. This is necessary for proper demultiplexing of the subcarrier tones and the payload data in both the subcarrier receiver and the payload data receiver. Thus, if the data rate is 10 Gbps, the subcarrier tones may be required to be higher than 15 GHz. Because of the difficulty of building narrow-band filters at such high frequencies, and because of the large number of subcarrier frequencies used in a high capacity

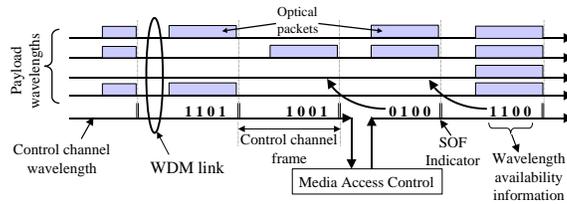


Fig. 4. The control channel conveys the availability of the wavelengths during a framed time period.

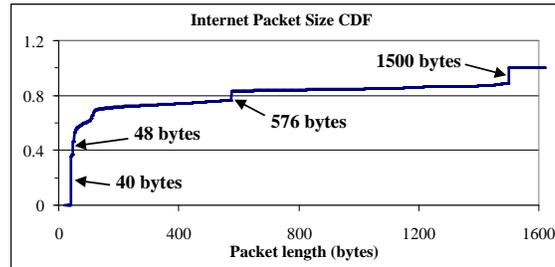


Fig. 5. This cumulative distribution function of IP packet sizes on a particular link measured by NLANR shows that packets range from 40 bytes to 1500 bytes.

network, the band for the subcarriers may stretch over several GHz. As a result, for a bit rate of 10 Gbps, the network nodes would likely be forced to use transmitters and receivers with a total bandwidth of 20 to 25 GHz, significantly increasing the cost of the network.

Possible replacements for the CSMA/CA protocol were investigated, some of which are described in [?]. Ultimately, the current approach for the *HORNET* MAC protocol evolved from these proposed designs. *HORNET* uses a control channel to convey the *wavelength availability information*. The control channel is carried on its own wavelength in the WDM network. That control wavelength is dropped and added in every node so that all nodes can process and modify the control channel. The implementation of the control channel is inexpensive if a wavelength of approximately 1310 nm is used to transport the control channel.

Figure 4 illustrates the operation of the control channel for the MAC protocol. The control channel is time-slotted into frames, much like any typical point-to-point high-speed data stream. The frame boundaries are demarcated with a *start-of-frame* (SOF) indicator byte. Within each frame is a bit-stream that conveys the *wavelength availability information* for the time period during the following frame. This allows the node to see one frame into the future. Potentially, the design could be modified to allow more look-ahead if it is determined to be beneficial.

The wavelength availability bit-stream is a sequence of bits of length W , where W is the number of wavelengths in the network. If bit w equals a '1,' then wavelength w is carrying a packet during the time period of the next control channel frame. A '0' bit indicates that the wavelength is available during the next frame. A node sorts its queued packets into virtually separated queues called virtual output queues (VOQs) [?], the classic technique to avoid the head-of-line (HOL) blocking problem [11]. Each VOQ corresponds to a wavelength in the network. When a node reads the bit stream, it determines the set of VOQs with a packet to transmit that overlaps with the set of available wavelengths. The node then determines which packet in the overlapping set it will transmit during the next frame. If the node decides to send a packet on wavelength w , it modifies bit w in the wavelength availability bit-stream to a '1.' All nodes clear the wavelength availability bit(s) corresponding to the wavelength(s) that they receive.

B. Variable-Sized Packets in *HORNET*

The framed format of the control channel makes the MAC protocol ideal for small, fixed-sized packets. However, Internet-working Protocol (IP) packets are inherently variable in size. Figure 5 shows a cumulative distribution function (CDF) of packet sizes measured on a typical IP link. This data is measured and reported by the National Laboratory for Applied Network Research (NLANR) [?]. As shown in the figure, IP packets have a very wide range of typical sizes, from 40 bytes to 1500 bytes.

Such a wide range of packet sizes is not compatible with a framed control channel with inflexible frame sizes. A simple solution exists for this problem that avoids any changes to the MAC protocol. As is done in IP-over-ATM, the variable-sized IP packets can be segmented into small, fixed-sized cells. The size of the segmented cell and the size of the control channel frame can be designed to match each other. Although the solution is simple, there is a significant drawback to the segmentation. Whenever a packet or a segment of a packet is transmitted, a header must be applied. The *HORNET* header includes information about the source and destination nodes, allowance for transmitter tuning time and clock recovery time, and a few other items as well. Thus, a long packet, such as a 1500-byte packet, will have the *HORNET* packet header applied to it a large number of times. This results in an excessive amount of overhead.

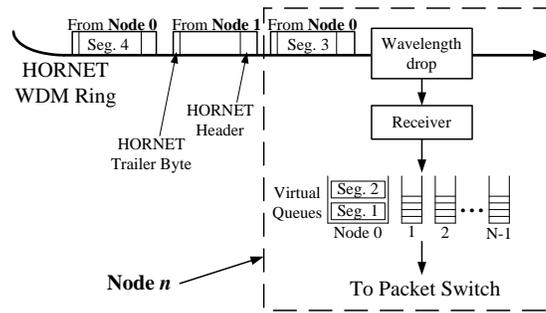


Fig. 6. After receiving the packet segments, the node queues them in separate queues sorted according to the source node. After the entire packet is received, it is passed onto the packet switch.

Adding only a small amount of intelligence into the MAC protocol can significantly reduce the overhead. Instead of automatically segmenting the packets such that each packet fits in one frame, the *HORNET* MAC protocol segments packets only when necessary. This modification to the MAC protocol is called *segmentation and re-assembly on demand* (SAR-OD). In this protocol, a node must begin to insert a packet in alignment with the beginning of the control frame. If the packet is longer than the control frame duration, the node *continues to transmit* the packet (without segmenting the packet and re-applying the header) until either the packet is complete or until the MAC protocol informs the transmitter that another packet is coming from upstream on the transmission wavelength. If an upstream packet on the node's current transmission wavelength passes through the node while the node is transmitting a packet, the node *ceases the transmission* of its packet at the end of the last available frame (i.e. the one before the frame that is carrying the oncoming packet). At the end of the packet segment, the transmitter applies a byte that indicates that the segment is an incomplete packet. The node is now free to send packets on different wavelengths while it waits for an opportunity to finish the packet it had begun. At the next opportunity, the node begins transmitting the segmented packet beginning with the location in the packet at which it was segmented. When the final segment of a packet is completely transmitted, the node finishes the packet with a byte that indicates that the packet is complete.

The receiver in a *HORNET* node has a slight amount of extra intelligence built into it to work with the SAR-OD protocol. The receiving process is illustrated in Figure 6. The receiver in a node maintains separate virtual queues for each node on the ring. When a packet arrives at the receiver, the receiver reads the packet header to determine the source node and then begins to write the payload of the arriving packet into the virtual queue corresponding to the source node. If the last byte of the segment indicates that the packet is *incomplete*, the segment remains in the queue. The next segment arriving at the receiver from the same source node will belong to the same packet, and thus the receiver will store this segment at the queue location immediately following the previously received segment, just like an FCFS queue. When the packet is fully received, it will be sent to the node's packet switch with the integrity of the IP packet completely preserved.

In the example shown in Figure 6, Node 1 is sending a long packet to Node n . Two of the segments already arrived to Node n and are stored in the queue waiting for the rest of the packet. After beginning the third segment, a packet from Node 0 to Node n passed through Node 1, forcing it to segment the packet again. After the packet from Node 0 has passed, Node 1 can begin the fourth segment of the packet for Node n . When the third and fourth segments arrive to Node n they will be stacked in the queue on top of the first two segments. If the fourth segment is the last, the final byte will indicate so, and Node n will pass the re-assembled packet on to the packet switch.

It is important to measure the performance benefit provided by SAR-OD to determine whether the extra complexity results in a meaningful performance advantage. For this and other similar purposes, a simulator was constructed to model the networking aspects of *HORNET* [2]. The performance advantage measured by the simulator is shown in Figure 7. The simulator uses the variable-sized packet cumulative distribution function shown in Figure 8. The graph shows a performance advantage of approximately 15%. Intuitively, this makes sense. The overhead measured by the simulator at the maximum load for the curve shown in Figure 7 is 10.5%. The average overhead for a network that segments all packets can easily be calculated to be more than 25% (16 bytes of overhead in every 64-byte slot, plus unused bytes at the end of the packet). As a result, a performance advantage of at least 15% is expected.

C. Fairness Control for the *HORNET* MAC Protocol

1) *Unfairness of the HORNET Architecture:* Although there are many advantages to using the bi-directional ring architecture for *HORNET*, there is a problem that arises because of it. Multiple-access ring networks are inherently unfair. The unfairness problem is most easily seen by considering only one of the network wavelengths and then unwrapping it, as is done in Figure 9. Consider the wavelength that is received by Node $N-1$ in Figure 9. When Node 0 wants to send packets to Node $N-1$, it is never blocked on the wavelength received by Node $N-1$. When Node 1 wants to send packets to Node $N-1$, it has to contend with (can occasionally be blocked by) the packets transmitted by Node 0 on the wavelength of Node $N-1$. Node 2 has to contend with Nodes 0 and 1, while Node 3 has to contend with Nodes 0, 1, and 2. This pattern continues around the ring to Node $N-2$, which

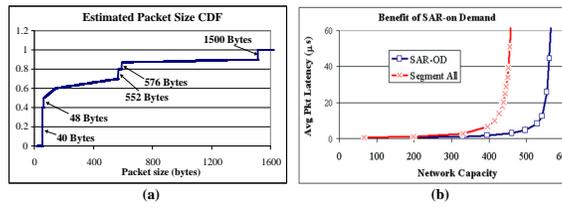


Fig. 7. This graph shows the advantage of using SAR-OD instead of automatically segmenting all packets into small, fixed-sized cells. The transmission rate is 10 Gbps.

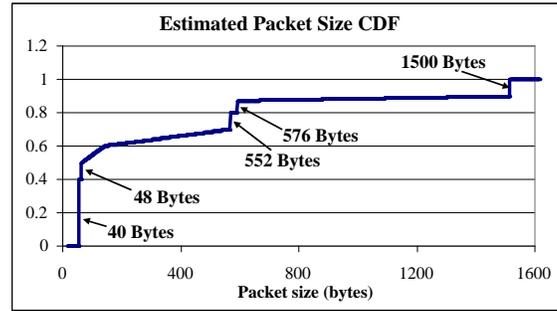


Fig. 8. The cumulative distribution function of packet sizes modelled by the HORNET simulator.

has to contend with all of the nodes except Node $N-1$, making it more difficult for Node $N-2$ to transmit packets to Node $N-1$ than for the nodes further upstream. Thus, the network is *biased against* nodes closer to the destination. As a result, the VOQs that are queuing packets for *unfortunate* source-destination pairs will experience lower throughput, resulting in *higher latency* for packets in the VOQs. Clearly, fairness control is necessary for the *HORNET* MAC to avoid this negative result.

In this work, fairness is considered on the basis of the *end user*, not the *node*, as explained in detail in [2]. The fairness control protocol designed for this work attempts to transform the ring into one *large distributed FCFS queue*. If a wavelength becomes oversubscribed, all nodes will suffer the same average packet latency, regardless of whether the some nodes are offering more traffic than others. Thus, a *user's* position on the ring becomes irrelevant. There is no disadvantage to being located closer to the destination, and there is no disadvantage to living in an area densely populated with similar users. This is in stark contrast to network architectures that attempt to allocate all nodes equal bandwidth, penalizing *nodes* for attempting to transmit too much traffic onto the ring.

2) *HORNET Fairness Control Protocol: DQBR*: The solution for the fairness control protocol developed in this work is a novel protocol established specifically for incorporation into the *HORNET* MAC protocol. It is called *Distributed Queue Bi-directional Ring* (DQBR) because the protocol attempts to transform *HORNET's* bi-directional ring architecture into a distributed FCFS queue. The protocol is an adaptation of an older protocol called *Distributed Queue Dual Bus* (DQDB) [?, ?, ?, ?], which was created for single channel dual-bus metro networks of the 1980's. It is also known as IEEE 802.6.

An example of the DQBR fairness control protocol is shown in Figure 10. In each control channel frame, a bit stream of length W called the *request bit stream* follows the *wavelength availability information*, where W is the number of wavelengths. When a node on the network receives a packet into VOQ w , the node notifies the *upstream* nodes about the packet by setting bit w in the *request bit stream* in the control channel that travels upstream with respect to the direction the packet will travel. For the case of variable-sized packets, the node places the number of requests corresponding to the length of the packet measured in frames [2].

All upstream nodes take note of the requests by incrementing a counter called a *request counter* (RC). Each node maintains an RC corresponding to each wavelength. Thus, if bit w in the *request bit stream* is set, RC_w is incremented (Figure 10 (a) and (b)). Each time a packet arrives to VOQ w , the node stamps the value in RC_w onto the packet and then clears the RC (Figure 10 (c)). The stamp is called a *wait counter* (WC). After the packet reaches the front of the VOQ, if the WC equals n it must allow n frame availabilities to pass by for downstream packets that were generated earlier (Figure 10 (c) and (d)). When an availability passes

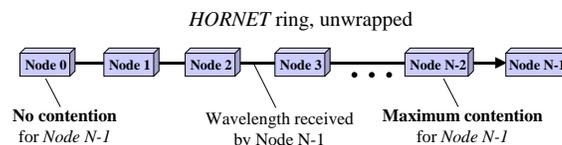


Fig. 9. The *HORNET* ring unwrapped, while focusing on the wavelength received by Node $N-1$. Nodes closer to Node $N-1$ experience more difficulty sending packets to Node $N-1$ than do the nodes further upstream.

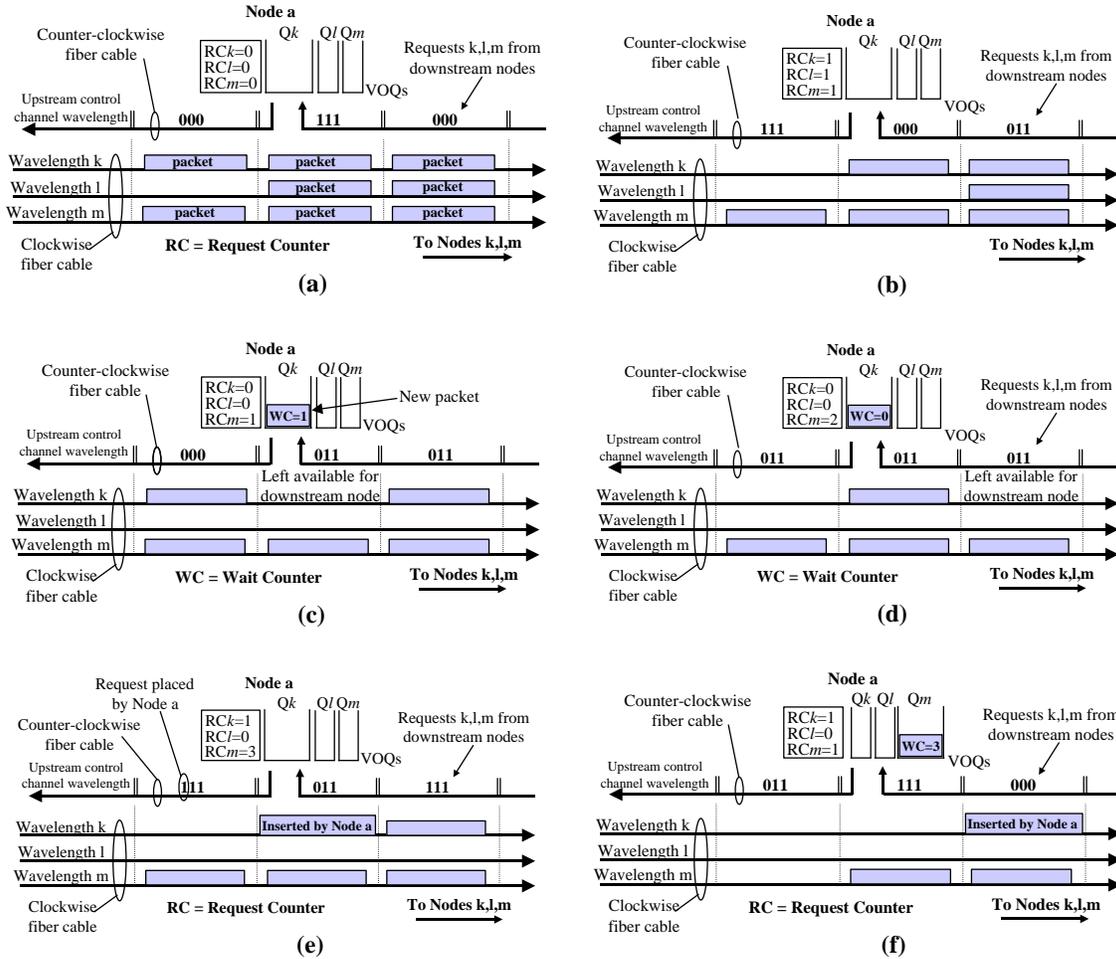


Fig. 10. DQBR operation: (a) A node monitors the requests on the upstream control channel coming from the downstream nodes. (b) The node increments the RC counters for any requests it sees. (c) When a packet arrives in a VOQ, the value in the corresponding RC counter is stamped onto the packet as the WC. The packet cannot be inserted into the availability because the WC value is nonzero. (d) The WC counter is decremented for every availability that passes by on the corresponding wavelength. (e) The packet can now be transmitted. (f) When a packet arrives to VOQ m , the value from RC_m is moved into the WC stamped onto the packet. The packet will have to allow three empty frames on Wavelength m to pass before it can be transmitted.

by the node on wavelength w , the WC for the packet in the front of VOQ_w is decremented (if the WC equals zero, then RC_w is decremented). Not until the WC equals zero can the packet can be transmitted (Figure 10 (e)). The counting system ensures that the packets are transmitted in the order that they arrived to the network.

The DQBR request-counting system attempts to ensure that if two packets arrive at two different nodes and desire the same wavelength, the one that arrived first will be transmitted first, as if the network is one large distributed FCFS queue. According to the definition of fairness presented in Section IV-C.1, the distributed FCFS operation is in fact fair to all users in the network. The ability of the protocol to guarantee equal opportunity for all users of any location is investigated later in Section ??.

3) *DQBR Measured Fairness Performance:* The *HORNET* simulator was used to measure the ability of DQBR to provide fairness control. To demonstrate the fairness control, the throughput of each node is measured when the network is saturated. To do this, the traffic conditions of the simulation are such that the total network load on the observed wavelength is significantly greater than the capacity of the wavelength. Because the wavelength is oversubscribed, a queue management protocol is necessary because otherwise the queue depths will grow uncontrollably. To ensure that the simulations are realistic under such conditions, the *Random Early Detection* (RED) protocol for congestion control [?] is implemented in the simulator because it is expected that a similar protocol would be used in a commercial *HORNET* network. In reality, it is preferred that the congestion control protocol presented in [?] is used because it penalizes users that do not properly respond to the congestion control protocol. It is assumed in this work that all users will behave properly, and thus the RED protocol is used.

Figures 11 and 12 show that DQBR resolves the unfairness problem in the *HORNET* architecture. Initially, fixed-sized packets are used because DQBR was explained under that assumption. Variable-sized packets are addressed in Section ???. Figure 11 shows the throughput for nodes sending packets to Node 18 on a 25-Node *HORNET* network. With DQBR, the throughput is equal for all nodes, whereas *without* DQBR, the nodes close to Node 18 have a very difficult time sending packets to Node 18. Also, recall from Section IV-C.2 that DQBR is designed to eliminate the unfairness condition that occurs in IEEE 802.6 due to

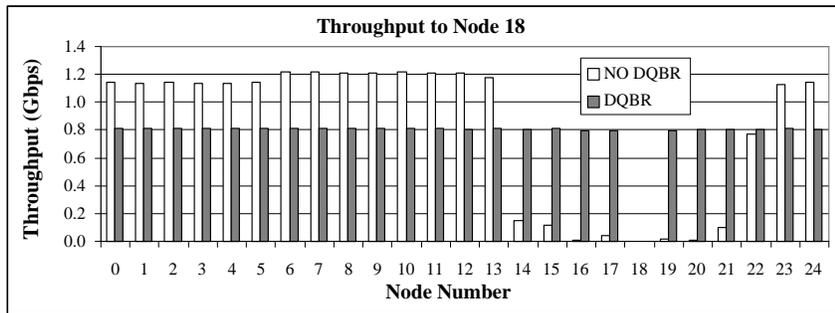


Fig. 11. Throughput for VOQ number 18 for the 25 nodes on a *HORNET* network. VOQ number 18 corresponds to Wavelength 18, which is received by Node 18. The total network load for Wavelength 18 is 1.5 times its capacity. There is enough propagation delay between nodes to hold 50 control frames.

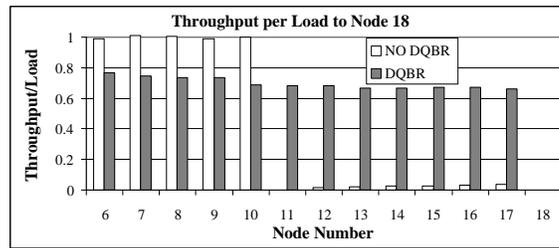


Fig. 12. *Throughput divided by load* for VOQ number 18 for several nodes. VOQ number 18 corresponds to Wavelength 18, which is received by Node 18. In this simulation, the load on VOQ 18 in Node 10 is 9.33 Gbps, and the load on VOQ 18 in Node 11 is 4.67 Gbps. All other nodes have only a small load.

propagation distance between nodes [?]. In this simulation, there is enough propagation distance between nodes to hold 50 control frames, yet the throughput is still equal for all nodes when DQBR is used. Thus, it is clear that propagation distance does not affect the fairness of DQBR.

The simulator computed the total throughput for the simulations presented in Figure 11. *Without* DQBR, the throughput is 0.999, while *with* DQBR the throughput is 0.965. Thus, the penalty of DQBR is only 3.5%. This is a very minor penalty, considering the tremendous benefit it provides. The cause of the minor penalty is described thoroughly in [2]

Figure 12 shows a simulation with an unbalanced traffic case that can be a major problem for an unfair architecture. In this traffic case, Node 10 has 9.33 Gbps of traffic arriving to its queue destined for Node 18, Node 11 has 4.67 Gbps destined for Node 18, and all other nodes have very little traffic. The wavelength can only support 10 Gbps, so it is heavily oversubscribed. As the figure shows, without DQBR controlling the fairness, the nodes close to Node 18 are unable to transmit packets on Wavelength 18, while in the DQBR network, all nodes have an equal ratio of *throughput to load* for Wavelength 18.

To justify the fairness of this situation, imagine that the simulation results of Figure 12 were generated by the following network conditions. There are 250 users of a *HORNET* network. All are sending 58.3 Mbps of traffic to Node 18. Attached to Node 10 are 160 of those users, while 80 are accessing the network through Node 11. The other ten users are each using one of the other nodes shown in the plot of Figure 12. Under a scheme that equalizes bandwidth to the nodes, such as DQBR's bandwidth balancing [?, ?], the users attached to Node 10 would be required to reduce their throughput to 29.7 Mbps each, while all other users continue to transmit at 58.3 Mbps. This is because Node 10 would be allocated 4.753 Gbps, allowing Node 11 to transmit at 4.664 Gbps, and all other nodes to transmit at 58.3 Mbps. This might be fair if nodes were users, but instead the users of Node 10 are penalized because they happen to be grouped in the same location. In contrast, DQBR allocates each node a *throughput to load* ratio of approximately 0.7, as shown in Figure 12, and thus each user receives 40 Mbps.

To verify this result further, the average packet latency and the packet drop probability can be analyzed. The average delay suffered by packets in the VOQs for Node 18 is plotted in Figure 13. The results are generated using the same unbalanced traffic case described above. As the figure shows, with DQBR packets suffer the same latency in all nodes. The packet drop probability at each node is shown in Figure 14 for the unbalanced traffic case. The packet loss probability in a *HORNET* network at all nodes is nearly equal when DQBR is used. Thus, all users of the network will experience the same packet loss probability, and as a result the transport control protocol will regulate the users' load in the same way.

V. *HORNET* CONTROL CHANNEL DESIGN

The control channel design significantly impacts both the performance and the cost of the network. This section discusses two important aspects of the control channel design. First, the structure and the optimal length of the control channel frame are described. Then, the synchronization of the control channel with the packets on the payload wavelengths is discussed. Included in this discussion is an experimental demonstration of a frame synchronization protocol developed for *HORNET*.

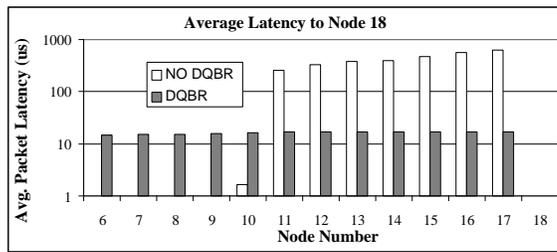


Fig. 13. Average packet latency in each *HORNET* node for the unbalanced traffic case (Nodes 10 and 11 have a heavy traffic load for Node 18 while all others have light traffic).

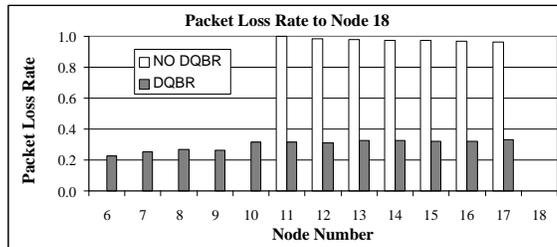


Fig. 14. Packet loss probability in each *HORNET* node for the unbalanced traffic case (Nodes 10 and 11 have a heavy traffic load for Node 18 while all others have light traffic).

A. Control Channel Frame Length

As was described in Section IV, the MAC protocol requires all packets to be inserted to coincide with the beginning of the control channel frame. If a packet that is being transmitted on a particular wavelength is completed somewhere in the middle of the control frame, then the rest of the control channel frame duration on that wavelength must go unused. The unused time period on the wavelength is considered overhead and detracts from the performance of the network. The minimization of this overhead occurs at the optimal match between control channel frame length and the distribution of IP packet sizes. Figure 15 illustrates the components of the control channel frame. Note that for a network with 128 wavelengths, the minimum frame size is 37 bytes.

The *HORNET* transmitter adds a header onto the front of the packet and a trailer to the rear of the packet. The packet size distribution used to determine the optimal control channel frame length must include the payload data, the TCP/IP header, and the *HORNET* header and trailer. The trailer indicates whether the packet is complete or segmented, as discussed in Section ???. The header has several purposes. It includes guard time for transmitter tuning, a sequence for bit-synchronization, source and destination address information, control information, and a frame check sequence (FCS) for determining the integrity of the packet. It is anticipated that a futuristic commercial implementation of *HORNET* would use a header of 16 bytes. The structure of a *HORNET* packet is illustrated in Figure 16. The values for *guard time* and *synchronization sequence* are heavily dependent on the progression of the transmitter and receiver technology.

The expected overhead for varying frame sizes is shown in Figure 17 (a). Smaller frame sizes result in less overhead because of the significant amount of small packets (see Figure ?? and because of the fact that when a packet finishes before the end of a frame, the remainder of the frame duration is *overhead*. The calculation uses the packet size distribution of Figure 8, and does not consider the overhead due to packet segmentation. For the calculation, the *HORNET* header/trailer is 16 bytes. The overhead is defined as the percentage of the transmission that does not contain payload, where the payload in this analysis includes the TCP/IP header and the data within the packet. The overhead bytes include the *HORNET* header and any *unused* bytes after the packet (before the next control frame).

The simulator can be used to verify the optimal control channel frame size. Figure 17 compares the performance of a 17-node

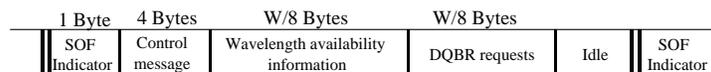


Fig. 15. Information contained within each control channel frame.

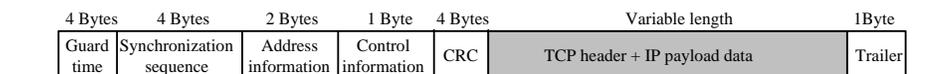


Fig. 16. Contents of a *HORNET* packet.

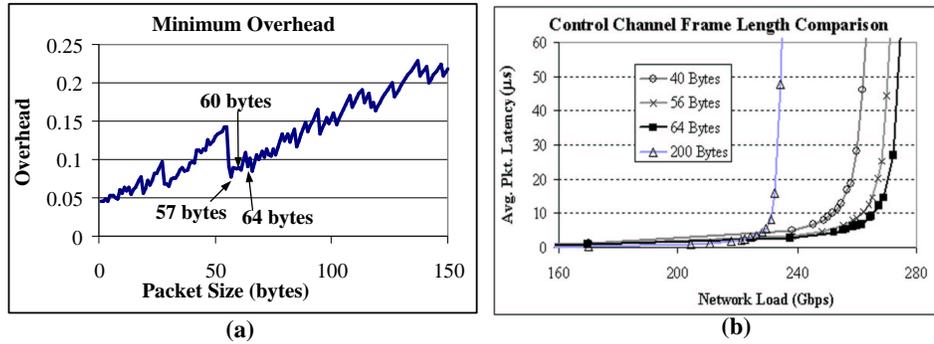


Fig. 17. (a) Expected overhead for *HORNET* with a packet size CDF shown in Figure 7. (b) Simulated performance with varying control frame sizes.

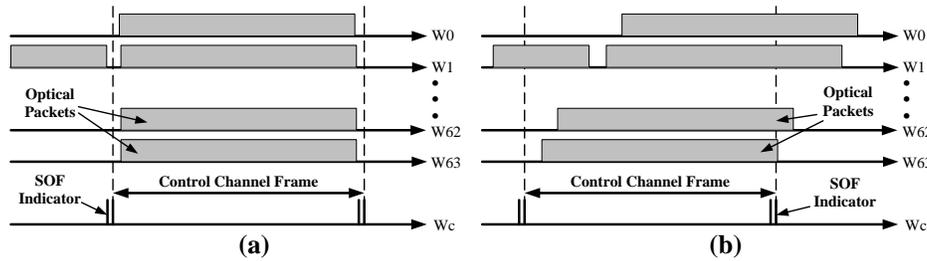


Fig. 18. Optical packets in a WDM system (a) *before* propagating through SMF, and (b) *after* propagating through SMF. W_c = control wavelength.

HORNET network with control channel frame sizes of 40, 56, 64, and 200 bytes while using the variable-sized packet distribution shown in Figure 7 (a). As Figure 17 shows, with a large control channel frame size (e.g. 200 bytes), performance is seriously degraded because of the amount of overhead incurred when transmitting short packets, which happen to dominate the packet size distribution. Performance is relatively similar for the three short control channel frame sizes, but 64 bytes has the best performance.

B. Dispersion Management for Control Frame Alignment

Frame misalignment occurs in *HORNET* because the group velocity dispersion (GVD) of standard single mode fiber (SMF) causes optical signals on different wavelengths to travel at different speeds. The misalignment is illustrated in Figure 18. If a packet becomes misaligned with the SOF indicators, then another packet may collide with it when inserted into the ring. It is necessary to insert dispersion compensating fiber (DCF) throughout the network to reverse the effect of the GVD of SMF. Optimized lengths of DCF can be concatenated with the transmission fiber at each node. It has been shown that commercially available fibers can correct the relative drift of the payload wavelengths that can occur in *HORNET* to within 10 ps/(km of SMF) [2]. If 1310 nm is used for the control channel wavelength, DCF is not sufficient to keep the SOF indicator aligned with the packets on the payload wavelength. However, the solution is simple because the control channel wavelength is separated from the payload wavelengths in every node. Fiber cable delays can be used to realign the control channel wavelength and the payload wavelengths.

C. Control Channel Frame Synchronization Protocol

In every *HORNET* node, the control channel is processed and retransmitted while packets on the payload wavelengths pass through an all-optical path. The control channel must be retransmitted in perfect alignment with those packets. However, two issues can prevent that from happening. The first issue is a lack of synchronization between the incoming and the retransmitted control channel at each node, while the second issue is the difficulty in manufacturing a node with a perfect match between the payload path and the control channel path. Both of these issues are solved with the establishment of a frame synchronization protocol for *HORNET*.

The node's control process is in general not perfectly synchronized with the incoming control channel, and thus the process will begin at a random moment with respect to the moment of arrival of the SOF indicator, which drives the control process. The random misalignment adds stochastically at each node, resulting in a large variance after several nodes of propagation. This issue can be easily solved by using a phase-locked loop (PLL) within the control channel receiver to synchronize the control channel process with the incoming control channel bit stream. Thus, the first requirement of the frame synchronization protocol is the use of a PLL to obtain synchronization from the incoming control channel.

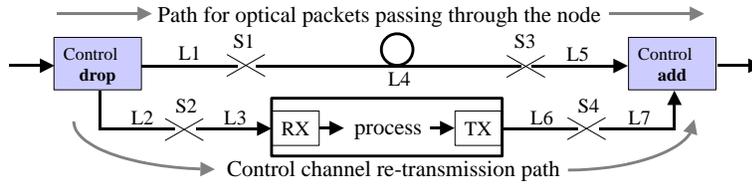


Fig. 19. The control channel path and the payload wavelength path. S_n denotes splice locations, L_n denotes fiber lengths.

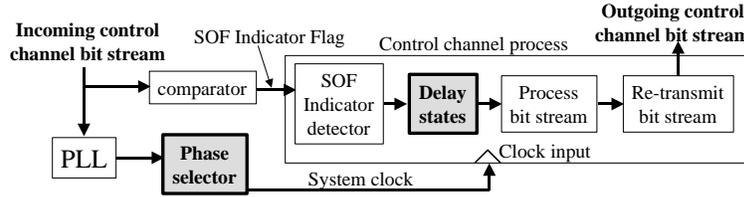


Fig. 20. The output phase of the PLL and the delay states are controlled by the node to provide perfect control channel frame synchronization.

The second issue that causes control channel frame misalignment is designing, manufacturing, and maintaining a perfect match in propagation delay between the control channel path and the payload wavelength path. Figure 19 illustrates the two paths, including splice locations. To make the paths match, splices and fiber lengths must be tightly controlled. More importantly, the design of the electronics and micro-code are critical because every modification in the design process and *every upgrade after the product release* may cause a path difference. Any error due to the micro-code will be present in every node, and thus the resulting misalignment will add as packets traverse the ring.

This issue is solved in the frame synchronization protocol by *automatically* calibrating the control channel path propagation delay to match the payload wavelength path. Figure 20 shows the important components involved in the calibration. The two highlighted components, the PLL phase selector and the delay states, are used to adjust the propagation delay through the control channel path. The node programs the delay states to adjust the propagation delay in increments of a process clock cycle. Also, the node can control the output phase of the PLL, which dictates the moment at which the incoming SOF indicator comparator output flag is sampled. Sampling the SOF comparator output flag near the beginning of its duration will shorten the propagation delay of the control channel path, just as sampling the flag near its end will lengthen the propagation delay.

The calibration requires two steps to achieve nearly perfect SOF indicator alignment. The first step is a laboratory calibration (*lab cal*) to put the node in a position to perform its auto alignment when in the system. This is a manual step performed by an operator before the node is installed in the network. Once the node is placed in the network and is turned on, one of the first things it must do is to perform the in-system calibration (*IS-cal*), the second calibration step. The two cal steps are thoroughly described in [2, 3].

D. Frame Synchronization Demonstration

An experimental testbed was assembled to demonstrate the *HORNET* frame synchronization calibration procedure [2]. As shown in Figure 21, three experimental *HORNET* nodes are connected together. The nodes use a PLL with an adjustable output phase to synchronize the control process with the incoming control channel, as specified by the protocol. Gigabit Ethernet (GbE) is used for the control channel, and thus the SOF indicator is the GbE 'comma' byte (*1100000101*). The lab cal procedure was performed on the nodes to set the reference condition. The IS-cal was then performed on Node 1. The IS-cal of Node 2 is described below.

Figure 23 shows the result of the experimental demonstration. Figure 24 compares the alignment of the SOF indicator and a packet after two nodes of propagation with and without the frame alignment protocol. The time-lapse image of Figure 24 (a) shows the random misalignment that occurs without the protocol.

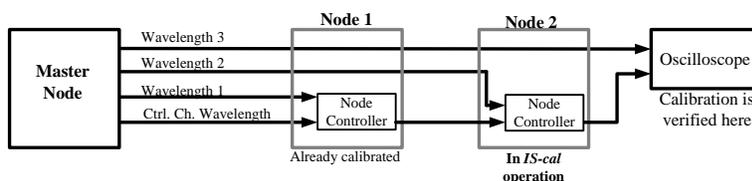


Fig. 21. The setup of the *IS-cal* procedure for a node downstream of a previously calibrated node.

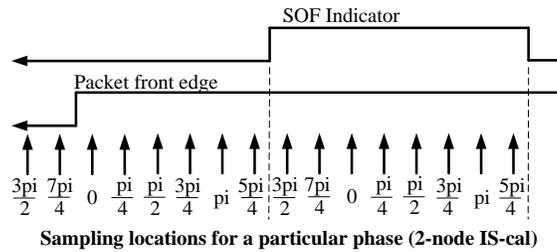


Fig. 22. The location of the samples for all phases for the two incoming waveforms in the *IS-cal* procedure of the second node.

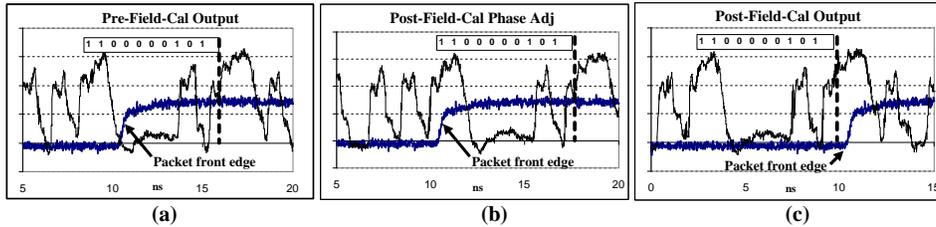


Fig. 23. (a) Alignment of retransmitted control channel SOF indicator with a packet passing through the node before the *IS-cal*; (b) After the phase adjustment portion of the *IS-cal*; (c) After the complete *IS-cal*.

The results of this experiment shown in Figures 23 and 24 show that the alignment accuracy is within *one bit* of the control channel bit rate (1 ns in this case, since GbE is used). This is because the adjustment precision of the PLL is $\frac{1}{80}$ of a clock cycle, or one bit. In general, the accuracy may only be as good as a few bits because of the possibility that the correct alignment would have the clock sampling the 'edge' of the SOF indicator (in such a case the sampling clock is adjusted slightly). As long as the accuracy is within *one byte*, then only one byte of guard band is necessary, and thus only one byte of overhead is used.

VI. HORNET NODE AND ITS KEY SUBSYSTEMS

A. HORNET Node Design

Figure 25 is a block diagram of the design of the *HORNET* node. At the input of the node, a wavelength drop removes the control channel wavelength from the WDM ring. The wavelength for the control channel should be well separated from the payload wavelengths (e.g. 1310 nm) to allow inexpensive transmitters to be used for the transmission of the control channel. The incoming control channel is received in the *Node Controller* where the bit stream is analyzed for wavelength availability information, DQBR requests, and any other control information.

After passing through the control channel wavelength drop, the signals on the payload wavelengths traverse an optimized length of dispersion compensating fiber (DCF) optic cable. This fiber cable is necessary to keep all of the packets on the payload wavelengths aligned with the control channel frames, as described in Section V-B.

Following the DCF, an optical amplifier is used to boost the power of all of the signals on the payload wavelengths. Note that it is important to drop the control channel wavelength before the amplifier because it only provides gain to the payload wavelengths. The amplifier will likely significantly attenuate signals on wavelengths outside of the payload wavelength band, such as at 1310 nm, because of filtering components within the amplifier subsystem. Several issues concerning the optical amplifier in the *HORNET* nodes will be discussed later in this work. Potential technologies for the amplifiers are discussed in Section VI-C. System issues related to the amplifiers are discussed in Section VIII. As is shown in that section, optical amplifiers are not necessarily contained in all nodes (thus the amplifier is illustrated with a dotted line in Figure 25).

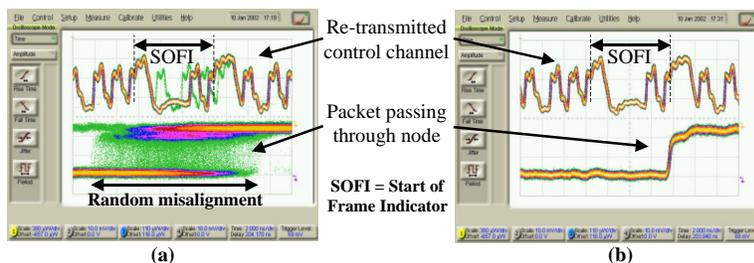


Fig. 24. Time-lapse image of the retransmitted control channel and packets after two nodes of propagation. (a) Random misalignment with no frame synchronization protocol. (b) Perfect alignment with the protocol.

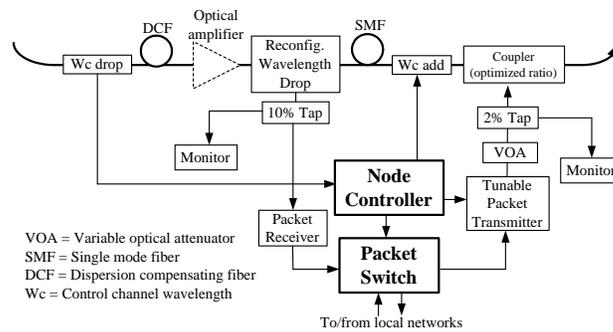


Fig. 25. Block diagram of the *HORNET* node.

After the WDM signal receives the necessary boost, a *wavelength drop* removes the wavelength(s) from the ring that is (are) destined for the node. Optical packets dropped into the node are received by an asynchronous packet receiver [2] and are then sent to the packet switch, where they are switched onto the local network to which they are destined. Ideally, the wavelength drop is reconfigurable, such that it can allow the network to provision a particular set of wavelengths for a node in order to efficiently accommodate varying traffic patterns. The reconfigurable optical drop should be designed such that it can drop between 1 and M wavelengths, where M is the most wavelengths the node will require, and is typically much smaller than W (the total number of wavelengths in the network).

The payload wavelengths will then pass through a *wavelength add* that multiplexes the control channel wavelength onto the backbone. It is imperative for the control channel frames to be multiplexed in perfect synchronization with the packets on the payload wavelengths, so a SMF delay line is located just before the wavelength add. The delay line (in addition to the other components between the control channel wavelength add and drop) holds the packets on the payload wavelengths while the control channel is being processed. Although the delay line can be designed to approximate the necessary delay to match the propagation delay of the control channel path, it is very difficult to maintain a perfect match between the payload wavelength path and the control channel propagation path, especially considering that the electronic design of the control channel propagation path will be upgraded several times after the product is deployed. To solve this problem, a calibration routine was developed to allow the *node controller* to automatically adjust the propagation delay through the control channel path such that it nearly perfectly matches the payload wavelength path. The calibration is described and demonstrated in [2, 3].

Near the output of the node, the fast-tunable packet transmitter inserts packets onto the backbone ring on the wavelength that is received by the packet's destination node. A variable optical attenuator (VOA) is placed at the output of the tunable transmitter to control the output power of the transmitter. This is necessary because the node must transmit its packets at a power level to match the power level of the packets passing through the node. This power level is dependent upon the location of the nearest optical amplifier (recall that amplifiers are not necessarily located in all nodes).

B. Fast-Tunable Packet Transmitter

The design of the tunable transmitter for the *HORNET* node is discussed and demonstrated in [12]. The transmitter has been demonstrated with a GCSR tunable laser [13] and with an SG-DBR tunable laser [14]. Experimental results have proven that the tunable transmitter developed for *HORNET* can tune throughout the conventional wavelength band between arbitrary pairs of wavelengths in less than 20 ns.

Other researchers have also investigated the fast-tunable packet transmitter subsystem [16–18]. Similar results have been achieved in each of those projects. The combination of the results generated in that research as well as in the *HORNET* project prove that the subsystem will likely be a viable technology in the very near future. Ultimately, it is desirable for the tuning duration to be consistently less than a few nanoseconds so that the overhead due to laser tuning is only a few bytes.

C. Constant-Gain Optical Amplifiers

The most interesting subsystem in the *HORNET* node is the optical amplifier. In conventional WDM networks, Erbium-doped fiber amplifiers (EDFAs) are used to supply gain. To provide the necessary output power for today's dense WDM systems, EDFAs must be operated in saturation. When the amplifier is not saturated, the gain is linear (the output power grows linearly with input power). In saturation, the gain is no longer linear, and thus it is dependent upon the input power (or output power) [19, 20]. This is not a major problem for conventional networks because the instantaneous power at the input of the EDFAs in a link is held constant by using techniques such as scrambling, coding, and transmitting idle packets when no data packets are to be sent.

However, in the *HORNET* network the instantaneous power at any point in the link is very dynamic. This is a result of the fact that nodes only transmit packets when they have a packet to transmit. Packet transmissions will thus occur at random. As a result, at any point on the link at any moment, the number of wavelengths carrying packets is random, and thus the optical power

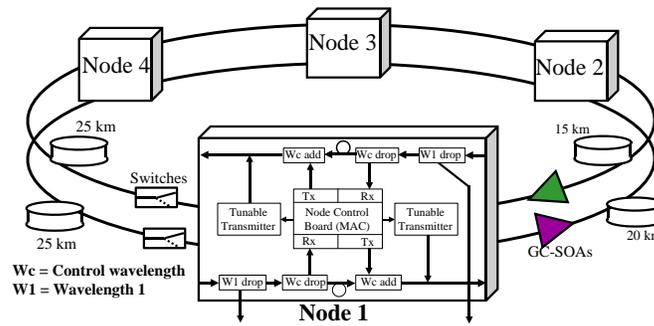


Fig. 26. Testbed constructed for the *HORNET* experimental demonstrations.

is random. The dynamic power on the network will affect the gain of the amplifier. As packets pass through the amplifier, the gain they receive will be dynamic, causing the amplitude of the packets at the output of the amplifier to be distorted. It is very difficult to design a receiver that can properly receive the bits in a packet with highly dynamic amplitude.

As a result, conventional EDFAs cannot be used in the *HORNET* network. The amplifiers for *HORNET* must provide constant gain when faced with dynamic conditions. Three potential solutions have recently emerged: gain-clamped semiconductor optical amplifiers (also known as *linear optical amplifiers*) [24], gain-clamped EDFAs [21, 22], and transient-control EDFAs [23].

The operation of gain-clamped semiconductor optical amplifiers (GC-SOAs) is based on a well-known principle from laser physics. When a gain medium is lasing, the inversion (the fraction of atoms that are energized) remains constant under dynamic conditions. As a result, the gain is constant, even when the input power is dynamic. GC-SOAs use a vertical cavity laser within the gain medium to *clamp* the gain [24]. The performance of the amplifiers has been demonstrated under dynamic conditions [24], although they have not been tested in an optical packet environment.

Gain-clamped EDFAs (GC-EDFAs) have a similar principle of operation. The difference is that the GC-EDFA uses an optical feedback loop around the Erbium-doped fiber to establish a laser within the gain medium [21]. GC-EDFAs have been successfully demonstrated in an optical packet environment [22], but the necessary output power for a high-capacity network like *HORNET* has not yet been achieved. Also, relaxation oscillations may be a problem, as shown by the simulations in [21].

Transient-controlled EDFAs, the third promising technology for constant-gain amplifiers, are quite different from the two types of gain-clamped amplifiers. A transient-controlled EDFA maintains constant gain by monitoring the changes in the input power and modulating the pump power(s) accordingly. These amplifiers have not yet been demonstrated in optical-packet-based networks like *HORNET*, but they have been successfully demonstrated under dynamic conditions [23]. They may very well have the potential to be a good solution for optical amplification in *HORNET*, but only experimentation can verify this.

VII. *HORNET* EXPERIMENTAL DEMONSTRATION

An experimental laboratory testbed was built to demonstrate the *HORNET 2FBPSR architecture*. Figure 26 shows the 4-node *HORNET* testbed. A node contains a wavelength drop for the node's drop-wavelength on each ring, a tunable transmitter subsystem for each ring, a wavelength add and drop for the control channel wavelength on each ring, and a *node controller*. The node's protocols are implemented in programmable logic devices (PLDs) on the node-controller circuit board clocked at 125 MHz. A Gigabit Ethernet (GbE) chip set is used for the transmission and reception of the control channel in the testbed.

The testbed was used to demonstrate the control-channel-based MAC protocol. Figure 27 shows two wavelengths of the optical output of Node 1. The modulated packets are the packets inserted by Node 1, while the un-modulated packets are inserted by nodes upstream of Node 1 destined for downstream nodes. Node 1 uses the control channel coming from upstream to determine the available wavelengths during every control frame. The figure verifies that the packets were successfully transmitted using the MAC protocol without causing collisions. The space between the packets is quite small because of the fast tuning time and the excellent synchronization provided by the MAC protocol.

The testbed was also used to demonstrate the survivability of *HORNET's* architecture. As shown in Figure 26, two optical switches are placed between Nodes 1 and 4 to cut the network. A function generator controls the switches so that the network can be periodically cut and repaired. An oscilloscope is used to monitor the output on the optical drops of one node at a time. Generally, only the transmitters in one node are turned on at a time so that one source-destination connection can be monitored. Four spools of fiber are located on the network (two in each direction) to provide realistic propagation delays between nodes.

The *node controller* circuit board stores the routing table for the node. Under normal circumstances a node transmits to the two nodes to its right using the counter-clockwise (CCW) ring, and to the other node using the clockwise (CW) ring. If the control channel is interrupted in either control channel receiver, then the node controller determines that a cut has occurred. It reprograms its routes so that no packets are sent toward the cut. If the node learns of the cut from a message on the control channel, then the node controller reads the source address of the message and determines the location of the cut. Since the controller knows that the

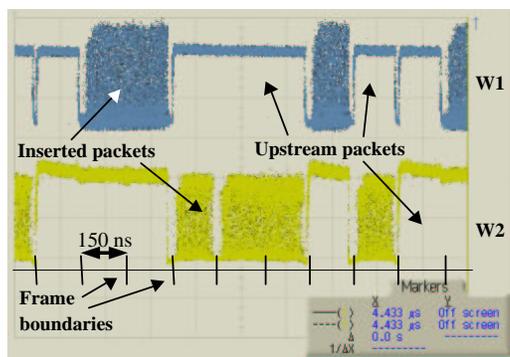


Fig. 27. Display on the DCA showing the successful transmission of packets using the MAC protocol. W1 = Wavelength 1.

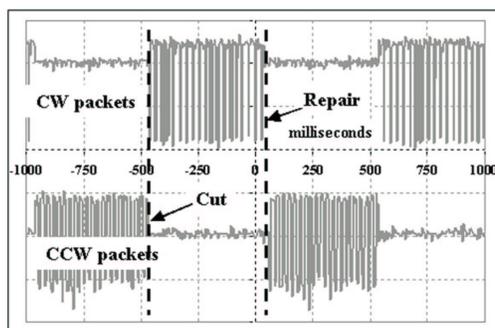


Fig. 28. Packets transmitted from Node 3 to Node 1 before and after a fiber cut.

nodes are numbered in increasing order in the CCW direction, it can determine if any of its routes need to be changed based on the location of the cut. It then reprograms its routes as necessary.

To monitor the connection between Node 3 and Node 1, an oscilloscope is connected to the receivers of *Node 1*, and only the transmitters in *Node 3* are connected to the network. When the cut occurs, both Node 1 and Node 4 detect it. Both send messages on the control channel around the ring away from the cut notifying other nodes. Node 3 receives the message and adjusts its routes. Figure 28 shows the occurrence from the perspective of Node 1. Node 1 originally receives packets from Node 3 on the CCW ring. However, after the cut, Node 3 is forced to use the CW ring. When it learns that the cut is repaired, it resumes transmitting in the CCW direction.

Because *HORNET* does not use point-to-point link-based protocols, no setup time is required to begin using a new path. Thus, the restoration of a path happens nearly instantly. The only cause for downtime between two nodes is the propagation delay of the control messages around the ring [4]. Figure 29 shows a zoomed-in view of the cut event and the repair event from the perspective of Node 1 as it receives packets from Node 3. The down time for the connection between Node 3 and Node 1 corresponds to the length of the fiber spools in the testbed [4].

VIII. HORNET SYSTEM ANALYSIS

IX. SUMMARY

REFERENCES

- [1] Resilient Packet Ring Alliance, "An Introduction to Resilient Packet Ring Technology," White Paper, available at <http://www.rpralliance.org>, October 2001.
- [2] I. M. White, *A New Architecture and Technologies for High-Capacity Next Generation Metropolitan Networks*. PhD dissertation, Stanford University, Department of Electrical Engineering, August 2002.
- [3] I. M. White, M. S. Rogge, K. Shrikhande, and L. G. Kazovsky, "Design of a Control-Channel-Based MAC Protocol for HORNET," *Journal of Optical Networking*, submitted September 2002.
- [4] I. M. White, M. S. Rogge, Y-L. Hsueh, K. Shrikhande, and L. G. Kazovsky, "Experimental Demonstration of the HORNET Survivable Bi-directional Ring Architecture," In *Optical Fiber Communications Technical Digest*, Anaheim, CA, pp. 346–349, March 2002.
- [5] N. M. Froberg, S. R. Henion, H. G. Rao, B. K. Hazzard, S. Parikh, B. R. Romkey, and M. Kuznetsov, "The NGI ONRAMP Test Bed: Reconfigurable WDM Technology for Next Generation Regional Access Networks," *Journal of Lightwave Technology*, Vol. 18, pp. 1697–1708, December 2000.
- [6] M. J. Spencer and M. A. Summerfield, "WRAP: A Medium Access Control Protocol for Wavelength-Routed Passive Optical Networks," *Journal of Lightwave Technology*, Vol. 18, pp. 1657–1676, December 2000.
- [7] R. Gaudino, A. Carena, V. Ferrero, A. Pozzi, V. De Feo, P. Gigante, F. Neri, and P. Poggiolini, "RINGO: A WDM Ring Optical Packet Network Demonstrator," In *Proceedings of the 27th European Conference on Optical Communications*, Amsterdam, Netherlands, September 2001.
- [8] A. Smiljanic and B. Loehfel, "Performance Evaluation of Optical Ring Network Based on Composite Packet Switching," In *Optical Fiber Communications Technical Digest*, Anaheim, CA, pp. 286–287, March 2002.

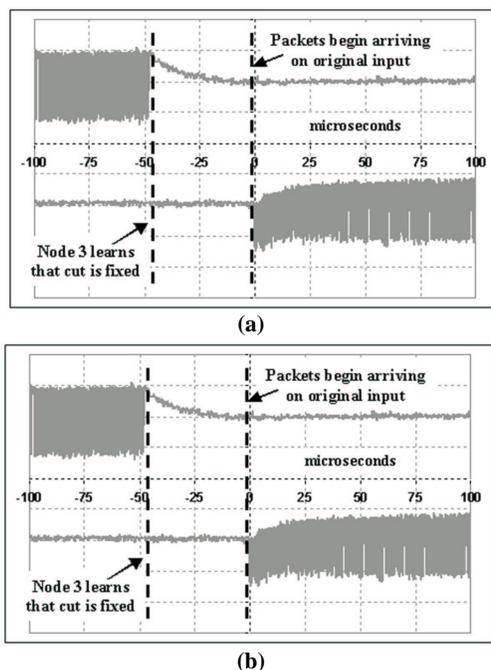


Fig. 29. (a) Restoration delay for the path from Node 3 to Node 1. (b) Transition of routes in Node 3 after the cut is reported as repaired (delay is only due to differences in fiber length along paths).

- [9] I. Haque, Wilhelm Kremer, and K. Raychaudhuri. Self-Healing Rings in a Synchronous Environment. *IEEE LTS*, pp. 30–37, November 1991.
- [10] D-R. Wonglumsom, I. M. White, K. Shrikhande, M. S. Rogge, S. M. Gemelos, F-T. An, Y. Fukashiro, M. Avenarius, and L. G. Kazovsky. "Experimental Demonstration of an Access Point for HORNET - A Packet-Over-WDM Multiple Access MAN," *IEEE Journal of Lightwave Technology*, Vol. 18, pp. 1709-1717, December 2000.
- [11] Y. Tamir and G. Frazier, "High Performance Multi-Queue Buffers for VLSI Communication Switches," *In Proceedings of the 15th Annual Symposium on Computer Architecture*, pp. 343–354, June 1988.
- [12] K. Shrikhande, I. M. White, M. S. Rogge, F-T. An, E. S. Hu, S. S-H. Yam, and L. G. Kazovsky, "Performance Demonstration of a Fast-Tunable Transmitter and Burst-Mode Packet Receiver for HORNET," *In Optical Fiber Communications Technical Digest*, Anaheim, CA, pp. ThG2:1–ThG2:3, March 2001.
- [13] B. Broberg, P-J. Rigole, S. Nilsson, L. Andersson, and M. Renlund, "Widely Tunable Semiconductor Lasers," *In Optical Fiber Communications Technical Digest*, San Diego, CA, pp. WH4:1–WH4:3, March 1999.
- [14] Y. A. Akulova, C. Schow, A. Karim, S. Nakagawa, P. Kozodoy, G. A. Fish, J. DeFranco, A. Dahl, M. Larson, T. Wipiejewski, D. Pavinski, T. Butrie, and L. A. Coldren, "Widely-Tunable Electroabsorption-Modulated Sampled Grating DBR Laser Integrated with Semiconductor Optical Amplifier," *In Optical Fiber Communications Technical Digest*, Anaheim, CA, pp. 536–537, March 2002.
- [15] M. L. Jajewski, J. Barton, and L. A. Coldren, "Widely Tunable Directly Modulated Sampled-Grating DBR lasers," *In Optical Fiber Communications Technical Digest*, Anaheim, CA, pp. 537–538, March 2002.
- [16] O. A. Lavrova, G. Rossi, and D. J. Blumenthal, "Rapid Tunable Transmitter with Large Number of ITU Channels Accessible in Less Than 5 ns," *In Proceedings of the 26th European Conference on Optical Communications*, pp. 23–24 (Paper 6.3.5), September 2000.
- [17] J. Gripp, M. Duelk, J. Simsarian, S. Chandrasekhar, P. Bernasconi, A. Bhardwaj, Y. Su, K. Sherman, L. Buhl, E. Laskowski, M. Cappuzzo, L. Stulz, M. Zirngibl, O. Laznicka, T. Link, R. Seitz, P. Mayer, and M. Berger, "Demonstration of a 1.2 Tb/s Optical Packet Switch Fabric (32×40 Gb/s) Based on 40 Gb/s Burst-Mode Clock-Data-Recovery, Fast Tunable Lasers, and High-Performance N×N AWG," *In Proceedings of the 27th European Conference on Optical Communications, Postdeadline Session 3*, Amsterdam, Netherlands, September 2001.
- [18] S. J. B. Yoo, Y. Bansal, Z. Pan, J. Cao, V. K. Tsui, S. K. H. Fong, Y. Zhang, J. Taylor, H. J. Lee, M. Jeon, and V. Akella, "Optical-Label Based Packet Routing System with Contention Resolution in Wavelength, Time, and Space Domains," *In Optical Fiber Communications Technical Digest*, Anaheim, CA, pp. 280–282, March 2002.
- [19] Y. Sun, J. L. Zyskind, and A. K. Srivastava, "Average Inversion Level, Modeling, and Physics of Erbium-Doped Fiber Amplifiers," *IEEE Journal of Selected Topics in Quantum Electronics*, Vol. 3, pp. 991–1007, August 1997.
- [20] L. Tancevski, A. Bononi, and L. A. Rusch, "Output Power and SNR Swings in Cascades of EDFA's for Circuit- and Packet-Switched Optical Networks," *Journal of Lightwave Technology*, Vol. 17, pp. 733–742, May 1999.
- [21] A. Bononi and L. Barbieri, "Design of Gain-Clamped Doped-Fiber Amplifiers for Optimal Dynamic Performance," *Journal of Lightwave Technology*, Vol. 17, pp. 1229–1240, July 1999.
- [22] M. Karasek, A. Bononi, L. A. Rusch, and M. Menif, "Gain Stabilization in Gain Clamped EDFA Cascades Fed by WDM Burst-Mode Packet Traffic," *Journal of Lightwave Technology*, Vol. 18, pp. 308–313, March 2000.
- [23] W. S. Wong, H-S. Tsai, C-J. Chen, H. K. Lee, and M-C. Ho, "Novel Time-Resolved Measurements of Bit-Error-Rate and Optical-Signal-to-Noise-Ratio Degradations Due to EDFA Gain Dynamics in a WDM Network," *In Optical Fiber Communications Postdeadline Papers*, Anaheim, CA, pp. 515–516, March 2002.
- [24] E. Tangdiongga, J. J. J. Crijns, L. H. Spiekman, G. N. van den Hoven, and H. de Waardt, "Performance Analysis of Linear Optical Amplifiers in Dynamic WDM Systems," *IEEE Photonics Technology Letters*, Vol. 14, pp. 1196–1198, Aug. 2002.

Bibliography

- [1] L. G. Kazovsky, S. Benedetto, and A. Willner. *Optical Fiber Communication Systems*, chapter 7, page 514. Artech House, 1996.
- [2] C. R. Giles and E. Desurvire. Modeling Erbium-doped fiber amplifiers. *Journal of Lightwave Technology*, 9(2):271–283, February 1991.
- [3] Y. Frignac, G. Charlet, W. Idler, R. Dischler, P. Tran, S. Lanne, S. Borne, C. Martinelli, G. Veith, A. Jourdan, J-P. Hamaide, and S. Bigo. Transmission of 256 wavelength-division and polarization-division-multiplexed channels at 42.7 Gb/s (10.2 Tb/s capacity) over 3x100 km of Teralight fiber. In *Optical Fiber Communications Postdeadline Papers*, pages FC5:1–FC5:3, March 2002.
- [4] D. G. Foursa, C. R. Davidson, M. Nissov, M. A. Mills, L. Xu, J. X. Cai, A. N. Pilipetskii, Y. Cai, C. Breverman, R. R. Cordell, T. J. Carvelli, P. C. Corbett, H. D. Kidorf, and N. S. Bergano. 2.56 Tb/s (256x10 Gb/s) transmission over 11,000 km using hybrid Raman/EDFAs with 80 nm of continuous bandwidth. In *Optical Fiber Communications Postdeadline Papers*, pages FC3:1–FC3:3, March 2002.
- [5] A Neukermans and R. Ramaswami. MEMS technology for optical networking applications. *IEEE Communications Magazine*, 39(1):62–69, January 2001.

- [6] J. E. Fouquet. Compact optical cross-connect switch based on total internal reflection in a fluid-containing planar lightwave circuit. In *Optical Fiber Communications Technical Digest*, pages TuM:1–TuM:3, March 2000.
- [7] C. R. Giles. Lightwave applications of fiber Bragg gratings. *Journal of Lightwave Technology*, 15(8):1391–1404, August 1997.
- [8] R. Ramaswami and K. N. Sivarajan. *Optical Networks*, chapter Appendix E, pages 569–576. Morgan Kaufmann, 1998.
- [9] D. J. Blumenthal, B-E. Olsson, G. Rossi, T. E. Dimmick, L. Rau, M. Masanovic, O. Lavrova, R. Doshi, O. Jerphagnon, J. E. Bowers, V. Kaman, L. A. Coldren, and J. Barton. All-optical label swapping networks and technologies. *Journal of Lightwave Technology*, 18(12):2058–2075, December 2000.
- [10] J. Gripp, M. Duelk, J. Simsarian, S. Chandrasekhar, P. Bernasconi, A. Bhardwaj, Y. Su, K. Sherman, L. Buhl, E. Laskowski, M. Cappuzzo, L. Stulz, M. Zirngibl, O. Laznicka, T. Link, R. Seitz, P. Mayer, and M. Berger. Demonstration of a 1.2 Tb/s optical packet switch fabric (32x40 Gb/s) based on 40 Gb/s burst-mode clock-data-recovery, fast tunable lasers, and high-performance NxN AWG. In *Proceedings of the 27th European Conference on Optical Communications, Postdeadline Session 3*, September 2001.
- [11] S. J. B. Yoo, Y. Bansal, Z. Pan, J. Cao, V. K. Tsui, S. K. H. Fong, Y. Zhang, J. Taylor, H. J. Lee, M. Jeon, and V. Akella. Optical-label based packet routing system with contention resolution in wavelength, time, and space domains. In *Optical Fiber Communications Technical Digest*, pages 280–282, March 2002.
- [12] Y. M. Lin, W. I. Way, and G. K. Chang. A novel optical label swapping technique using erasable optical single-sideband subcarrier label. *IEEE Photonics Technology Letters*, 12(8):1088–1090, August 2000.

- [13] G. Barish and K. Obraczka. World Wide Web caching: Trends and techniques. *IEEE Communications Magazine*, pages 178–185, May 2000.
- [14] T. T. Tay, Y. Feng, and M. N. Wijesundera. A distributed Internet caching system. In *Proceedings of the 25th Annual IEEE Conference on Local Computer Networks*, pages 624–633, November 2000.
- [15] Resilient Packet Ring Alliance. An introduction to resilient packet ring technology. White Paper, available at <http://www.rpralliance.org>, October 2001.
- [16] N. M. Froberg, S. R. Henion, H. G. Rao, B. K. Hazzard, S. Parikh, B. R. Romkey, and M. Kuznetsov. The NGI ONRAMP test bed: Reconfigurable WDM technology for next generation regional access networks. *Journal of Lightwave Technology*, 18(12):1697–1708, December 2000.
- [17] M. J. Spencer and M. A. Summerfield. WRAP: A medium access control protocol for wavelength-routed passive optical networks. *Journal of Lightwave Technology*, 18(12):1657–1676, December 2000.
- [18] M. A. Marsan, A. Bianco, E. Leonardi, M. Meo, and F. Neri. MAC protocols and fairness control in WDM multirings with tunable transmitters and fixed receivers. *Journal of Lightwave Technology*, 14(6):1230–1244, June 1996.
- [19] R. Gaudino, A. Carena, V. Ferrero, A. Pozzi, V. De Feo, P. Gigante, F. Neri, and P. Poggiolini. RINGO: A WDM ring optical packet network demonstrator. In *Proceedings of the 27th European Conference on Optical Communications*, September 2001.
- [20] A. Smiljanic and B. Loehfelm. Performance evaluation of optical ring network based on composite packet switching. In *Optical Fiber Communications Technical Digest*, pages 286–287, March 2002.

- [21] J-P. Faure, L. Noirie, A. Bisson, V. Sabouret, G. Leveau, M. Vigoureux, and E. Dataro. A scalable transparent waveband-based optical metropolitan network. In *Proceedings of the 27th European Conference on Optical Communications*, September 2001.
- [22] N. LeSauze, A. Dupas, E. Dotaro, L. Ciavaglia, M. H. M. Nizam, A. Ge, and L. Dembeck. A novel, low cost optical packet metropolitan ring architecture. In *Proceedings of the 27th European Conference on Optical Communications*, September 2001.
- [23] Y. Sun, J. L. Zyskind, and A. K. Srivastava. Average inversion level, modeling, and physics of Erbium-doped fiber amplifiers. *IEEE Journal of Selected Topics in Quantum Electronics*, 3(4):991–1007, August 1997.
- [24] D. A. Francis, S. P. DiJaili, and J. D. Walker. A single-chip linear optical amplifier. In *Optical Fiber Communications Postdeadline Papers*, pages PD13:1–PD13:3, March 2001.
- [25] A. Bononi and L. Barbieri. Design of gain-clamped doped-fiber amplifiers for optimal dynamic performance. *Journal of Lightwave Technology*, 17(7):1229–1240, July 1999.
- [26] W. S. Wong, H-S. Tsai, C-J. Chen, H. K. Lee, and M-C. Ho. Novel time-resolved measurements of bit-error-rate and optical-signal-to-noise-ratio degradations due to EDFA gain dynamics in a WDM network. In *Optical Fiber Communications Postdeadline Papers*, pages 515–516, March 2002.
- [27] T. Anderson, S. Owicki, J. Saxe, and C. Thacker. High speed switch scheduling for local area networks. *ACM Transactions on Computer Systems*, 11(4):319–352, November 1993.

- [28] I. M. White, M. S. Rogge, K. Shrikhande, Y. Fukashiro, D. Wonglumsom, F-T. An, and L. G. Kazovsky. Experimental demonstration of a novel media access protocol for HORNET: A packet-over-WDM multiple-access MAN ring. *IEEE Photonics Technology Letters*, 12(9):1264–1266, September 2000.
- [29] K. Shrikhande, I. M. White, D. Wonglumsom, S. M. Gemelos, M. S. Rogge, Y. Fukashiro, M. Avenarius, and L. G. Kazovsky. HORNET: A packet-over-WDM multiple access metropolitan area ring network. *IEEE Journal on Selected Areas in Communications*, 18(10):2004–2016, October 2000.
- [30] D. Wonglumsom, I. M. White, S. M. Gemelos, K. Shrikhande, and L. G. Kazovsky. HORNET - a packet-switched WDM metropolitan area ring network: Optical packet transmission and recovery, queue depth, and packet latency. In *1999 IEEE LEOS Annual Meeting Conference Proceedings*, pages 653–654, November 1999.
- [31] Y. Tamir and G. Frazier. High performance multi-queue buffers for VLSI communication switches. In *Proceedings of the 15th Annual Symposium on Computer Architecture*, pages 343–354, June 1988.
- [32] National Laboratory for Applied Network Research, Measurement Operations and Analysis Team. <http://pma.nlanr.net/Datacube/>.
- [33] IEEE Standard 802.6. Distributed queue dual bus (DQDB) subnetwork of a metropolitan area network (MAN), December 1990.
- [34] R. M. Newman, Z. L. Budrikis, and J. L. Hullett. The QPSX MAN. *IEEE Communications Magazine*, 26(4):20–28, April 1988.
- [35] E. L. Hahne, A. K. Choudhury, and N. F. Maxemchuk. Improving the fairness of Distributed-Queue-Dual-Bus networks. In *Infocom '90*, pages 175–184, 1990.

- [36] E. L. Hahne, A. K. Choudhury, and N. F. Maxemchuk. DQDB networks with and without bandwidth balancing. *IEEE Transactions on Communications*, 40(7):1192–1204, July 1992.
- [37] I. Haque, Wilhelm Kremer, and K. Raychaudhuri. Self-healing rings in a synchronous environment. *IEEE LTS*, pages 30–37, November 1991.
- [38] L. Y. Chan, C. K. Chan, D. T. K. Tong, F. Tong, and L. K. Chen. Upstream traffic transmitter using injection-locked Fabry-Perot laser diode as modulator for WDM access networks. *Electronics Letters*, 38(1):43–45, January 2001.
- [39] Y-L. Hsueh and K. Shrikhande. Mathematical model for group velocity dispersion. Unpublished work at the time of this dissertation, March 2001.
- [40] S. Floyd and V. Paxson. Difficulties in simulating the Internet. *IEEE/ACM Transactions on Networking*, 9(4):392–403, August 2001.
- [41] A. Karasaridis and D. Hatzinakos. Network heavy traffic modeling using α -stable self-similar processes. *IEEE Transactions on Communications*, 49(7):1203–1214, July 2001.
- [42] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.
- [43] R. Pan, B. Prabhakar, and K. Psounis. CHOKe: A stateless active queue management scheme for approximating fair bandwidth allocation. In *Infocom 2000*, pages 942–951, 2000.
- [44] C-C. Lin, W. A. Martin, and J. S. Harris. Optomechanical model of surface micromachined tunable optoelectronic devices. *IEEE Journal on Selected Topics in Quantum Electronics*, 8(1):80–87, January 2002.

- [45] M. S. Wu, E. C. Vail, G. S. Li, W. Yuen, and C. J. Chang-Hasnain. Tunable micromachined vertical cavity surface emitting laser. *Electronics Letters*, 31(19):1671–1672, September 1995.
- [46] F. Kano and Y. Yoshikuni. Frequency control and stabilization of broadly tunable SSG-DBR lasers. In *Optical Fiber Communications Technical Digest*, pages 538–540, March 2002.
- [47] M-C. Amann and J. Buus. *Tunable Laser Diodes*, chapter 7, pages 173–177. Artech House, 1998.
- [48] M-C. Amann and J. Buus. *Tunable Laser Diodes*, chapter 7, pages 167–173. Artech House, 1998.
- [49] V. Jayaraman, M. E. Heimbuch, L. A. Coldren, and S. P. DenBaars. Widely tunable continuous-wave InGaAsP / InP sampled grating lasers. *Electronics Letters*, 30(18):1492–1494, September 1994.
- [50] B. Broberg, P-J. Rigole, S. Nilsson, L. Andersson, and M. Renlund. Widely tunable semiconductor lasers. In *Optical Fiber Communications Technical Digest*, pages WH4:1–WH4:3, March 2002.
- [51] Y. A. Akulova, C. Schow, A. Karim, S. Nakagawa, P. Kozodoy, G. A. Fish, J. DeFranco, A. Dahl, M. Larson, T. Wipiejewski, D. Pavinski, T. Butrie, and L. A. Coldren. Widely-tunable electroabsorption-modulated sampled grating DBR laser integrated with semiconductor optical amplifier. In *Optical Fiber Communications Technical Digest*, pages 536–537, March 2002.
- [52] M. L. Jajewski, J. Barton, and L. A. Coldren. Widely tunable directly modulated sampled-grating DBR lasers. In *Optical Fiber Communications Technical Digest*, pages 537–538, March 2002.

- [53] C. F. C. Silva, V. Mikhailov, P. Bayvel, and A. J. Seeds. Zero frequency error locking of widely tunable lasers in high spectral efficiency systems using optical injection phase lock loops. In *Optical Fiber Communications Technical Digest*, pages 540–541, March 2002.
- [54] D. C. J. Reid, D. J. Robbins, A. J. Ward, N. D. Whitbread, P. J. Williams, G. Busico, A. C. Carter, A. K. Wood, N. Carr, J. C. Asplin, M. Q. Kearley, W. J. Hunt, D. R. Brambley, and J. R. Rawsthorne. A novel broadband DBR laser for DWDM networks with simplified quasi-digital wavelength selection. In *Optical Fiber Communications Technical Digest*, pages 541–543, March 2002.
- [55] Agility Communications. <http://www.agility.com>, 2001.
- [56] K. Shrikhande, I. M. White, M. S. Rogge, F-T. An, E. S. Hu, S. S-H. Yam, and L. G. Kazovsky. Performance demonstration of a fast-tunable transmitter and burst-mode packet receiver for HORNET. In *Optical Fiber Communications Technical Digest*, pages ThG2:1–ThG2:3, March 2001.
- [57] O. A. Lavrova, G. Rossi, and D. J. Blumenthal. Rapid tunable transmitter with large number of ITU channels accessible in less than 5 ns. In *Proceedings of the 26th European Conference on Optical Communications*, pages 23–24 (Paper 6.3.5, September 2000).
- [58] I. Chlamtac, A. Fumagalli, L. G. Kazovsky, P. Melman, W. H. Nelson, P. Poggiolini, M. Cerisola, A. N. M. Masum Choudhury, T. K. Fong, R. T. Hofmeister, C-L. Lu, A. Mekittikul, D. J. M. Sabido IX, C-J. Suh, and E. W. M. Wong. CORD: Contention resolution by delay lines. *IEEE Journal on Selected Areas in Communications*, 14(5):1014–1029, June 1996.
- [59] T. Saeki, M. Mitsuishi, H. Iwaki, and M. Tagishi. A 1.3-cycle lock time, non-PLL / DLL clock multiplier based on direct clock cycle interpolation for clock on demand. *IEEE Journal of Solid-State Circuits*, 35 (11): 1581-1590, November 2000.

- [60] C-K K. Yang, R. Farjad-Rad, and M.A. Horowitz. A 0.5- μm CMOS 4.0 Gbit/s serial link transceiver with data recovery using oversampling. *IEEE Journal of Solid-State Circuits*, 33(5):713-722, May 1998
- [61] A.K. Srivastava, Y. Sun, J.L. Zysking, and J. W. Sulhoff. EDFA transient response to channel loss in WDM transmission systems, *IEEE Photonics Technology Letters*, 9(3):386-388, March 1997.
- [62] M. Karasek, A. Bononi, L.A. Rusch, and M. Menif. Gain stabilization in gain clamped EDFA cascades fed by WDM burst-mode packet traffic. *Journal of Lightwave Technology*, 18(3):308-313, March 2000.
- [63] L. Tancevski, A. Bononi, and L.A. Rusch. Output power and SNR swings in cascades of EDFA's for circuite- and packet-switched optical networks. *Journal of Lightwave Technology*, 17(5):733-742, May 1999.
- [64] L. G. Kazovsky, K. Shrikhande, I. M. White, M. Rogge and D. Wonglumsom, "Optical Metropolitan Area Networks," Optical Fiber Communication conference, Anaheim, CA, pp. WU1-1, March, 2001 (Invited paper).
- [65] L. G. Kazovsky, I. M. White, K. Shrikhande and M. S. Rogge, "High Capacity Metropolitan Area Networks for the Next Generation Internet," Asilomar Conference on Signals and Systems, Monterey, CA, p. MA1b-1, November, 2001, (Invited paper).
- [66] I. M. White, M. S. Rogge, Y.-L. Hsueh, K. Shrikhande and L. G. Kazovsky, "Experimental demonstration of the HORNET survivable bi-directional ring architecture," Optical Fiber Communications Conference (OFC 2002), Anaheim, CA, p.WW1, March, 2002.
- [67] K. S. Kim and L. G. Kazovsky, "Design and performance evaluation of scheduling algorithms for unslotted CSMA/CA with backo_ MAC protocol in multiple-access WDM ring networks," JCIS 2002, Research Triangle Park, NC, USA, pp. 1303-1306, March, 2002.
- [68] K. S. Kim, H. Okagawa, K. Shrikhande and L. G. Kazovsky, "Unslotted Optical CSMA/CA MAC Protocol with Fairness Control in Metro WDM Ring Networks," GlobeCom 2002, Taipei, November, 2002.
- [69] I. M. White, M. S. Rogge, K. Shrikhande and L. G. Kazovsky, "Design of a control-channel-based media-access-control protocol for HORNET," *Journal of Optical Networking*, 1, pp. 460-473, December, 2002.
- [70] K. S. Kim and L. G. Kazovsky, "Design and performance evaluation of scheduling algorithms for unslotted CSMA/CA with backo_ MAC protocol in multiple-access WDM ring networks," *Information Sciences*, 149/1-2, pp. 135-148, January, 2003.

Submitted papers:

- [71] I. M. White, K. Shrikhande, M. S. Rogge and L. G. Kazovsky, "A MAC protocol with fairness control for the HORNET metro network architecture," submitted to Computer Networks journal, 2003.

- [72] I. M. White, K. Shrikhande, M. Rogge, and L. G. Kazovsky, "A Summary of the HORNET Project: A Next Generation Metropolitan Area Network," submitted to Journal of Selected Areas of Communications, special issue on Optical Networks, 2003.