2

Technical Report
985

Control and Reliability of Optical Networks in Multiprocessors

J.J. Olsen

22 December 1993

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

Prepared for the Department of the Air Force
under Contract F19628-90-C-0002.

This report is based on studies performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. The work was sponsored by the Department of the Air Force under Contract F19628-90-C-0002.

This report may be reproduced to satisfy needs of U.S. Government agencies.

The ESC Public Affairs Office has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Gary Tutungian

Gary Tutungian
Administrative Contracting Officer
Directorate of Contracted Support Management

Non-Lincoln Recipients

PLEASE DO NOT RETURN

Permission is given to destroy this document when it is no longer needed.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

CONTROL AND RELIABILITY OF OPTICAL NETWORKS IN MULTIPROCESSORS

J.J. OLSEN
Group 21

TECHNICAL REPORT 985

22 DECEMBER 1993

Approved for public release; distribution is unlimited.

LEXINGTON MASSACHUSETTS

ABSTRACT

Optical communication links have great potential to improve the performance of interconnection networks within large parallel multiprocessors, but the problems of semiconductor laser drive control and reliability inhibit their wide use. These problems have been solved in the telecommunications context, but the telecommunications solutions, based on a small number of links, are often too bulky, complex, power hungry, and expensive to be feasible for use in a multiprocessor network with thousands of optical links.

The main problems with the telecommunications approaches are that they are, by definition, designed for long-distance communication and therefore deal with communications links in isolation, instead of in an overall systems context. By taking a system-level approach to solving the laser reliability problem in a multiprocessor, and by exploiting the short-distance nature of the links, one can achieve small, simple, low-power, and inexpensive solutions, practical for implementation in the thousands of optical links that might be used in a multiprocessor.

Through modeling and experimentation, this report demonstrates that such system-level solutions exist and are feasible for use in a multiprocessor network. Semiconductor laser reliability problems are divided into two classes: transient errors and hard failures. Solutions to each type of problem are developed in the context of a large multiprocessor.

This report finds that for transient errors, the computer system would require a very low bit error rate (BER), such as 10^{-23}, if no provision were made for error control. Optical links cannot achieve such rates directly; a much more reasonable link-level BER (such as 10^{-7}) would be acceptable with simple error detection coding. This report then proposes a feedback system that will enable lasers to achieve error levels even when laser threshold current varies. Instead of telecommunications techniques that require laser output power monitors, this report describes a software-based feedback system using BER levels for laser drive control. A BER-based laser drive control experiment demonstrates that this method is feasible. It maintains a BER of 10^{-9}, which is much better than the error control coding system would need. The feedback system can also compensate for optical medium degradation and can help control hard failures by tracking laser wear-out trends.

For hard failures, a common telecommunications solution is to provide redundant spare optical links to replace failed ones. Unfortunately, this involves the inclusion of many extra, otherwise unneeded optical links, most of which will remain unused throughout the system lifetime. This report presents a new approach called "bandwidth fallback," which allows continued use of partially-failed channels while still accepting full-width data inputs. This provides, at a very small performance penalty, a high reliability level while needing no spare links at all.

In conclusion, the drive control and reliability problems of semiconductor lasers do not bar their use in large scale multiprocessors because inexpensive system-level solutions to them are possible.

ACKNOWLEDGMENTS

I would like to thank Anant Agarwal, my thesis supervisor, for his assistance and guidance in shaping my research program. I am grateful to my readers, Bob Kennedy and Tom Goblick, for their valuable advice. My thanks also go to the many other people at Lincoln Laboratory and the Laboratory for Computer Science who gave me advice and assistance. I would also like to thank my wife Batya for her support and encouragement in this work.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS
(Continued)

LIST OF ILLUSTRATIONS
(Continued)

LIST OF TABLES

1. OVERVIEW

For many years the promise of massively parallel computing (combining large numbers of inexpensive processors into powerful systems) has beckoned computer architects. It is apparent that the interconnection network between processors is one of the primary factors determining the overall performance of a multiprocessor system. The important parameters governing interconnection network cost and performance include connection density, bandwidth, and reliability.

Recent advances in optical technology suggest that optical interconnect networks will soon be viable for use in multiprocessors. Optical networks offer the potential of vastly increased bandwidth and connection density, compared with electrical networks.

Currently, the most widely used paradigm for optical data communication relies on semiconductor lasers as optical sources. Such lasers present a number of reliability problems. In the telecommunications context, where such optical communications find increasingly wide use, these problems have already been solved.

However, telecom solutions, based on a relatively small number of long-distance links, are often ill-suited to a multiprocessor network containing thousands of short-distance links. This report investigates reliability solutions that are more appropriate to the multiprocessor context.

1.1 The Reliability Problems

The use of optical communication networks in multiprocessors poses two main reliability questions.

1. Multiprocessor systems are exceedingly intolerant of data transmission error. How can the optical links achieve a sufficiently low data error rate?
2. Semiconductor laser failure data, when extrapolated to systems with large numbers of lasers, suggest that overall reliability of such a system might be unacceptably low. Does this mean that semiconductor lasers are impractical for use in large-scale multiprocessors?

Telecommunications links intended for telephone use are often designed to achieve bit-error-rates (BERs) on the order of 10^{-9} or 10^{-12}, with the best systems achieving values around 10^{-15} [1]. However, data communications within multiprocessors will require much better BER levels to assure reliable operation, such as 10^{-23} for example. (A 10^{-23} rate would be needed in a 4096-channel system with 64-bit 1-Gword/s channels, if one allowed 0.1 error/year.) Optical device research continues to seek lower BER levels, but the extremely low levels required on multiprocessor applications will almost certainly remain elusive on raw channels without error coding.

Another aspect of semiconductor laser operation is relevant to both transient-error and hard-failure performance: laser threshold current variation. As a laser ages, it wears out and eventually fails. This wear out manifests itself as an increase in laser threshold current [2], and the laser drive must provide sufficient current to compensate for the increased threshold. Threshold current also increases with laser temperature. Increased threshold, with fixed drive, will result in lower laser output and higher error rates. Also, high-speed lasers must be driven with a bias current approximately equal to the threshold, and an old (or hot) laser therefore "fails" if the drive circuit does not supply sufficient bias current to compensate for the increased threshold.

Semiconductor laser reliability is constantly improving, but it is still potentially a limiting item for system reliability. For example, a high-reliability laser might have a mean or median time between failure (MTBF) specification of 100,000 h. A 1024-node multiprocessor might employ 300,000 lasers; linear extrapolation of the failure rate would lead one to conclude that such a system would have an MTBF around 20 min: a completely unacceptable level of reliability. (Some researchers have gone so far as to conclude that this renders semiconductor lasers unsuitable for use in large-scale multiprocessor networks and have instead pursued other optical sources, such as light modulators pumped by a central, high-power gas laser [3].)

Laser failures can be divided into two broad categories: wear-out failures and random failures. While random failure times follow an exponential distribution, that is, they can be modeled as arrivals of a Poisson process, wear-out failure times follow a log-normal distribution [2]. Therefore the simple linear extrapolation used above:

$$\text{System MTBF} = \frac{\text{Component MTBF}}{\text{Number of Components}} \tag{1}$$

is valid only with respect to random failures, but not with respect to wear-out failures because a set of n identical Poisson processes of rate λ is equivalent to one Poisson process of rate $n\lambda$, but such a relation does not apply to a process with log-normally distributed arrival times.

Nevertheless, the true system MTBF will still likely be inadequate without special provision for recovery from laser failure. For example, even if there is a random-failure laser MTBF of 10,000,000 h, the system MTBF due to random failure in a 300,000 laser system will be about 30 h, which is still unsatisfactory. As the system ages, wear-out failures will also come into play, further increasing the failure rate.

1.2 Reliability Solutions

The solution to the laser reliability problem discussed in this report has three major parts:

- Control transient errors by error detection coding (EDC) and retransmission on error
- Control laser drive current with an intelligent software-based feedback loop based on the observed BER
- Control of hard failures with "bandwidth fallback," which provides the reliability benefits of large-scale redundant sparing, without the expense of providing spares

1.2.1 EDC

It is apparent that some form of error control coding scheme will be required to solve the BER problem. Error control coding strategies can be placed in two broad categories: forward error correction (also called error-correction coding), in which enough redundant data are included in each transmission to rederive the original data in spite of one or more transmission errors, and automatic retransmission request, in which only enough redundant information is sent to enable the detection of errors and the sender is asked to retransmit any data received in error.

Data storage systems, such as magnetic disks, are generally constrained to use forward error correction because it is impossible to ask for a retransmission of data if it has been corrupted on the storage medium. Similarly, high-speed telecommunication systems will use forward error correction because retransmission, while theoretically possible, would be impractical due to very long transmission delays (relative to bit transmission times).

By contrast, the sender and receiver in a multiprocessor system are relati -ly close and automatic retransmission request is quite feasible. [An exception to this would be multiprocessors that are constrained to operate in preset instruction sequence such as single-instruction, multiple-data (SIMD) systems because the times when retransmission is required can, of course, not be determined ahead of time.] A retransmission strategy, combined with a very simple EDC scheme, allows the use of data links with dramatically relaxed error rate requirements.

1.2.2 Intelligent Laser Drive Control

High-speed operation of most semiconductor lasers usually requires the adjustment of the laser drive current level as the laser threshold current varies with age and temperature. Design for long-term reliable operation of an optical network requires that the problem of laser drive level control be addressed.

The standard solution to this problem (in a telecommunications context) is to implement a laser light output power monitor, and to use this to form a feedback loop controlling the laser drive level (illustrated in Figure 24). This is an excellent approach for telecom applications, where the size, optical complexity, and expense of such a feedback loop is easily absorbed in an already large and expensive support system for each laser.

The multiprocessor context is different. When one considers the use of dozens of lasers on each channel, and many thousands of lasers in each system, the complexity and size of each laser link becomes much more important. Implementation of a laser light monitor system in a multiprocessor network involves an oppressive hardware overhead and would be much more difficult to justify than in the telecommunications context.

Fortunately, the coding approach described in the previous section, and the error-rate flexibility it allows, can be exploited to implement an alternative approach: intelligent (i.e., software-based) laser drive feedback control using the data-link error rate as the feedback control variable. The error rate is, in fact, the data-link parameter that ultimately needs to be controlled. Light-monitor-based approaches control the laser light output level for feedback only as an easily accessible and controllable (at link level) surrogate for the error rate.

Given the automatic retransmission request error-control system outlined above, one may implement an intelligent laser drive control system with the addition of one or two very simple digital-to-analog converters (DAC) per laser (or set of lasers), easily implemented *en masse* in an integrated circuit. The DAC are controlled by the sending processor, which adjusts the drive level based on how frequently its neighbors request retransmission of data.

1.2.3 Bandwidth Fallback

From the cursory analysis given in Section 1.1, it is clear that some provision must be made for continued system operation after laser failure occurs. Even if the intelligent drive control system could completely eliminate the wear-out problem (not a very likely prospect), random failures would still have to be handled. In telecommunications, a common approach to failure problems is to provide redundant spares to be used in place of failed links. Section 1 examines how the provision of redundant spare optical links could be implemented in a multiprocessor network and shows that this would be an adequate approach to the hard-failure problem if one could afford the cost of providing enough spare links.

Two approaches to eliminate the cost of spare links are examined: use of time-shared alternative paths to replace failed channels, and providing for variable-width data transmission to exploit whatever remaining data width is available (which is called "bandwidth fallback"). Bandwidth fallback is an idea for efficient use of the remaining bandwidth in a partially failed channel. With the addition of one multiplexer and one register per bit, the channel can be made usable at certain fractions of its full bandwidth, of the form 1/2, 3/4, 7/8, 15/16, etc. It can provide almost the same performance as large-scale redundant sparing, without needing any spares. (Bandwidth fallback is also applicable to electrical networks, given that off-board connections are generally much less reliable than on-board or on-chip connections.)

1.3 Summary of Results

As long as one avoids the type of inflexible system design that precludes the use of automatic retransmission request error control [such as SIMD design], an optical multiprocessor network need only implement coding for error detection, as opposed to both detection and correction. This report demonstrates that a very simple linear EDC can achieve the needed level of error control with minimal overhead: relaxing the optical-link BER requirement from 10^{-23} to 10^{-7}.

Given such a coding scheme, laser drive control can be accomplished without the need for light output power monitors and the associated analog feedback loops by using the link's BER to control the laser drive level. A software-based laser drive control system has been designed and implemented for an experimental free-space optical data link. Experiments have been conducted on the drive control system over varying temperatures, showing excellent control of error rate.

(Interestingly, the laser output level is controlled quite well without the use of a light level monitor.) This report shows that a 5-bit DAC would be adequate for level control in the experimental setup and that the experirental feedback system is stable with a gain margin of at least 10 dB. Additional experiments demonstrate recovery from optical medium degradation: a capability not possessed by traditional monitor-based control schemes. This report also explains how an intelligent laser drive control system can help deal with laser wear-out failures by facilitating long-term monitoring of laser wear-out trends.

This report finds that providing redundant spare lasers (if one could afford enough of them) would result in acceptable system reliability. This report also shows that, with only a small performance penalty, the implementation of alternative routing and bandwidth fallback can provide the same reliability benefits without the spares. Alternatively, they can provide the same performance as redundant sparing, with a great reduction in the number of spares needed.

The final section offers suggestions for further research and concludes that the laser reliability problem can, in fact, be solved by proper use of system-level reliability solutions.

2. WHY OPTICS?

Why would one want to use optical interconnect in a multiprocessor? Studies of multiprocessor optical interconnect have cited many advantages of optical over electronic interconnect [3,4,5,6]. The major advantages are outlined below.

2.1 Connection Density

From a network theory perspective, the most interesting advantage of optical interconnect is its potential for increased connection density. This is due to the contrasting natures of the electron and the photon.

Electrons are fermions and carry a charge; they therefore interact strongly with each other, both due to electromagnetic effects and due to the Pauli exclusion principle. Photons are bosons and electrically neutral; they interact weakly, if at all. Because of this, multiple light beams can cross the same point in space without interfering, while multiple wires cannot. This property of optical interconnect may render some previous analyses of multiprocessor networks, which implicitly assumed connection density limits, inapplicable.

2.2 Bandwidth

One of the most immediate benefits of optical interconnect is increased bandwidth. While electrical network interconnects have been pushed to 1 Gbit/s [7], they become increasingly cumbersome as frequency components progress higher into the microwave spectrum. Optical links now in use for telecommunications achieve 3.4 Gbit/s data rates over distances of 50 km [8]. Laser modulation rates of 15 GHz have been reported [9].

Fundamentally, an optical link is a modem (modulator-demodulator), with an optical carrier signal. Phenomenal bandwidth potential is not surprising because the carrier being modulated typically has a frequency of 200 THz[1] or more. The bandwidth available on an optical link is not limited by the link itself, rather by the electronic circuits on either end. Progress in optoelectronic integrated circuits (OEICs) suggests that the highest-speed electronic signals associated with the link will need to extend no further than a single chip (or chip module). This implies that serialization (multiplexing) of the electrical signals will be needed in order to exploit the full bandwidth of the optical link.

Data rates in conventional electronic signaling (that is, baseband signaling) are limited by dispersion effects, due to the very broad bandwidth relative to the center frequency. Optical telecommunications links also suffer from dispersion, but over the relatively short distances encountered within multiprocessor interconnection networks ($\ll$ 1 km), dispersion is very small.

[1] At $\lambda = 1.5\mu$m: the longest wavelength currently in wide use.

2.3 Fan Out

Electronic logic signals have traditionally supported high fan-out (that is, multidrop) connections. Conventional bus architectures (for example, VMEbus or NuBus) rely on electronic fan-out to broadcast signals among all the boards on the bus.

Unfortunately, the multidrop model breaks down at high switching speed, due to transmission line reflections. As Knight points out [10], while it is theoretically possible to adjust and tune the transmission line to produce a matched multidrop line, in practice it is so cumbersome as to be impractical. Practical high-speed ($\geq$ 1 Gbit/s) electrical interconnect will generally be point-to-point.

Optical interconnect poses no such restriction. Multidrop connections are perfectly feasible, limited only by the total transmitted power available. A fundamental question (to be answered by further research) is how best to exploit this capability or whether it is worth exploiting.

2.4 Electromagnetic Interference (EMI) Immunity

Unlike electrical signals, optical signals are virtually immune to EMI and similar effects, such as ground-loop noise. This is an important practical advantage as anyone who has constructed large-scale high-speed computing systems can attest. Although quite valuable, this EMI immunity does not appear to have a direct impact on the choice of a particular computer architecture. Instead it makes implementation of any large-scale system more practical.

2.5 Communication Energy

Power dissipation is an important factor in all high-speed computing systems and a large part of the power is dissipated in communication, not in calculation [11]. Miller [12] argues that, fundamentally, optical communication requires less energy than does electrical communication, over all but the shortest distances.

Miller's argument is based on the characteristic impedance of free space, $Z_0 = 377$ ohms. All practical transmission lines will have a characteristic impedance much less than Z_0 because the impedance depends only logarithmically on the line's dimensions. (At one conference [13], Miller put it: "All transmission lines are 50 ohms." A bit of hyperbole, but not far from the truth.) This is illustrated by the impedance formula for a vacuum-filled coaxial line: $Z = Z_0 \frac{1}{2\pi} \ln(r_2/r_1)$. Even to get the impedance up to Z_0, the outer conductor would have to be 500 times larger than the inner conductor and any dielectric material would make it even more difficult.

Small electronic devices can pass only small currents and are therefore high-impedance devices poorly matched to driving transmission lines. Traditionally, large buffer amplifiers (pad drivers) are placed between the logic elements on a chip and the transmission lines off-chip. Because the transmission line must be properly terminated (somewhere), the low-impedance drivers use up considerable power in charging and discharging the line. Low-impedance drive is also the source of current-switching transient (dI/dt) noise: a major design constraint.

Optical signaling avoids the free-space impedance problem because it operates on quantum mechanical (photons↔electrons) rather than classical physical principals. Photodiodes are already well matched to logic inputs, and multiple-quantum-well techniques may produce low-threshold lasers and/or optical modulators that can be driven directly by microelectronic gates.

It now seems that optical signaling is the best approach for long-distance communications, and electronic signaling is the best approach for communication over microscopic distances. Ross [8] (probably relying on Miller's work [12]), offers a graph like the one in Figure 1 to argue that optical communication is preferable to electrical communication over distances longer than some critical length. The exact value of this critical length depends on the specific parameters of the links being compared, but Ross puts it at 200μm. (This is based on fundamental energy considerations. If, instead, one used present-day economic considerations, one might find the the critical length to be closer to 200 cm, or even 200 m.)

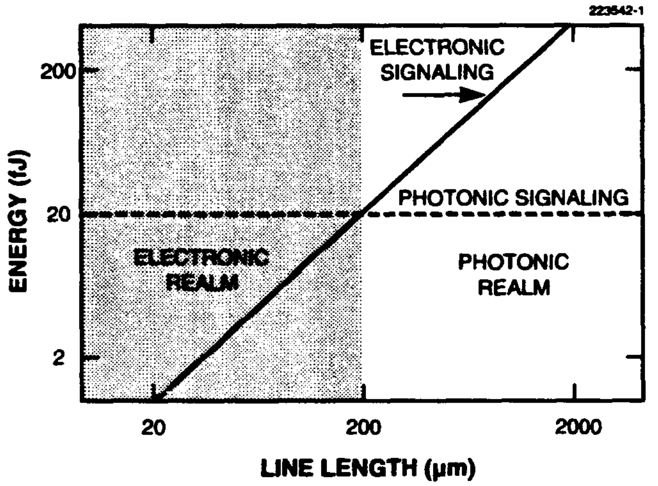

Figure 1. Communication energy vs distance (based on Ross and Miller [8,12]).

If energy is the ultimate limitation, it seems that optics may eventually supplant electronics for all interconnect above the intrachip level. It does so now for communications over long enough distances, and the criterion for "long enough" will certainly grow shorter and shorter as optical technology progresses.

3. MULTIPROCESSOR OPTICAL NETWORKS

Optical interconnect is an active research field, and the state of the art is advancing rapidly. This section describes the state of the art as it applies to optical networks for multiprocessors.

3.1 Data Communication vs Telecommunication

Driven by the telecommunications market, optical communication hardware is now highly developed and continues to improve. Telecom hardware now achieves phenomenal performance, such as fibers with 0.1 dB/km attenuation, and 3.4 Gbit/s links with 50-km distances between repeaters [8]. Current research topics such as soliton-based signaling [14] promise even more impressive results.

Unfortunately, this superb telecommunications hardware is not directly applicable to multiprocessor networks because it is designed to satisfy different constraints. In telecom, long-distance performance is the key because this governs the number of repeaters required on a link, and repeaters form a major part of the cost of a link. Also, telecom size and power constraints (per link) are often less stringent than the corresponding constraints for computer hardware.

Because of this, telecom optical links tend to make the cost-performance trade-off in favor of high performance and high cost per link: too high a cost for use in parallel processor networks, which need many, many links. (Many links are needed in order to increase bandwidth without increasing electrical switching speed and to reduce the latency inherent in serializing the data for transmission. Such latency is no problem in telecom systems because it is overwhelmed by the speed-of-light transmission delay.)

Optical datacom (that is, short-distance communication) hardware is well behind the leading edge of performance. For example, the fiber distributed data interface (FDDI) local-area network [15] is only now coming into wide use, and it uses a signaling rate of 125 Mb/s: over 25 times slower than the fastest telecom links.

Given the differing cost constraints, it seems likely that datacom hardware performance will remain well behind the state of the art in telecom. However, this state of the art is advancing so rapidly that optical datacom hardware can be expected to achieve current telecom performance within a few years.

3.2 Data Communication Hardware

While there are many possible implementations of a fixed optical communication link, such implementations share some common features. A link consists of an optical source, an optical receiver, and a transmission medium.

3.3 Optical Sources

The two major optical sources for communication links are light-emitting diodes (LEDs) and semiconductor lasers. Another option is the use of optical modulators acting on an externally generated laser beam.

3.3.1 LEDs

LEDs are very simple devices and have a number of advantages for use in an interconnect network: they are cheap, easy to fabricate, and reliable. In some moderate-performance applications, LEDs are the perfect choice. For example, FDDI [16] uses them as optical sources.

Unfortunately, the LED has limited bandwidth. LEDs are quite capable of signaling rates around 100 Mb/s (as in FDDI) and with some effort can be made to switch at 1 GHz, but at higher switching rates they become increasingly impractical. For >1 Gbit/s rates, lasers or optical modulators appear to be the most reasonable sources.

3.3.2 Semiconductor Lasers

The most straightforward laser light source is the semiconductor laser with its drive current modulated by the desired data stream. Modulation of semiconductor lasers at 15 GHz has been demonstrated, and it seems safe to say that the laser switching rate will likely be limited by the drive electronics before it is limited by the laser itself. Microfabrication of surface-emitting semiconductor laser arrays is showing impressive results, such as densities of 200,000,000 lasers/cm^2 [8], and threshold currents around 1 mA (potentially much lower [17]).

Semiconductor lasers have a number of problems, but probably the most difficult one for this application is laser reliability. Laser reliability figures are hard to come by and depend on several factors, but the current state of the art is on the order of 10^5 or 10^6 hours MTBF. For a massively parallel processor with thousands of lasers, laser failure will be a fairly common event and must be handled gracefully. A semiconductor-laser-based system will likely have to cope with the laser reliability problem by providing considerable fault tolerance in the higher-level design.

3.3.3 Optical Modulators

To avoid the semiconductor laser reliability problem, an alternative approach using optical modulators has been proposed [3]. In this scheme, large, high-power external lasers are used to provide an optical "power supply" that is routed to the inputs of optical modulators. The external laser can be made quite reliable and wavelength-stable. The modulator outputs produce a modulated light beam similar to that otherwise produced by the semiconductor laser. The modulators and lasers have broadly similar characteristics; in fact, multiple-quantum-well modulator arrays (see Section 3.6.1) are almost identical to multiple-quantum-well laser arrays, omitting the laser mirror on one side.

Modulator-based schemes have their own problems, chief among them the need for optical power distribution. However, a number of groups (notably Honeywell) are convinced of their superiority, especially in systems subject to wide temperature variations, such as military systems.

The modulator vs laser question is still an open one, with no consensus yet apparent. A significant factor in deciding the issue will be the extent to which it is practical to handle the laser reliability problem with inexpensive, system-level solutions.

3.4 Optical Receivers

Optical receivers are relatively straightforward. The optical signal illuminates a diode, causing a photoelectric current. This is then amplified to the desired logic levels (or used directly [18]).

Either PIN (*p*—intrinsic—*n*) or avalanche diodes may be used. The avalanche diode produces a stronger signal, but is more difficult to fabricate. Either diode may be fabricated in silicon, but at the high data rates of interest here, III-V materials such as GaAs are also of interest.

This report assumes the use of suitable photodiodes, with high-speed amplifiers on the same chip or module, if needed.

3.5 Transmission Media

Optical transmission media are another important concern in optical network research because the medium partly determines the topological and other constraints on the optical network.

3.5.1 Free Space

The simplest medium is free space. The pure free-space link (a light source shining directly on a receiver) is finding applications now, over short distances such as those in interboard communication. While pure free-space links are interesting, some other free-space approaches are potentially more useful in computer networks. A number of these alternatives involve free-space links with some reflective or refractive device between source and receiver.

Free-space communication schemes (and holographic schemes, to some extent) offer the opportunity to exploit the ability, mentioned in Section 2.1, of optical signals to cross in the same point in space. This is a significant advantage, but it is not without cost. All free-space approaches require extremely accurate mechanical alignment, posing problems for manufacturing and, more particularly, for repair. (Realignment of a replacement component in the field may be quite difficult.) Some form of adaptive alignment would be a significant advantage.

Considerable research has dealt with the use of compound optics as a means of distributing light from source to receiver (see Figure 2). This generally imposes a constraint of "space-invariance" on the transmission pattern: the output pattern (relative to the input beam) must be the same from all ports.

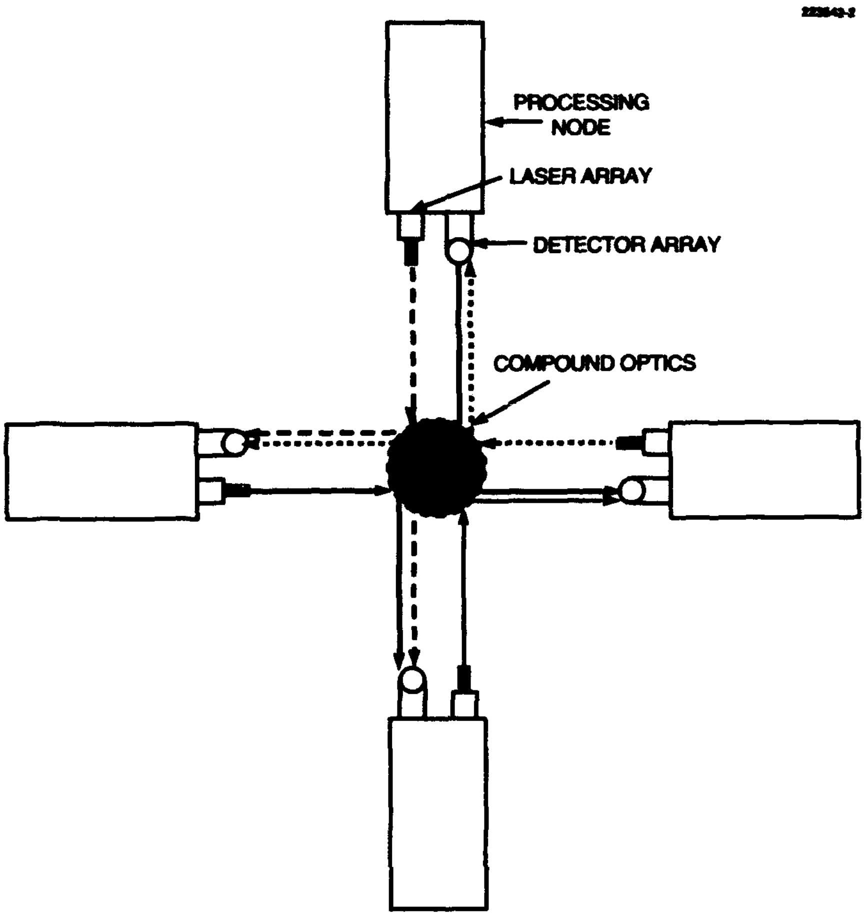

Figure 2. Free-space communication with compound optics.

3.5.2 Holographic Networks

Holographic interconnect is another form of free-space link, with the optical distribution device being a hologram [19] (see Figure 3). Holograms do not impose a space-invariance constraint. The pattern reflected by the hologram is basically arbitrary, within certain angular constraints. These angular constraints can be significantly relaxed by employing multilayer holograms (that is, several holograms sandwiched together).

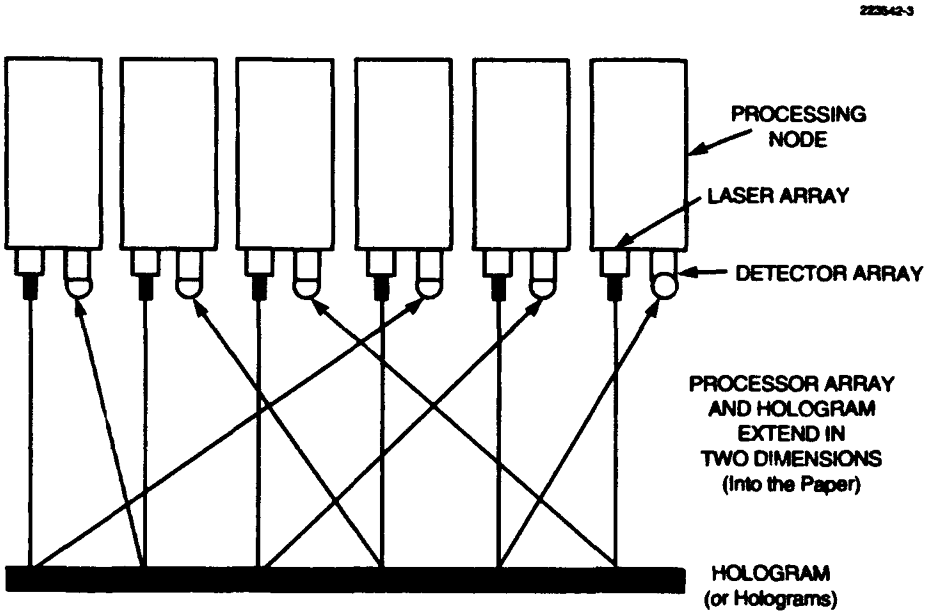

Figure 3. Free-space communication with holograms.

3.5.3 Planar Waveguides

Oversimplifying, one might say that a planar waveguide [20] is a printed-circuit board for optical signals (see Figure 4). It offers connections between arbitrary points in a plane, with crossovers, combiners, and splitters available. Being self-contained, it obviates most of the alignment problems of free-space approaches such as holograms. Unfortunately, their transmission efficiency is still quite low, currently around 0.3 dB/cm. (Compare this to 0.1 dB/km for high-quality fiber.)

Planar waveguides offer moderate connectivity with moderate implementation difficulty. They appear to be a possible compromise between point-to-point fiber connections (easily implemented,

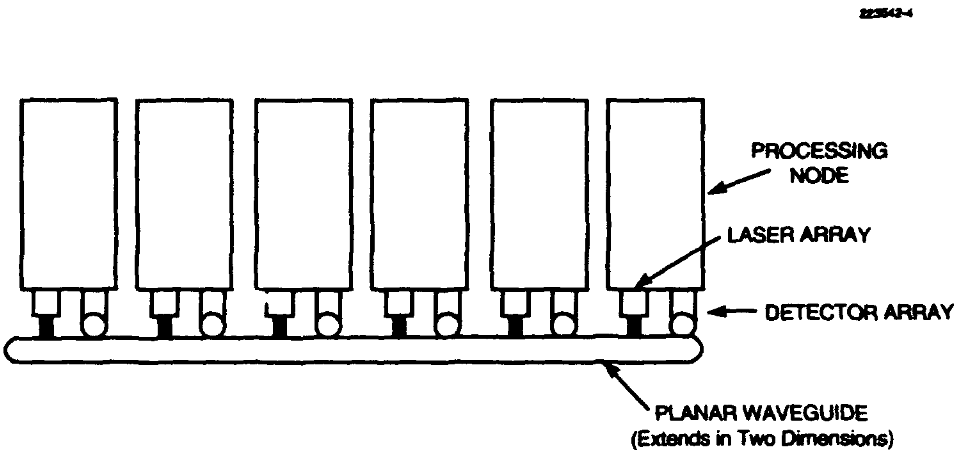

Figure 4. Communication via planar waveguide.

but low connectivity) and holograms (difficult implementation problems, but very dense connectivity). Lin [21] suggests an interesting combined hologram/waveguide approach.

3.5.4 Fiber Optics

Optical fiber is by far the most highly developed optical medium today. Optical fiber is now produced, which imposes only a few decibels of loss over a distance of hundreds of kilometers. Unfortunately, such long-distance performance is of little relevance to multiprocessor networks, where distances will be less than 10 m.

The basic disadvantage of fiber is that it is implemented on an individual link basis, one fiber at a time. Progress is being made on the use of fiber bundles: parallel groups of fibers terminated together [22], but it is still premature at this time to consider a large-scale multiprocessor with interconnections solely based on fiber bundles.

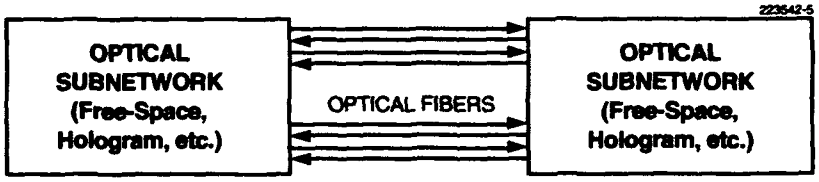
223542-5
OPTICAL SUBNETWORK (Free-Space, Hologram, etc.)
OPTICAL FIBERS
OPTICAL SUBNETWORK (Free-Space, Hologram, etc.)

Figure 5. Subnetwork interconnection via optical fiber.

However, fiber does have one important advantage: flexible geometry. All of the previously-mentioned optical media require the communicating nodes to be aligned in a predetermined geometry and fixed in that alignment. Optical fibers, on the other hand, offer flexibility in two senses: mechanical flexibility, allowing communication between points that are not in rigid alignment, and design flexibility, allowing arbitrary placement of the two endpoints of the fiber.

Fiber bundles might therefore be of considerable usefulness in connecting optical subnetworks that were implemented by other, less flexible means. The subnetworks could be made as large as practical, and then used to create a still larger network via fiber connections. Figure 5 illustrates this idea.

3.5.5 Assumptions of the Analysis

Because the laser reliability questions are substantially the same for all the optical media presented here, the analysis need not assume the use of any particular medium. However, the analysis of transient errors in Section 4.1 is particularly important for the less power-efficient media, such as planar waveguides, because the importance of the power vs error-rate trade-off is magnified in those cases.

3.6 Optical Switching and Computing

A potentially strong motivation for the use of optical networks is optical switching and computing. There is no intrinsic distinction between optical switching and optical computing: any reasonably efficient switching system can be made to do computation. However, while "optical switching" means what it says, the term "optical computing" has had the special connotation of *purely* optical computing. [To confuse matters further, the meaning of "optical computing," at least in some quarters, seems to be evolving toward "optical communication between electronic logic devices." This is the sort of thing the report examines.]

3.6.1 Optical Switching

Optical switching has been studied for some time, and optical switches for telecom have been well developed. However, switches for massively parallel systems must be extremely fast and compact. Presently available switches are generally interferometer designs (such as Mach-Zehnder), which tend to be large, or liquid-crystal designs, which tend to be slow. Current research into newer designs, such as Bell Laboratories' work on multiple-quantum-well (MQW) devices [17], is quite interesting for this application.

MQW switches might serve as the modulators for the external-laser design mentioned in Section 3.3.3. The external-laser design tends to blur the dichotomy between optical communication and optical switching.

3.6.2 Optical Computing

All of the electrooptical approaches to computing are highly mismatched in speed with the electrical portion as the speed-limiting factor. If true optical computing could be achieved, it might improve computer performance by orders of magnitude.

True optical gates have been demonstrated [23], but they involve high power levels and are not fully restoring. Pure optical gates are far from practical computing implementation for the time being and seemingly for the near future as well.

Reflecting this discouraging prospect, the term "optical computing" is apparently evolving toward a different meaning: electronic computing with purely optical interconnection. (In a sense, this is what the purely optical gates do as well, because their optical nonlinearity is fundamentally based on electronic interactions.) Bell Laboratories researchers vigorously assert that their self-electrooptic devices (SEEDs) [24] implement "optical computing."

SEEDs and other "optical computing" devices in the new sense of the term, while remarkable devices, are fundamentally electronic logic gates with optical input and output, and therefore limited by electronic switching speeds.

3.6.3 Assumptions of the Analysis

While high-speed optical *communication* hardware is practical now and is in wide use, high-speed (that is, high reconfiguration speed) optical *switching* hardware is less advanced, and optical *computing* hardware (in the original sense) will not be practical for some time, if ever. Therefore, the analysis in this report treats the reliability of optical communication only. Analysis of optical switching and computing is an area for further research.

3.7 Multiprocessor Networks

Multiprocessor networks have been a fertile subject of research for over two decades and continue to be so. A vast literature is available on various interconnection schemes. However, no consensus has been reached as to which approaches are optimal, even in the extensively studied electronic regime, much less in the optical regime.

Nevertheless, some interesting mathematical results on performance limits of various networks have been achieved (for example, Dally's work on mesh networks [25]). Many of these analyses assume that network bisection bandwidth is the limiting factor. The assumption may be true for electronic interconnect (although even this is disputed), but is unlikely to be valid for optical networks. On the other hand, the analysis of Agarwal [26] would seem to be applicable to optical networks, in the case where he assumes a limit on node size instead of network bandwidth.

3.7.1 Mesh Networks

Mesh networks, also called grid or nearest-neighbor networks, possess an elegant simplicity. The basic idea is to place nodes in a plane (or a volume) in a regular pattern and form a communication network by linking each node with its nearest neighbors. Their short communication lengths and point-to-point links make mesh networks especially well-suited to electronic implementation, negating a number of the advantages mentioned in Section 2.

An optical mesh network may still be worthwhile. As noted in Section 3.5.1, point-to-point free-space interconnect is well advanced and is now close to practical implementation. This could be used to implement an optical mesh network. The existing research into electronic mesh networks should be directly applicable with the optical links looking like very-high-speed wires.

If the optical mesh network is not used, a very interesting possibility is a dual network combining a nonmesh optical network with an electronic mesh. The mesh network is so well suited to electronic implementation that the addition of an electronic mesh network might provide a significant performance gain for a small extra cost. This idea is illustrated in Figure 6.

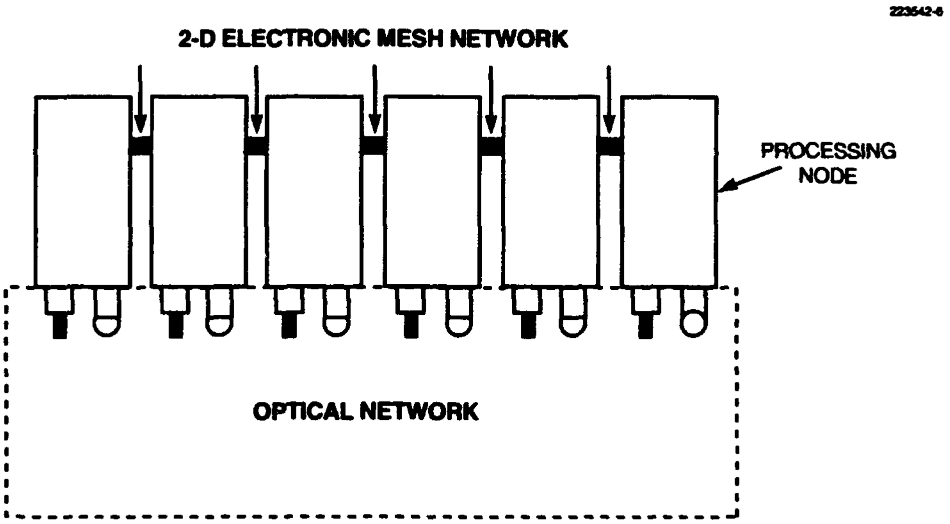

Figure 6. Dual electronic/photonic interconnection network.

3.7.2 Hierarchical Networks

In the current literature on multiprocessor networks, there is considerable work in hierarchical networks. These are large, complex networks consisting of a hierarchy of smaller, simple networks.

Hierarchies both of busses [27] and of crossbars [28] have been studied. They offer opportunities to exploit the ability of optics to support high fan-out connections, discussed in Section 2.3.

3.7.3 Butterfly-Type Networks

Butterfly networks (omega networks) are in wide use for multiprocessor interconnect, notably in the Bolt, Beranek, and Newman (BBN) Butterfly multiprocessor [29]. The multibutterfly, a variation on the butterfly network, is another possibility for an optical network. When the connections of a multibutterfly are randomized under certain constraints, it can achieve impressive fault-tolerant performance, as discussed by Leighton [30]. The randomized connection stage might be particularly well suited to holographic implementation.

3.7.4 Assumptions of the Analysis

While the various network topologies offer fascinating opportunities for development of new multiprocessor architectures, the optical reliability questions are fundamentally similar across all of them. For the reliability analyses in Sections 6, 8, and 9, this report assumes the use of the simplest topology: the 2-D mesh.

4. OPTICAL DATA LINK RELIABILITY PROBLEMS

Because Section 3.6.3 discusses optical technology for communication only, this report therefore defines optical data link problems simply as events that cause the data received from the link to differ from the data originally transmitted. Such events can be divided into two classes: *errors* of short duration (transient errors) and *failures* of much longer duration (hard failures).

The dividing line between these two classes is the amount of time required for the system to recognize, locate, and verify the problem. The exact division is not critical because most of these events are either very short (on the order of one bit transmission time) or permanent.

The goal of this section is to characterize the behavior of both transient errors and hard failures and to develop mathematical models describing them. Sections 6 to 9 use these models in developing and evaluating solutions to the reliability problems.

4.1 Transient Errors

No method of data transmission is perfect; there is always a chance that a given transmission has been corrupted by transient errors.

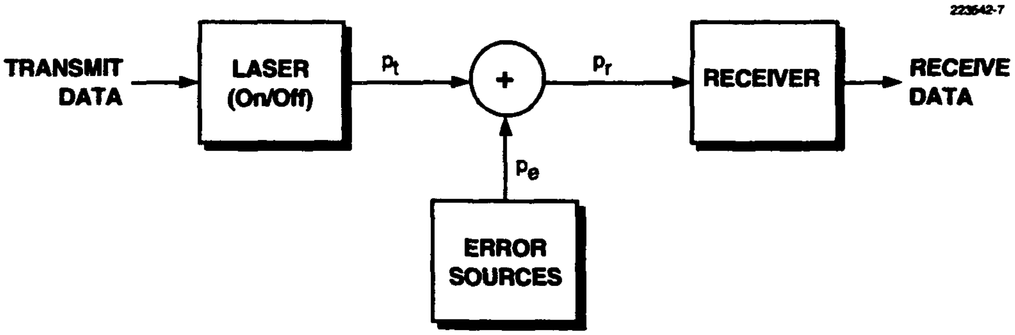

Figure 7. Optical communication error model.

The simplest model of such errors is that of two-level modulation (such as "on" and "off") and a Gaussian additive noise source, which will occasionally cause the receiver to report an incorrect value [31]. Figure 7 shows an example of such a system.

A laser, modulated by an incoming data stream, produces a transmitted optical signal p_t at one or two intensity levels: 0 (off) or L (on). All the sources of error are lumped into one error signal p_e, a zero-mean Gaussian signal with standard deviation σ. To simplify the analysis, $\sigma = \sigma^2 = 1$

is set. This implies that the signal-to-noise ratio (SNR)= L. The receiver compares the received signal $p_r = p_t + p_e$ with the threshold $L/2$, and reports "on" if $p_r < L/2$ and "off" if $p_r > L/2$.

When $p_t = 0$, the probability that a bit will be received in error is the probability that $p_e > L/2$, which is

$$\text{P(error)} = \text{BER} = \frac{1}{\sqrt{2\pi}} \int_{L/2}^{\infty} e^{-\frac{1}{2}p^2} dp = \frac{1}{2}\text{erfc}\left(\frac{L}{2\sqrt{2}}\right) \tag{2}$$

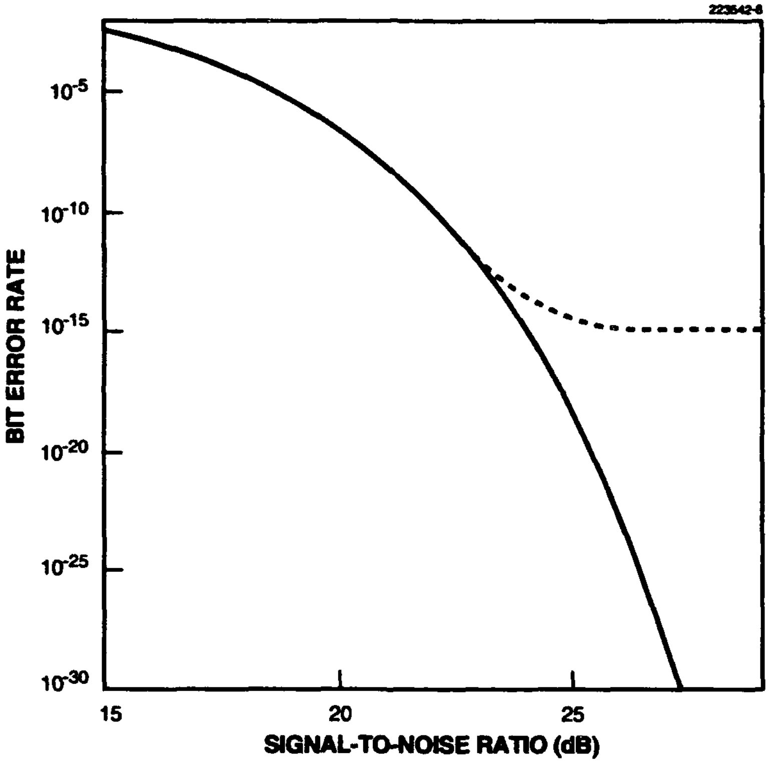

Figure 8. Error rate vs signal-to-noise ratio.

By symmetry, the same analysis applies when $p_t = L$. Figure 8 shows a plot of BER values given by Equation (2).

Note, however, that this idealized analysis will eventually break down at some point. When the power level is sufficiently high, the model developed above breaks down and other error sources become apparent, causing an "error floor" in the BER vs SNR curve. The dashed line in Figure 8 is an illustration of such a floor.

In an optical link (especially with a power-inefficient transmission medium), the trade-off between optical power and BER (lower power → higher BER) can be quite important. The optical sources require extremely fast electrical drivers, and the higher the power required of the drivers, the more difficult their design becomes. Also, there is a power vs reliability trade-off in semiconductor lasers (lower power → higher reliability).

[The power↔BER trade-off will also become important in Section 7, when this report discusses intelligent control of laser drive power. In that case, this report relies on the monotonic nature of this trade-off to implement a power-control feedback loop with data link BER as the feedback variable.]

Generally, one needs extremely low error rates in a multiprocessor network because an undetected error can be catastrophic. Section 6.1 derives a maximum tolerable BER of 10^{-23} for a prototype network. The best achievable BER values now are around 10^{-15}[1] and even these levels are only possible with painstaking care in designing every part of the optical channel.

How then can one tolerate an optical network in a multiprocessor? Section 6, a simple error-control coding scheme solves this dilemma quite neatly, provide that a usable ($< 10^{-7}$, for example) BER can be maintained. (Section 6 gives a strategy for maintaining such error rates as the system ages.) In addition to making optical links usable in multiprocessors, error-control coding can also make their design and implementation much simpler and more cost-effective by relaxing the otherwise stringent error-rate requirements.

4.1.1 Assumptions of the Analysis

The most fundamental assumption in the analysis of reliability solutions is that errors in different bit positions are independent. Because this report suggests a system with wide optical channels, having separate optical channels for each bit, the assumption is quite plausible to a first order approximation.

This report will rely on this independence assumption in the error-control analysis. Any significant error correlation between channels (for example, EMI between channels at the receiver module) might compromise the error-control results. (On the other hand, such correlation might simplify the drive-control tasks addressed in Section 6 because one might only have to control the drive level on a per-module basis instead of a per-laser basis.)

Because such error problems are likely to be implementation-specific, this report will merely note the possibility of such problems and conduct subsequent analyses under the independent-error assumption.

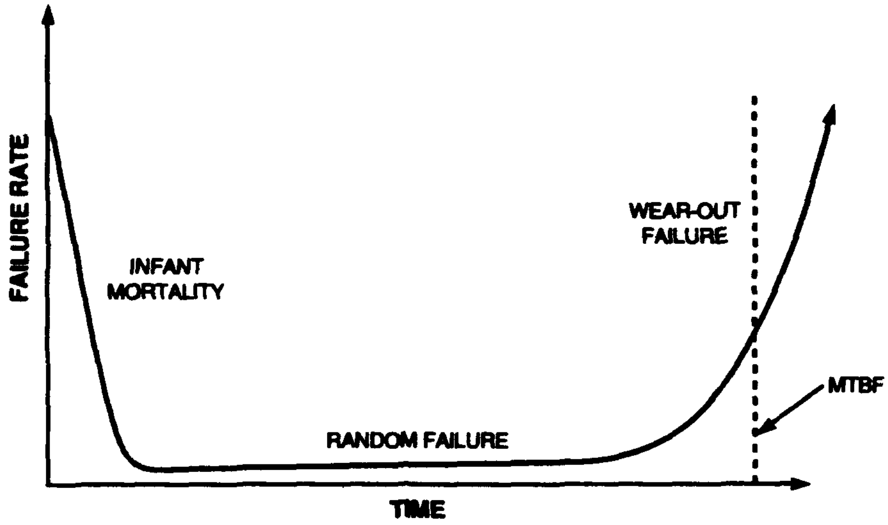

Figure 9. Component failure rate vs time.

4.2 Hard Failure

All electronic components degrade with age. Figure 9 shows the typical relationship (called the "bathtub curve") between age and failure rate for any electronic component. Note that the entire curve is customarily characterized by one number: MTBF, an ambiguous acronym that can mean either mean or median time between failures. This report generally considers MTBF to refer to the median time and makes it clear when the other sense is meant and the distinction is important. (In this case, MTTF, mean/median time to fail, is more appropriate, but this report uses the more common term.)

Initially, there is a high failure rate due to flawed components (infant mortality). There follows a long period of few, randomly-distributed failures (random failure). Finally there is an increasing rate again (wear-out failure), as the component aging processes begin to take their toll.

Infant mortality can be controlled by proper screening and burn-in procedures [32] before the system begins operation. Random and wear-out failures, however, must be handled in the field, either by fault-tolerant redundancy schemes, or by repair.

Unfortunately, semiconductor lasers tend to degrade faster than most other components in an optical communication network, so their failures will tend to dominate the failure characteristics of the system. This report examines the nature of semiconductor laser failure, considering wear-out failures in this section and random failures in Section 4.2.2.

4.2.1 Gradual Wear out

There are numerous aging processes that lead to wear-out failure in semiconductor lasers [2]; they can collectively be modeled by a log-normal lifetime distribution, where log(lifetime) is a Gaussian random variable. If t_ℓ denotes the wear-out lifetime of an individual laser, then the probability distribution of t_ℓ will be given by

$$p_{t_\ell}(t) = \frac{1}{\sqrt{2\pi}} e^{\frac{-\ln^2(t/t_M)}{2\sigma^2}} \tag{3}$$

where t_M is the median lifetime, and σ is the standard deviation. Figure 10 shows the shape of the log-normal distribution for various values of σ.

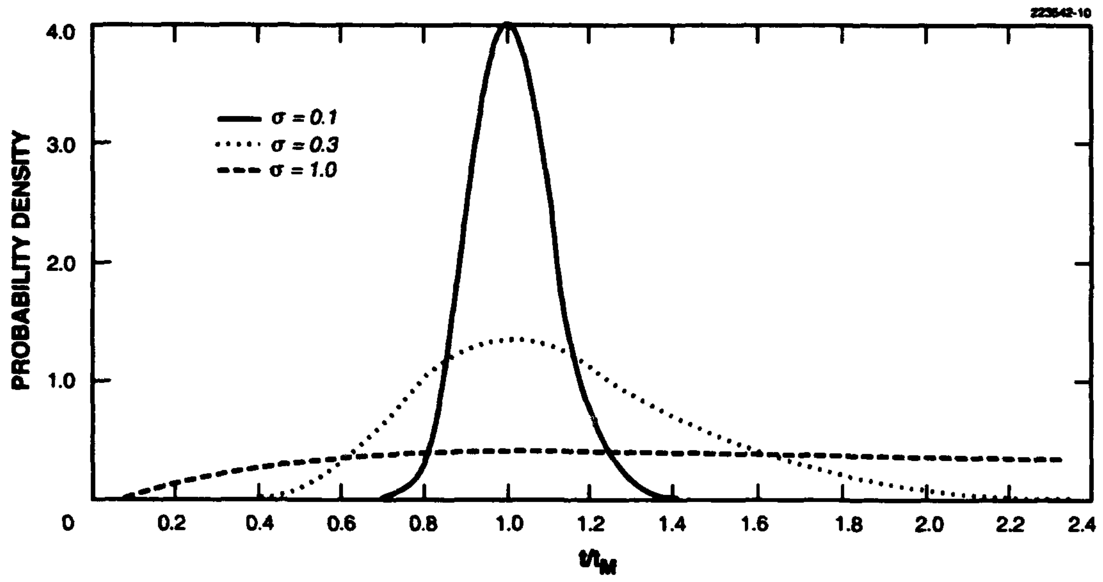

Figure 10. Log-normal probability distributions.

[Note that this is one of the situations where the mean vs median ambiguity in acronyms such as MTBF can be misleading. For larger values of σ, the distribution is skewed toward larger values of t_ℓ, and the mean lifetime will be considerably larger than the median. For example, with $\sigma = 0.4$, the mean lifetime is 144% of the median.]

An important aspect of Figure 10 is that all of the curves are skewed away from time $t = 0$. This is why, for multiple lasers, the simplistic relation in Equation (1) does not apply. What relation *does* apply to wear-out failures in multiple lasers? This question will be answered in Section 8, when system-level solutions to the failure problem are considered.

Laser wear out vs time. Taken as a whole, the lifetimes of lasers in a system are random and follow the probability density given in Equation (3). Taken individually, each laser has its own deterministic wear-out lifetime. When this time comes, how does the wear out manifest itself?

Wear-out failures are not instantaneous; they occur quite gradually. As a laser wears out, its light output gradually decreases for the same drive current. (Or, conversely, the drive current required to maintain the same light output slowly increases). Laser MTBF figures are obtained by setting an arbitrary wear-out criterion such as a 50% output power decrease or a 100% drive current increase and declaring a laser "failed" when this criterion is reached. However, the gradual degradation continues considerably beyond this point, until finally the laser stops lasing. Figure 11 illustrates this concept in the case of constant laser drive current.

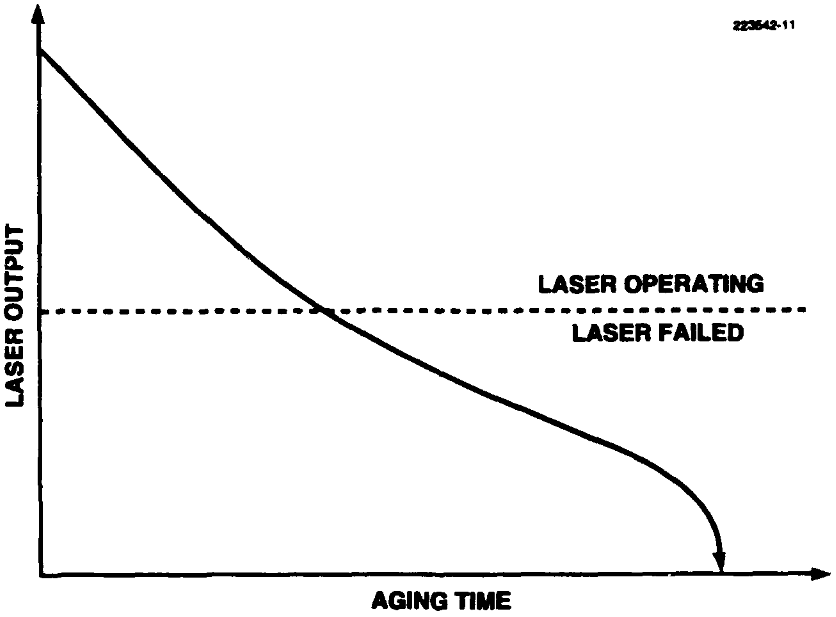

Figure 11. Laser wear out vs time.

The light output of a semiconductor laser is a linear function of the drive current (I_D) in excess of the threshold current (I_{th}) for that laser, that is

$$\text{Laser light output} \propto I_D - I_{th} \tag{4}$$

The wear-out processes generally manifest themselves as a gradual increase in threshold current. Using $I_{th}(t)$ to denote threshold current as a function of time, this gradual increase can be modeled as

$$\frac{I_{th}(t) - I_{th}(0)}{I_{th}(0)} \propto t^n$$

where n is a constant characteristic of the particular laser type used ($n \approx 0.5$ [33]).

Letting α denote the drive current increase used for calculating the laser lifetime t_ℓ (for example, a 50% increase means $\alpha = 0.5$), and letting I_0 denote the initial drive current used,

$$I_{th}(t) = I_{th}(0) + I_0\alpha(t/t_\ell)^n \tag{5}$$

Therefore, with a constant drive current, the laser light output will gradually diminish. If the drive current is adjusted for constant light output, the current required will gradually increase.

If the available laser drive current limit is the same as that used for calculating the wear-out lifetime statistics (for example, 50% increase from initial drive current), then the standard lifetime figures will be directly applicable. However, if the available laser drive is different, the actual wear-out lifetime will vary accordingly.

This gradual degradation process presents both a problem and an opportunity. The problem is that a laser's performance will begin to decay even before its nominal wear-out lifetime arrives; therefore, additional power margins will have to be included in the link design to allow for acceptable operation even with somewhat aged lasers. The opportunity to be exploited is that the increasing drive current or decreasing output power will give quite ample warning that a laser is wearing out; this could enable the repair, replacement, or readjustment of failing lasers to be performed at conveniently long intervals. Section 7.5.2 considers the system implications of this.

4.2.2 Random Failure

Random failure in semiconductor lasers is rather simple compared to the wear-out failure discussed in the previous section. Random failure is the failure type shown in the center section of the bathtub curve, Figure 9. Random failures occur due to problems such as bond-wire breakage or die attachment failure; they can be modeled as arrivals of a Poisson process.

If a large number of semiconductor lasers were to be operated continuously until failure, relatively few of the lasers would fall victim to random failure, because the random failure MTBF is generally much longer than the wear-out MTBF. However, unlike wear-out failures, random failures occur at the same rate throughout the system lifetime, and therefore *do* follow the simple relation given in Equation (1).

Also unlike wear-out failures, random failures are generally abrupt, clear-cut failures as opposed to gradual fade-outs. This means that random failures, as the name implies, can occur unpredictably, without warning, at any time. Therefore, system design must provide for handling of random failures whenever the system is large enough to reduce, via Equation (1), the large random failure MTBF down to an unacceptably short system MTBF. Sections 8 and 9 will examine such designs.

4.3 Hard Failure Probability

The total failure rate will be the sum of the random failure rate and the gradual wear-out rate. While the random failure rate is more or less fixed, the wear-out rate will grow over time. This report analyzes both types of failure in the next two sections and then arrives at a composite failure model for use in simulations.

4.3.1 Gradual Wear-out Probability

As previously mentioned, the gradual wear-out failure rate is not constant, but instead increases with time.

The probability that a particular laser has worn out by time t (assuming no random failures) is derivable from Equation (3), the wear-out lifetime probability distribution. If $P_w(t)$ denotes this probability,

$$\begin{aligned} P_w(t) &= \int_0^t p_{t_\ell}(\tau) d\tau \\ &= \frac{1}{\sqrt{2\pi}} \int_0^t e^{\frac{-\ln^2(\tau/T_w)}{2\sigma^2}} d\tau \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\ln(t/T_w)} e^{\frac{-\tau}{2\sigma^2}} d\tau \\ &= \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{\ln(t/T_w)}{\sqrt{2}\sigma}\right)\right) \end{aligned} \tag{6}$$

where T_w denotes the median laser wear-out lifetime and σ is the standard deviation of the log-lifetime $\ln(t_\ell)$.

Figure 12 plots $P_w(t/T_w)$ for different values of σ. Note that for small σ, wear-out failure does not become significant until close to the laser lifetime (about $0.75T_w$ for $\sigma = 0.1$), and is virtually complete a short time afterward (about $1.25T_w$). On the other hand, with large σ ($\sigma = 1$, for example), wear-out failure starts around $0.25T_w$ and continues at a relatively constant rate through $t = 2T_w$.

Obviously, the log-lifetime deviation σ is an important parameter in the laser wear-out function. In the limit $\sigma \rightarrow 0$, every laser is preordained to wear out at exactly $t = T_w$ (obviously convenient for replacement scheduling). In the limit $\sigma \rightarrow \infty$, half the lasers fail immediately and the other half last forever. Between these (unreasonable) extremes, a reasonable real-world range of σ values is between 0.1 and 1.

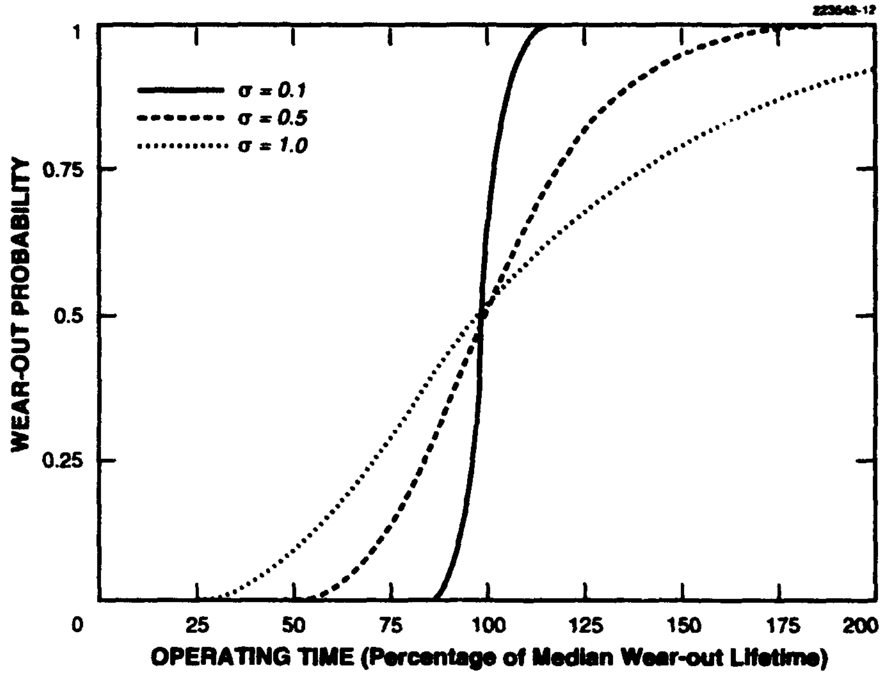

Figure 12. Laser wear-out probability $P_w(t/T_w)$.

Unfortunately, σ is often not well characterized for a particular laser design. For the moment, leave σ as an unspecified system parameter and move on to consideration of random failures.

4.3.2 Random Failure Probability

As mentioned in Section 4.2.2, random failure is a rather simpler phenomenon and can be modeled as a Poisson process. Analogous to $P_w(t)$, define $P_r(t)$ as the probability that a particular laser has suffered a random failure by time t (assuming no wear-out failures). If T_r denotes the mean time before random failure, the Poisson arrival rate will be $\lambda = 1/T_r$ and

$$P_r(t) = 1 - e^{\lambda t} = 1 - e^{-t/T_r} \tag{7}$$

Figure 13 shows $P_t(t/T_r)$, which is, of course, a simple exponential function. Unlike wear-out failure, which is characterized by both a median lifetime and a deviation, random failure is fully characterized by its arrival rate λ or its mean interarrival time T_r.

Consider the overall failure function, which is the combination of wear out and random failures.

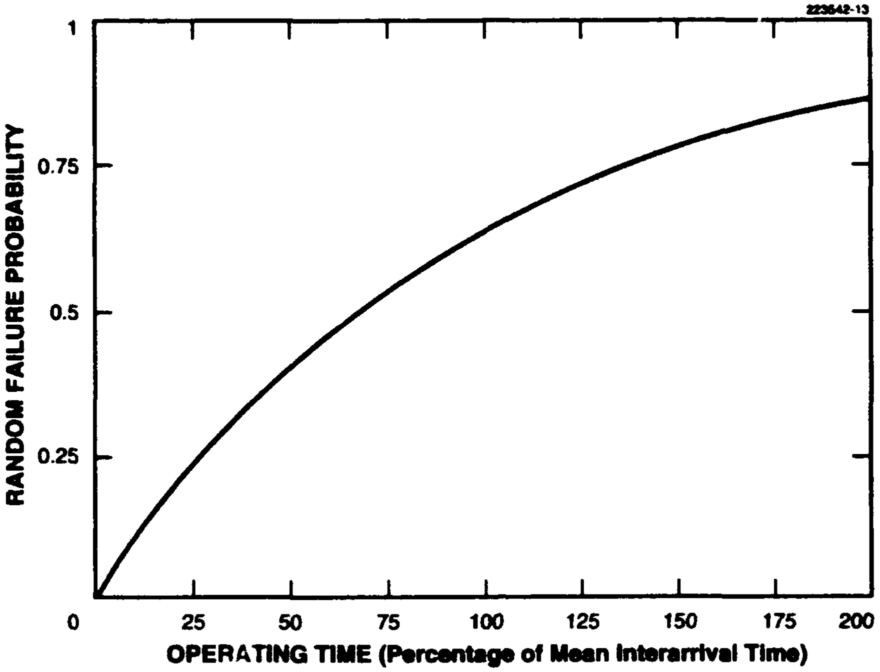

Figure 13. Random failure probability $P_r(t/T_r)$.

4.3.3 Combined Failure Probability

To arrive at the combined error probability (random and wear-out failures), this report assumes that the two types of failure occur independently. This seems a reasonable assumption because there are different failure mechanisms involved in the two types of failure.

If $P(t)$ denotes the probability that a particular laser has failed by time t (either random or wear-out failure),

$$\begin{aligned} P(t) &= P_r(t) + (1 - P_r(t))P_w(t) \\ &= 1 - e^{-t/T_r} + \frac{1}{2} e^{-t/T_r} \left(1 + \operatorname{erf}\left(\frac{\ln(t/T_w)}{\sqrt{2}\sigma}\right)\right) \\ &= 1 + \frac{1}{2} e^{-t/T_r} \left(\operatorname{erf}\left(\frac{\ln(t/T_w)}{\sqrt{2}\sigma}\right) - 1\right) \qquad (8) \end{aligned}$$

Let us define an additional parameter as $\gamma = T_w/T_r = T_w\lambda$, so that the value of γ is a measure of the relative importance of random failures vs wear-out failures. If $\gamma = 0$ then there are no random failures, if $\gamma = \infty$ then there are *only* random failures, and if $\gamma = 1$ then random and wear-out failures are equally important.

Laser reliability is most commonly specified with only one number: MTBF, comprehending both wear-out and random failures. Let us then define T as MTBF, that is, the median time before laser failure, be it wear-out or random failure. Then there are three parameters to fully characterize the laser failure distribution: T, σ, γ.

The lifetimes T_w and T_r are now functions of T, σ, and γ, such that $T_r = T_w/\gamma$ and $P(T) = 0.5$. Because T is the value establishing our time scale, $T \propto T_w = \gamma T_r$.

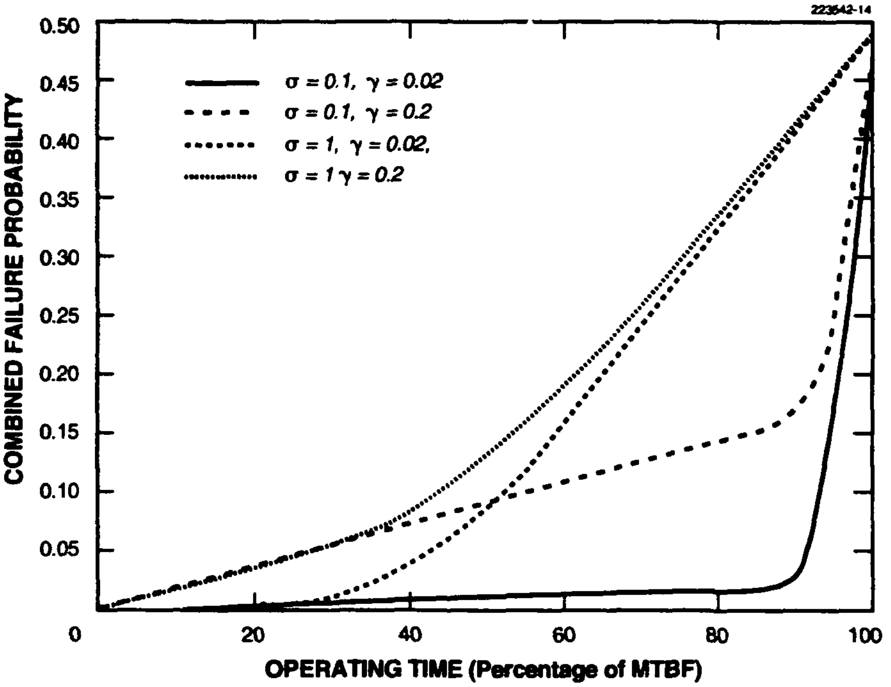

Figure 14. Combined failure probability $P(t/MTBF)$.

A closed-form solution for T_w and T_r (and therefore $P(t)$) as a function of T, σ, and γ is difficult to arrive at, but numerical solutions can easily be found. Figure 14 plots $P(t)$ for reasonable values of σ and γ. Note the importance of γ (the random failure parameter) in the early part of the lifetime axis. This demonstrates the discussion in Section 4.2.2: random failure exerts its influence earlier than does wear-out failure.

4.3.4 Probability Model for Analysis

The network failure analyses that will be described in Sections 8 and 9 are actually oblivious to the particular failure models and parameters one might choose to apply. The analyses and simulations work with particular values of the failure probability $P(t)$ and use the failure models only to relate these probabilities to times. Therefore, although this report will set out some failure model assumptions here, the analysis and simulation results will be applicable to other failure models, with some adjustment to the system-life timeline.

For these purposes, this report assumes that the models are using high-reliability lasers, of 100,000-h MTBF. Also this report assumes a random-failure MTBF of 1,000,000 h, and therefore $\gamma = 0.1$. As Fukuda points out [2], the random-failure rate is a strong function of the manufacturer's experience with the fabrication processes in use: as processes become established and well-known, the random-failure rate can be reduced dramatically. Of course, new processes can offer great performance advantages, at the cost of somewhat less reliability, so there is an engineering trade-off here. The value $\gamma = 0.1$ is the best guess of the highest reasonable value for γ, which would place the most stress on the network's fault tolerance features.

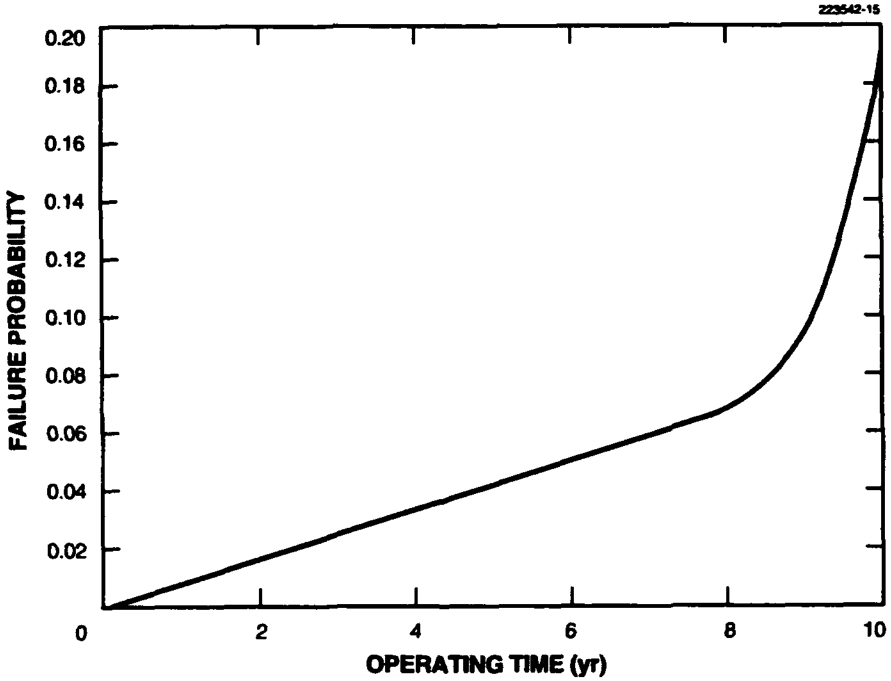

Figure 15. Simulation failure probability model.

This report assumes a log-lifetime deviation $\sigma = 0.25$. Figure 15 shows the $P(t)$ function resulting from the laser lifetime assumptions. A value of 0.25 is reasonable for σ, but it could be considerably worse (perhaps $\sigma = 1$) for some lasers. In that case, the laser wear out would start around $t = 4$ years, and reach $P(t) = 0.2$ at $t = 7$ years. One should keep this in mind when looking at the analysis and simulation results in Sections 8 and 9: a high value of σ for the lasers in use would result in a higher rate of failure starting around $t = 4$ years.

4.3.5 Assumptions of the Analysis

As with transient errors, this report assumes that hard failures in different bits are statistically independent. Unlike the transient-error assumption, this assumption is almost certainly false.

This report assumes the use of arrays of lasers as components of very wide optical data channels. There will almost certainly be significant correlations among the lifetimes of lasers within the same array.

Modeling and simulating such correlations would complicate the simulations and analyses considerably. To make the problem more tractable, this report assumes independent failures to arrive at a baseline result. Further research could refine this result to include correlated-failure effects.

5. PROTOTYPE NETWORK

For the analyses and computer simulations of optical network reliability solutions, this report will posit a particular multiprocessor network design, the reliability problems of which will hopefully be representative of those a real design would encounter.

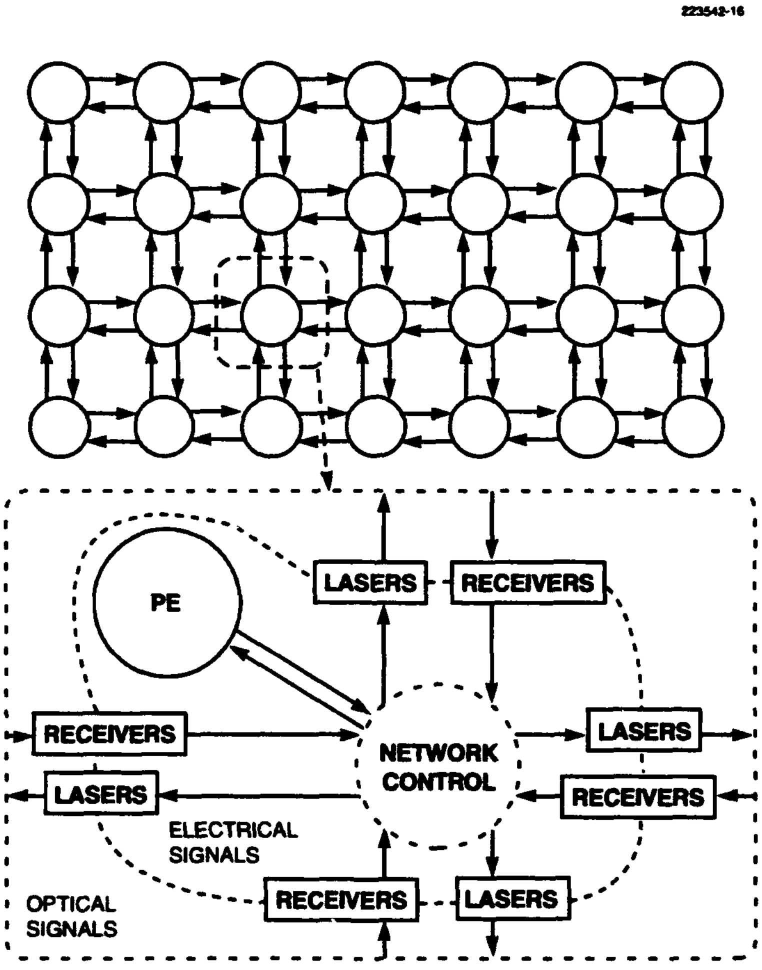

Figure 16. Prototype multiprocessor network.

The prototype network is a 2-D mesh-network multiprocessor shown in Figure 16. All signaling within a node is done electrically, and all signaling between nodes is done optically.

The mesh network is interesting and has been the subject of much study; an optically-implemented mesh network might have some significant advantages over an electrically-implemented one. However, that is not the reason for choosing the mesh for analysis. Rather, the mesh, while simple to analyze, is complex enough to involve the same type of reliability problems and solutions as do the other, more complex topologies discussed in Section 3.7.

5.1 Optical Channels

Each of the node-to-node connections shown in Figure 16 is a bidirectional pair of optical channels, each composed of a number of one-bit optical links. As mentioned in Section 3.2, each optical link consists of an optical source, an optical receiver, and a transmission medium. The link design for the prototype network is illustrated in Figure 17.

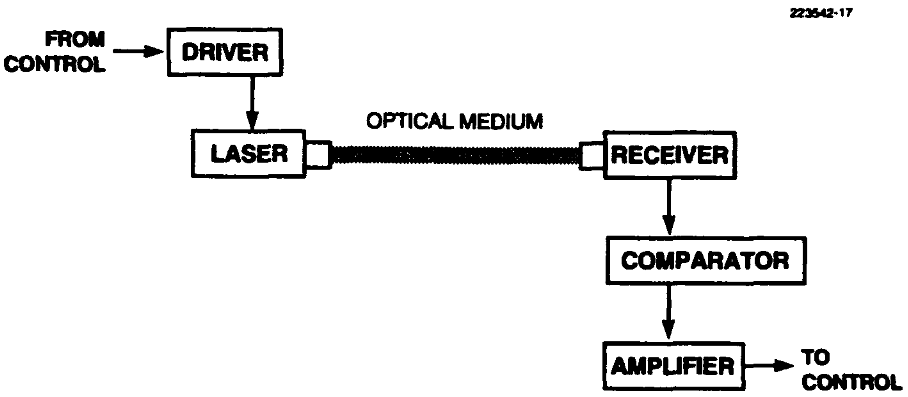

Figure 17. Prototype optical link.

Let us first consider the optical transmission medium. As observed in Section 3.5.5, one need not make hard-and-fast assumptions about the medium to be used. Any intraprocessor links will be at most a few meters long (the distance from one cabinet to another in the same room), and probably on the order of centimeters, or even millimeters. Therefore the dispersion problems that dominate telecom optical-fiber design are completely irrelevant to the design.

The speed of light, however, is a constraint that is not so readily overcome, so it is necessary to discuss the distances involved in order to get an estimate of the transmission delay. In free space, every 30 cm traveled introduces 1 ns of delay; in other media there is even more delay that is proportional to the square root of the dielectric constant of the medium. For the analysis, this report assumes a reasonably compact system, such that transmission from one node to the next occurs within 1 ns.

Next, let us consider the data transmission rate. As noted in Section 3.1, while optical links for telecommunication are now running at multi-Gbit/s data rates, datacom systems are running at data rates in the low hundreds of Mbit/s. For the prototype network analysis, this report assumes that this situation has progressed enough to allow a signaling rate of 1 Gbit/s for each laser/receiver link. (This, with the distance constraint, means that each optical transmission will be completed within one bit time.)

Next, this report considers the optical channel width, that is, the number of parallel optical links in each channel. Present-day optical datacom systems are dominated by single-bit (that is, one laser and one optical receiver) approaches, because multiple arrays of lasers and optical receivers, while under intensive research and development, are not yet commercially feasible. For the prototype network, this report anticipates the availability of laser and optical-receiver arrays, and assumes that each of the optical channels carries multiple bits. For the quantitative analyses and simulations, this report assumes that each channel carries 64 data bits (plus overhead for error control, signaling, spares, etc.), because 64-bit processors are likely to dominate the high-performance market in the near future and for some time to come.

Next, let us consider the lasers and receivers. This report will not assume any particular laser design, but rather assume different laser reliability characteristics and examine their impact on system design and reliability. For the optical receiver, this report assumes a simple photodiode design with appropriate decision circuitry on the output.

Next, this report considers the optical modulation and reception methods. Because this will be a massively parallel system, where cost-per-link will be a vital concern, high-performance but technologically challenging approaches such as phase modulation and coherent detection will not be considered. (Soliton transmission, of course, is another challenging approach that is of vital interest to telecom but is completely irrelevant to the short-distance datacom links considered here, where dispersion is negligible.) This report therefore assumes simple on-off modulation of the laser and incoherent detection at the receiver.

As Tsang demonstrates [18], the driver, amplifier, and comparator sections in Figure 17 needn't be complex. His data link experiments have demonstrated 1 Gbit/s transmission using a standard emitter-coupled logic (ECL) gate for the driver, and an ECL flip-flop for both amplifier and comparator. Such circuitry could readily be integrated for use with multiple arrays of lasers and receivers, along w:th error control and other circuitry to be discussed in Sections 6 and 9. (The interesting topic of whether to have a constant laser drive level, or whether to control the drive level to maintain a constant output level, is discussed in Section 7.)

To summarize, the prototype network uses optical channels for internode communication, transmitting 64 data bits in parallel in both directions (128 data bits in all), plus overhead bits, every nanosecond. Each link is a straightforward on-off modulated laser, passing through a short optical medium to a photodiode, with each bit arriving within 1 ns.

5.2 Network Nodes

At each node of the prototype mesh network, there is a processing element (PE), that is, a computer processor or processors, memory, and associated circuitry. The question as to how much processing power to place in each node (and therefore having a greater or lesser number of nodes) is a fascinating one, but not directly relevant to the reliability concerns.

For this discussion, the internal workings of a PE are not considered: it is considered to be an independent, random generator of network traffic. Each PE, at randomly determined times, generates a data packet of randomly determined size for transmission to another PE in the network. The PE gives the packet to the data network for transmission and tells the network to which other PE the data is destined.

Also at each node is a network-control switching circuit that

- Accepts new packets from the PE, codes them for error control, and sends them toward their destination node
- Receives packets destined for this node, checks their error-control code, and gives them to the PE
- Receives other packets, and forwards them toward their their destination node
- (Optionally) reconfigures the network to bypass failed optical links

Extensive analyses of such networks are available elsewhere [25,34]. Section 5.3 gives a brief outline of such an analysis. (The error-control coding and network reconfiguration will be discussed in Sections 6 and 9, respectively.)

The PE expects the network control hardware to route and deliver the data without further help from the PE, *in most cases.* However, unlike the telecommunications system scenario, the data-source vs data-network boundary can be blurred whenever it is advantageous. Conceptually, the PE controls every aspect of the data network, and the network control logic merely helps it handle the common cases quickly.

Of course, the rate of data transfer is so fast that almost all network operations must in fact be handled by the network control logic, but the "processor-centric" concept above expressed is valuable when considering how to handle important but comparatively rare events, such as transmission errors or laser failures. Intervention by the processor can provide a powerful, intelligent response to such occurrences, with negligible effect on total system performance because it happens so infrequently. (Note that this concept is merely a more general application of the ideas applied to the cache coherence problem in the Alewife multiprocessor [35].)

5.3 Prototype Network Operation

The mesh network, while providing direct connectivity only between a node and its four nearest neighbors, must be able to provide communication between any two nodes in the network.

Obviously, when a message must be passed between two nodes that are not adjacent to each other, it must be forwarded through the intervening nodes. Figure 18 shows an example of this, with a packet originating in node "O," forwarded by several nodes "F," and arriving at its destination node "D."

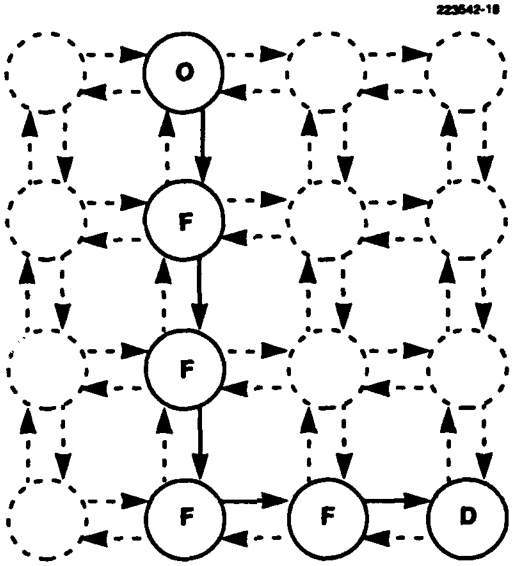

Figure 18. Mesh routing example.

The traditional approach in such situations is a "store and forward" network, where each intervening node receives and stores the message before forwarding it on its way. Unfortunately, the store-and-forward approach has a number of problems, such as a potentially unbounded amount of storage required for messages in-transit, given certain patterns of message traffic. Of more direct importance in the present case is the delay incurred in receiving and buffering the entire message before retransmitting it.

If the transit time from one node to the next is t_t, the time to transmit the entire packet is t_p, and the packet must be transmitted or retransmitted (forwarded) N_t times, then the time required to store-and-forward the packet to its destination will be $N_t(t_t + t_p)$.

For the prototype simulations, this report uses a different method: *wormhole routing* [36]. Rather than store the entirety of each packet in each node, it is immediately sent out as soon as received, as long as the next link in the packet's path is available. (This means that a long message will spread out, like a worm slithering across the network, hence the name.) If the next link is not available, the message's progress is blocked until it becomes available, and therefore each node needs to provide only a small amount of buffering. With wormhole routing, the message delivery time is reduced to $N_t t_t + t_p$, a considerable speedup for all but the smallest packets.

In such a forwarding network, it is unfortunately very easy to run into a deadlock situation, where different data transfers interact such that each is waiting for the other to complete, with the result that none of them can ever be completed. Figure 19 illustrates such a situation. Each of the four nodes is trying to send a two-hop message, has taken the first hop already, and cannot take the second hop until the other transfers are completed, which can never happen.

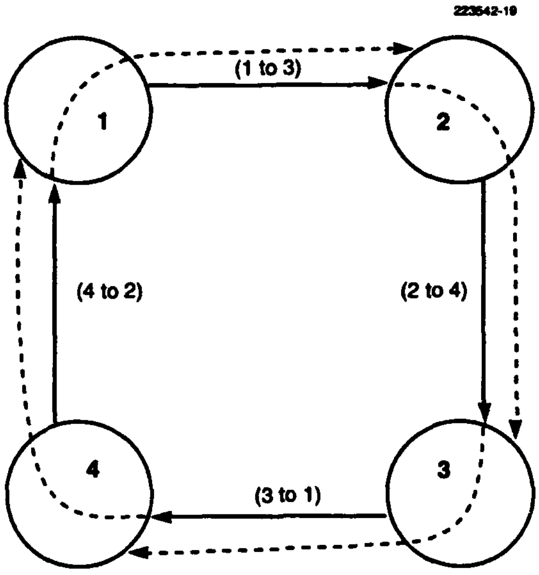

Figure 19. Mesh routing deadlock.

Fortunately, there is an elegant solution to the deadlock problem in mesh networks [37]. (Now, more sophisticated solutions are also available [25,38].) If one merely ensures that all packets are routed in the same dimensional order (for example, move in the X-direction first, then in the Y-direction). This makes it impossible to form the cycle of dependencies that is required to produce a deadlock. (In Figure 19, node 1 is blocked by node 2, which is blocked by node 3, which is blocked by node 4, which is blocked by node 1, forming the deadlock cycle. This is possible because nodes 1 and 3 are routing their packets in the order X-then-Y, while nodes 2 and 4 are routing Y-then-X.)

This report assumes the use of wormhole routing in the prototype mesh network. This implies that the beginning of each packet contains an address, indicating the destination node and that each channel has at least one reverse-direction bit for flow control, indicating whether further data can be accepted, or whether the packet must wait for a needed link to become free.

6. ERROR CONTROL CODING

This section considers and evaluates solutions to the transient-error reliability problems described in Section 4.1, given the systems context outlined in Section 5. The evaluation will be in the context of a prototype multiprocessor network.

This section considers the best way to provide error-control coding as a solution to the problem of transient errors. Error correcting codes (EEC) are compared with simple EDC, and one can conclude that the latter are fully sufficient and much more cost-effective for this application.

6.1 Acceptable Error Levels

When considering the solution of a reliability problem, one must answer the question, "How reliable a system does one need?" (or the related question "How reliable a system is one willing to pay for?"). This section establishes the sort of reliability expectations that will be assumed for transient errors in conducting analyses and simulations. A similar discussion of hard-failure expectations will be found in Section 8.1.

As mentioned in Section 4, a multiprocessor network will tolerate only a very low undetected BER, because one bit in error might crash the multiprocessor program, or worse, introduce a subtle error into the results. However, this stringent criterion only applies to *undetected* errors: a packet that suffers a transient error, when detected, can simply be discarded and retransmitted. The only impact of detected errors is the performance reduction from having to perform the retransmission.

For the analysis, this report assumes that the optical network system has a lifetime of ten years (3×10^8 s) and arbitrarily declares that an average of one undetected transient error during that time is acceptable. For a 1024-node system, with each node having four outgoing channels, each channel having 74 links (64 data + 10 overhead), and each link running at 1 Gbit/s, there is an expected number of errors, over ten years, of

$$\mathrm{BER} \cdot 1024 \cdot 4 \cdot 74 \cdot 10^9 \cdot 3 \times 10^8 = 10^{23} \cdot \mathrm{BER}$$

Undetected BER (that is, rate of bit errors that are not detected) of about 10^{-23} or less is therefore required to meet the transient-error-rate criterion.

6.2 Coding Theory

The goal in the area of transient error control is to relax the error-performance requirements on the individual optical links. If one could engineer cost-effective links with BER of 10^{-23}, the problem would disappear. Unfortunately, achieving such a BER value is impossible, and approaching it is costly, both in money and in loss of design flexibility.

Fortunately, as in many other applications, coding theory can come to the rescue. An appropriate error-control coding system can theoretically convert an abysmally poor channel into a reliable one. Of course, such a coding scheme is not guaranteed to be simple to design or practical to implement.

6.2.1 EDC vs ECC

The first major decision is whether to use an ECC or merely an EDC with automatic retransmission request (ARQ) for messages received in error. In a multiprocessor network, there will be four major constraints on an error-control coding strategy:

1. Coding/decoding logic complexity
2. Transmission latency
3. Bandwidth overhead[2]
4. Undetected/miscorrected error rate

On balance, based on these criteria, an ECC strategy is less desirable than EDC scheme.

Coding complexity is higher for ECC than for EDC, as can trivially be seen from the fact that error-detection is a subset of error-correction: an error-correction system must provide facilities to detect errors in order to know whether a corrected value is needed. In addition, it must implement a system to calculate the error-corrected value, and the error-correction section of such a system is always considerably more complex than its error-detecting counterpart.

Transmission latency is an important consideration in the choice of ECC vs EDC. When a request for retransmission would take a 10^9 bit-transmission times to make the round-trip between receiver and transmitter (as, for example, in a satellite telecommunication application), the ARQ strategy becomes unworkable at the data link level. However, in a multiprocessor network the transit times are trivially small by comparison, and an EDC ARQ strategy is viable.

The coding systems work by ensuring that all possible code words are different from each other by a specified amount or distance. The standard metric of code word difference is the *Hamming Distance*, which is just the number of bit positions that differ between two code words. The important metric for consideration here is the *minimum distance*, which is denoted by d_m, of a given code, which is the minimum Hamming distance between any two code words that the code might generate. When the receiver sees a code word that the code could not have generated, it signals that an error has been detected, and either the correction circuitry (under ECC) or a retransmission (under EDC) should provide the correct value.

[2] That is, how much of the channel is needed for transmitting extra check bits, as opposed to transmitting the actual data.

An EDC that will detect e_d errors must have $d_m \geq e_d + 1$. If d_m were less than or equal to e_d, then a set of e_d errors could transform one valid code word into another, creating an undetectable error at the receiver.

An ECC system contains error detection as a subset. In addition, its correction circuitry, given an invalid code word c_r from the receiver, chooses the valid code word c_c that is closest (in Hamming distance) to c_r. In this case, a system to correct e_c errors will require $d_m \geq 2e_c + 1$, because with $d_m \leq 2e_c + 1$ a set of e_c errors, while they might be detectable, can alter c_r sufficiently so that the closest valid code word is not the one transmitted, and the correction circuitry will therefore produce an incorrect result or miscorrection.

This report generalizes the analysis by considering the possibility of coding systems to detect e_d errors and correct e_c errors, with $e_d > e_c$. These systems require a minimum distance $d_m \geq e_c + e_d + 1$. For a given level of error detection, each additional ECC correction increases the required minimum code word distance d_m by one bit.

Both ECC and EDC work by using the principle of taking the data that is to be transmitted, let us say it is k bits wide, and generating based on this a set of r check bits. The check bits with the data bits form an $n = k + r$-bit-wide *code word*, that is transmitted in place of the original data.

In normal operation, EDC will consume less transmission bandwidth than ECC, due to the nature of the two systems. Recall that the coding scheme transmits $n = k + r$ bits of code word for every k bits of actual data. The coding overhead in bandwidth is therefore r/k. For a given coding scheme and data width k, number of required checkbits r is a strongly increasing function of d_m. Therefore any decision to reduce the amount of error-correction power or the code will reduce d_m, reduce r, and therefore reduce the bandwidth consumed by the coding scheme.

When an error occurs, EDC will require additional bandwidth and impose additional latency for the ARQ and reply. The ECC system will have an advantage in this case However, if the probability of a given code word being received in error is low, the average performance of the system is governed almost completely by the case where the code word is received without error. Therefore the ECC system, which imposes extra overhead on *every* transmission (with errors or not) and involves additional logic complexity as well, is simply not appropriate. This report will therefore assume the use of an EDC with ARQ on error.

6.2.2 EDC

There are several excellent references on coding theory and practice available [39,40], and all that is needed is a basic understanding of coding techniques because this discussion concerns error detection, not error correction.

Because the system will require an error-control code that imposes the minimum possible latency on the data transmissions, a *linear block code* will be adopted, where each of the p check bits is generated by calculating the exclusive-or of a given subset of the k data bits. This is illustrated in Figure 20.

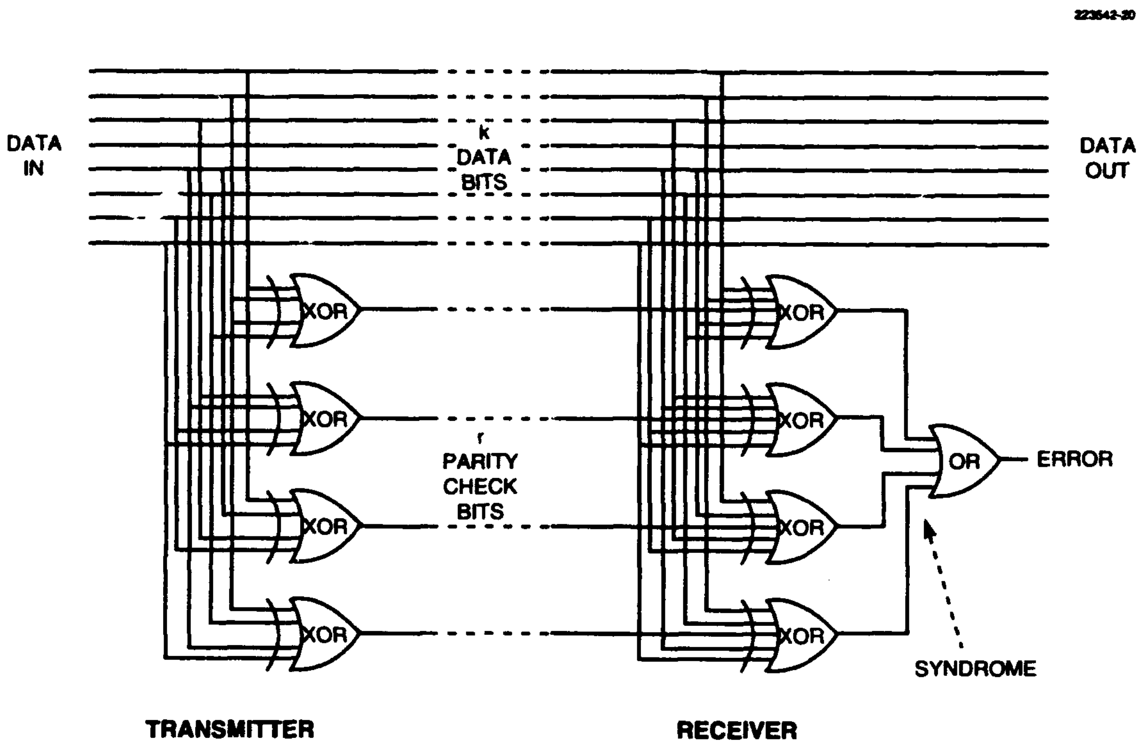

Figure 20. Linear block code generation and checking.

In the receiver, each received check bit is exclusive-or'ed with the same set of data bits used to generate it. If the word is received with no errors, the set of results from all these calculations (called the *syndrome*) should be zero. If any bit in the syndrome is nonzero, then the received word is declared to be in error, and retransmission of it must be requested.

For this discussion, there are two codes of particular interest: the parity code, and the Hamming code. The parity code is the simplest possible linear block code: merely the exclusive-or of all the data bits. The parity code adds $r = 1$ check bits to the calculation and has a minimum distance $d_m = 2$. It can therefore detect one error.

The Hamming code is a distance-3 code, and can therefore detect two errors. A Hamming Code with r check bits can handle up to $k \leq 2^r - r - 1$ data bits. A parity check bit can be added to the Hamming Code, forming a distance-4 code that can handle $k \leq 2^{r-1} - r - 1$ data bits.

If a transient-error BER is required, after detection, of 10^{-23} or less, and the actual BER is much greater than this, then the EDC system must have the ability to detect a very large percentage of the errors that do occur. Because the BER calculation in Section 6.1 assumed a code word width of 74 bits, the equivalent word-error-rate (WER) requirement is $10^{-23} \cdot 74 \approx 7 \times 10^{-22}$, that is, of every 7×10^{-22} words received, one can afford to pass, on average, only one word with undetected errors.

Given that each data and check bit is transmitted and received separately, by different transmitters and receivers, this report assumes that transient errors on different bits in the same word are independent. Therefore, the number of errors occurring in a code word will be equivalent to the number of arrivals in a Bernoulli process, over $n = k + r$ trials. Therefore, if the probability that an individual bit is received in error is denoted as p_e (note that p_e=BER) and e_w denotes the number of errors occurring in a code word of n bits, then assuming p_e = BER $\ll 1$, the probability mass function of e_w is given by

$$p(e_w) = \binom{n}{e_w}(1 - p_e)^{n-e_w} p_e^{e_w} \approx \binom{n}{e_w} p_e^{e_w} \qquad (9)$$

Therefore, if a coding system is implemented that can detect e_d errors in a code word, the undetected WER will be $\sum_{e_w=e_d+1}^{\infty} p(e_w)$. Setting WER= 7×10^{-22}, and the data width $k = 66$ (64 data bits + 1 flow control bit + 1 error signal bit), the relations between d_m, e_d, r, and p_e, for the codes discussed earlier, are shown in Table 1.

TABLE 1

EDC Characteristics

Type of Code	Minimum Distance (d_m)	Detected Errors (e_d)	Check Bits (r)	Maximum (link) WER	Maximum BER (p_e)
None	1	0	0	7×10^{-22}	10^{-23}
Parity	2	1	1	10^{-11}	6×10^{-13}
Hamming	3	2	7	2×10^{-7}	2×10^{-9}
Hamming	4	3	8	10^{-5}	2×10^{-7} and Parity

It can be seen that a simple code such as the distance-4 Hamming code (Hamming + parity) relaxes the raw BER requirement of the optical link from 10^{-23} to more than than 10^{-7} at a very modest bandwidth overhead of $r/k = 8/66 = 12\%$.

Let us now test the hypothesis that the retransmission overhead due to using EDC (instead of error-correction) is negligible. Supposing the optical links were operating at the relaxed BER$\approx 10^{-7}$ allowed by the distance-4 Hamming EDC, the raw WER (before detection) would be 7×10^{-6}. Supposing that regular transmission of a code word takes one cycle, ECC correction takes one additional cycle, and error-detection automatic retransmission takes 25 cycles (as might be the case for a heavily pipelined system), then there would be error-correction calculation overhead of $1 \cdot 7 \times 10^{-6} = 0.00001\%$, and an error-detection retransmission overhead of $25 \cdot 7 \times 10^{-6} = 0.0002\%$. In both cases, the performance overhead due to errors is trivial, and the hypothesis is validated.

6.2.3 EDC System Implementation

Figure 20 already gives the basic form of implementation for the EDC system. The complexity of the circuit is minimal, and the latency will depend on the number of inputs required to the exclusive-or gates at both receiver and transmitter.

Fortunately, the codes used are *linear*, that is, the n-element code word C can be considered as being the product of a k-element input data vector D and a $k \times n$ code generator matrix G: $C = DG$. The receiver checks the received code word $\hat{C}$ by multiplication with a parity-check matrix H, resulting in a syndrome that will be checked for the equality $\hat{C}H = 0$.

Because both the parity generation and checking are linear processes, they can be reconfigured by standard linear transformations. Hsiao [41] has examined such transformations on the distance-4 Hamming code and has found its optimal minimum-weight form (that is, the form requiring the fewest exclusive-or inputs).

With a data width $k = 66$, and $r = 8$ parity check bits, the Hsiao code has a total weight of 226, or an average of 28.25 inputs to the parity-generating exclusive-or gates (six 28-input gates, and two 29-input gates). This could be implemented in five levels of 2-input exclusive-or gates, in one or two 1-ns clock cycles.

6.3 Error Diagnosis

The laser drive control scheme described in the following section assumes a knowledge of error rates for each of the optical links. The fault tolerance schemes outlined in Sections 8 and 9 assume that failures can be easily localized and identified. Fortunately, such diagnostic tasks are straightforwardly implemented in the EDC scheme proposed here.

6.3.1 Transient Error Diagnosis

In Figure 20, recall that the error-checking circuit generates a *syndrome* as a part of the error detection algorithm. In the case of the Hsiao code, the syndrome is eight bits wide. If all the syndrome bits are zero, then no (detectable) error has occurred; if the syndrome is nonzero, an error has been detected.

Each possible one-bit error has a unique syndrome associated with it. A simple table lookup could therefore identify the bit in error, if there were only one. (This is, in fact, the fundamental basis of ECC schemes.) This lookup might be done in software, if error rates were low enough, or with specialized hardware, if it proved to be too great a load on the processors.

The offending bit could be diagnosed in each single-bit error word received; then, that information could be used in the control schemes described in the following sections. The rare multibit errors would be noted, but not otherwise diagnosed.

Would this diagnosis system therefore be implementing ECC after all? No, because it is not actually relying on the error diagnosis to protect data integrity. Because the error diagnosis would only be used for statistical purposes, one can occasionally omit the diagnosis (in the rare case of two-bit errors) or tolerate a few misdiagnoses (in the rarer case of three-bit errors). Also, the error diagnosis is not in the chain of data transmission, so it can be done with very little constraint on the amount of time required to perform the diagnosis.

6.3.2 Hard Failure Diagnosis

A hard failure will be detected quickly. It will first be detected by the error-detection circuitry, which will request a retransmission. The retransmission will also be in error, so another will be requested, which will be in error, and so forth. After some number of retransmissions (perhaps three or four), the system would flag a possible error. Because hard failure is a rare event, its diagnosis can be handled completely in software, with negligible performance impact.

The syndrome diagnosis described in the previous section would point out the likely culprit, and transmission of a few test patterns could easily confirm the diagnosis. The failure diagnosis would then be acted on by one of the algorithms described in Sections 8 and 9.

6.4 Summary

The multiprocessor network described in Section 5 would require an undetected BER of 10^{-23} or less. This is completely impractical to realize by direct implementation, but it can be achieved using error control coding with moderately reliable communication links.

For this application, EDC is more appropriate than ECC. Assuming independent errors, a Hsiao EDC can relax the BER requirement to around 10^{-7}, with only 12% parity-bit overhead and a pipeline delay of one or two cycles. Such a system can also diagnose transient errors and hard failures for the use of laser control and fault tolerance systems.

7. INTELLIGENT LASER DRIVE CONTROL

The error-control strategy proposed in the preceding section, in addition to directly solving the transient-error problem, can facilitate the solution of another system problem that at first glance might seem to be unrelated: the problem of laser drive level control.

A proposal for solving this problem is intelligent (that is, software-based)[3] laser drive control. Instead of presently-used methods that need monitor photodiodes, intelligent laser drive control uses the error rate itself (derived from the error detection system) to control the laser drive levels.

In addition to solving the drive control problem this can provide additional benefits, including detection of and compensation for optical medium degradation, and tracking of laser wear-out trends to allow optimal repair and replacement planning.

7.1 The Laser Drive Problem

To transmit data via a laser diode, one may employ any of a number of modulation methods, such as frequency, phase, or amplitude modulation. This report uses one of the simplest possible (and most widely used) laser modulation methods: on-off signaling.

Figure 21 gives the typical light output L vs drive current I relationship for a semiconductor laser. To achieve a desired light output power level L_{on} (governed by the receiver characteristics and the BER requirements), the system must apply a corresponding drive current I_{on} for the "on" bit signal, and the drive current must be removed for the "off" bit signal.

However, the "off" current I_{off} cannot usually be set to zero, because the laser output will take some considerable time to increase from zero to the level corresponding to I_{th}, the onset of lasing. (The laser operates as a light-emitting diode in the $0 \leq I \leq I_{\mathrm{th}}$ range of drive currents.) High-speed operation therefore does not allow I_{off} to be set much lower than the laser threshold current I(th). Additionally, I_{on} must be set sufficiently above I_{th} to produce enough light output, because $L \propto I_{\mathrm{on}} - I_{\mathrm{th}}$.

Figure 22 shows a conceptual design for a laser drive circuit that is used to meet these constraints. The current source I_{bias} is constantly applied to the laser, and the current source I_{pulse} is switched through the laser or not, depending on the input data, therefore

[3] By using the term "intelligent," it is not implied that the software-based feedback scheme need be complex, much less that it include artificial intelligence. Rather, a software-based scheme gives one the flexibility to tailor the feedback algorithm to have as much or as little sophistication as the situation requires.

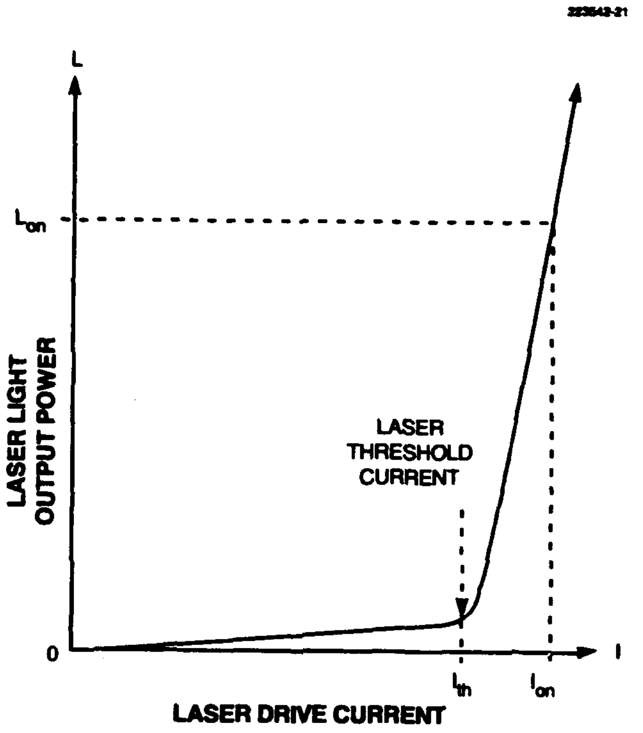

Figure 21. Semiconductor laser output power vs drive current.

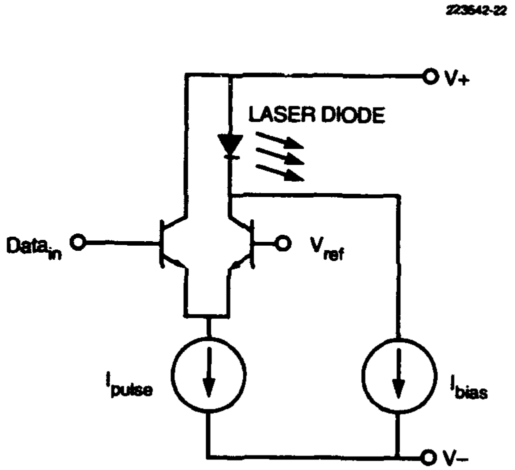

Figure 22. Semiconductor laser driver circuit.

$$I_{\mathrm{off}} = I_{\mathrm{bias}} \tag{10}$$

$$I_{\mathrm{on}} = I_{\mathrm{bias}} + I_{\mathrm{pulse}} \tag{11}$$

The problem addressed here is that of properly controlling the I_{bias} and I_{pulse}.

As noted in the analysis of laser wear-out failures in Section 4.2.1, such failures can be modeled as a gradual increase in laser threshold current over time. This will result in a decrease in laser light output unless the laser drive circuit compensates for it by increasing the drive current.

There is a also more immediate laser-threshold-current problem: temperature dependence. Laser threshold current increases exponentially with temperature, so the drive current must be compensated for this as well, or else the drive level must be set high enough for the hottest laser operating temperature (and oldest laser age) that will be encountered. Of course, the high drive level may ensure that the laser actually does reach a high temperature.

7.2 Conventional Solutions

Laser-diode-based systems are in wide use now, in spite of the laser threshold current variation problem. A number of approaches to solving the problem are in use now and are described below, but none of them seems completely satisfactory for the multiprocessor optical networks envisioned in this report.

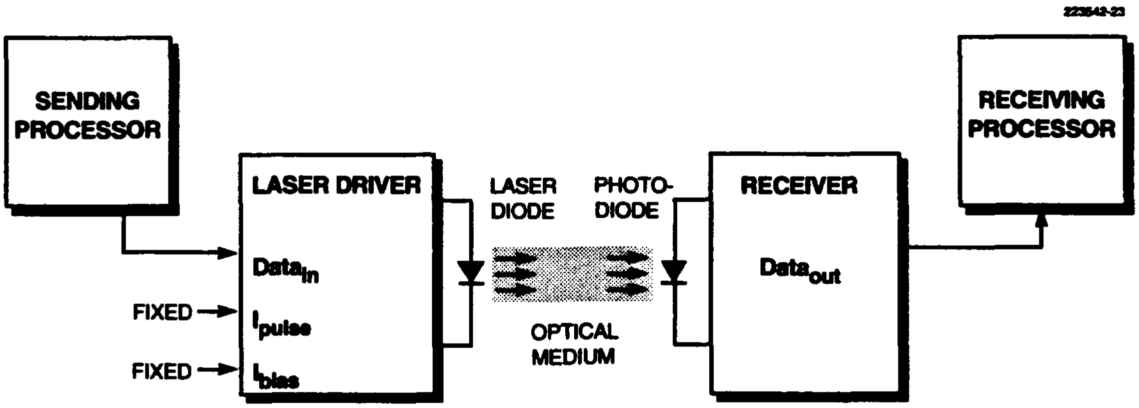

Figure 23. Fixed laser drive.

7.2.1 Fixed Laser Drive

The simplest approach (shown in Figure 23) to these problems is to fix I_{bias} and I_{pulse} for acceptable performance over temperature and laser lifetime. This can involve significantly higher drive levels than would otherwise be required. High drive levels, of themselves, shorten laser lifetime, and they contribute to higher operating temperatures, which shorten all component lifetimes. Also, this means that at low threshold currents (that is, at low temperature and/or using new lasers) the lasers may not be completely "off" at I_{bias}, degrading the laser output extinction ratio, that is, the ratio between "on" and "off" light levels.

7.2.2 Analog-Feedback Laser Drive Control

Figure 24 shows the approach conventionally used in telecommunications [42]. A small photodetector is added to the laser package, detecting some of the light leaked from the rear facet of the laser (or tapped from the laser output), and producing a photocurrent proportional to it. This current is used to control I_{bias}, by implementing an analog feedback loop to keep the laser output level constant. I_{drive} is usually kept fixed, but some of the more elaborate systems vary it as well.

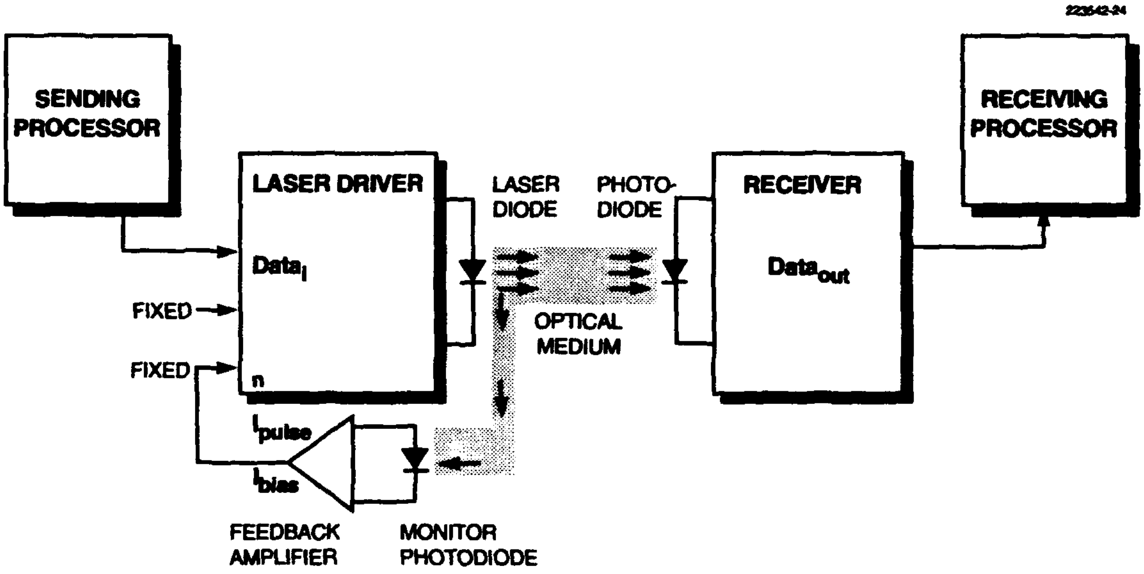

Figure 24. Analog-feedback laser drive control.

[In telecommunications, the control system is often more complex: in addition to the light-monitor laser drive control loop, a temperature-control feedback loop is implemented as well. A temperature sensor is used for feedback control of a Peltier-effect thermoelectric cooler, to help keep the laser temperature constant. Even with this addition, the light-monitor control of laser drive levels is usually required.]

Unfortunately, this approach does not transfer well to the wide-path laser-array interconnects that this report anticipates. Implementing monitor diodes on such laser arrays adds a further complication to an already difficult optoelectronic device design. Even with the monitor photodiodes, proper implementation of the required analog feedback loops would be another stumbling block.

7.2.3 Very-Low-Threshold Lasers

Because of these problems, some researchers in this field have restricted themselves to laser designs with very low threshold currents, and therefore with weaker dependencies on temperature and age, to enable them to use fixed-current drivers. This is a very sensible approach, provided that lasers with suitably low threshold currents are available.

A usable laser-drive-control system may be able to widen the field of potential laser array designs to include higher-threshold lasers. The criterion of "suitably low threshold" would then become part of an engineering trade-off: very-low-threshold designs using fixed drive, vs moderately-low-threshold designs using laser drive control. Even very-low-threshold designs may benefit from drive control, as it may extend their utility by restricting laser power to the minimum required, thereby reducing system power and increasing component lifetimes.

The drive control scheme proposed in the next section also gives other benefits, such as laser wear-out tracking and optical medium degradation detection and compensation. These benefits might make it worthwhile to include drive control even in very-low-threshold laser designs that did not strictly require it.

7.3 Intelligent Laser Drive Control

The new idea proposed here is to use the actual link error performance, rather than the light output, to control the laser drive levels: in the final analysis, it is the *performance* of a data link that counts, rather than the intensity of the light used to implement it.

The lowest-possible error rate, which most communications-link designers have sought, is not really required for the multiprocessor optical network. As already shown, a network with error-detection and automatic retransmission request can tolerate a BER on the order of 10^{-8} or 10^{-9} and still achieve the desired system-level data integrity.

It therefore might be possible to use the BER rather than a direct measurement of laser output power to control the laser drive because short periods of elevated error rates can be tolerated. [If one is to use the BER as a feedback control variable, one must allow it to fluctuate somewhat.]

Figure 25 shows an intelligent (software-based) laser drive control loop. The entire control loop is digital, except for the laser driver itself and the DACs controlling it. The addition of two

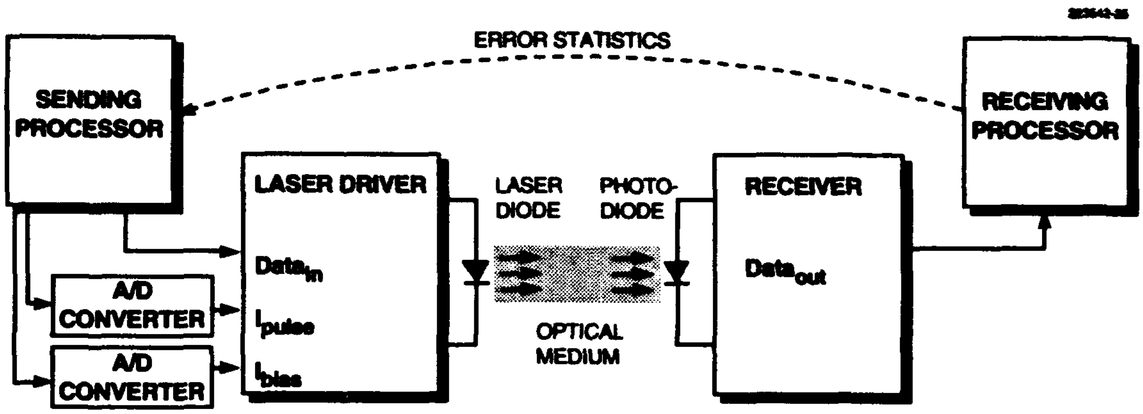

Figure 25. Intelligent laser drive control.

DACs per laser might seem a significant increase in complexity, but this application is particularly undemanding for the DACs because none of the three (high speed, high precision, and high accuracy) individually or in combination is required. (The question of just how much precision might be required is investigated in Section 7.4.7.) They can therefore be easily implemented *en masse* in silicon very-large-scale integration (VLSI) integrated circuits for all the lasers in a particular data channel.

A fundamentally important aspect of this approach is that it is performed in software. This allows much more flexibility in the control algorithm than would be available in a simple analog control loop. For example, it can easily control two parameters (I_{bias} and I_{pulse}) instead of one. It can also record past values, enabling the system to perceive anomalies such as optical medium degradation and long-term trends such as laser threshold increase due to aging.

7.4 Laser Drive Control Experiments

To test the feasibility of the intelligent laser drive control concept, it was implemented on an experimental 1 Gbit/s free-space optical data link. Experiments were conducted to determine feedback system's performance with

- Temperature variation
- Optical loss variation

The temperature variation would be a real problem in itself, and it is also a surrogate for laser aging because a hot laser is similar to an old laser: both have elevated threshold currents. The optical loss variation could occur from medium degradation or from age-related reduction in laser efficiency.

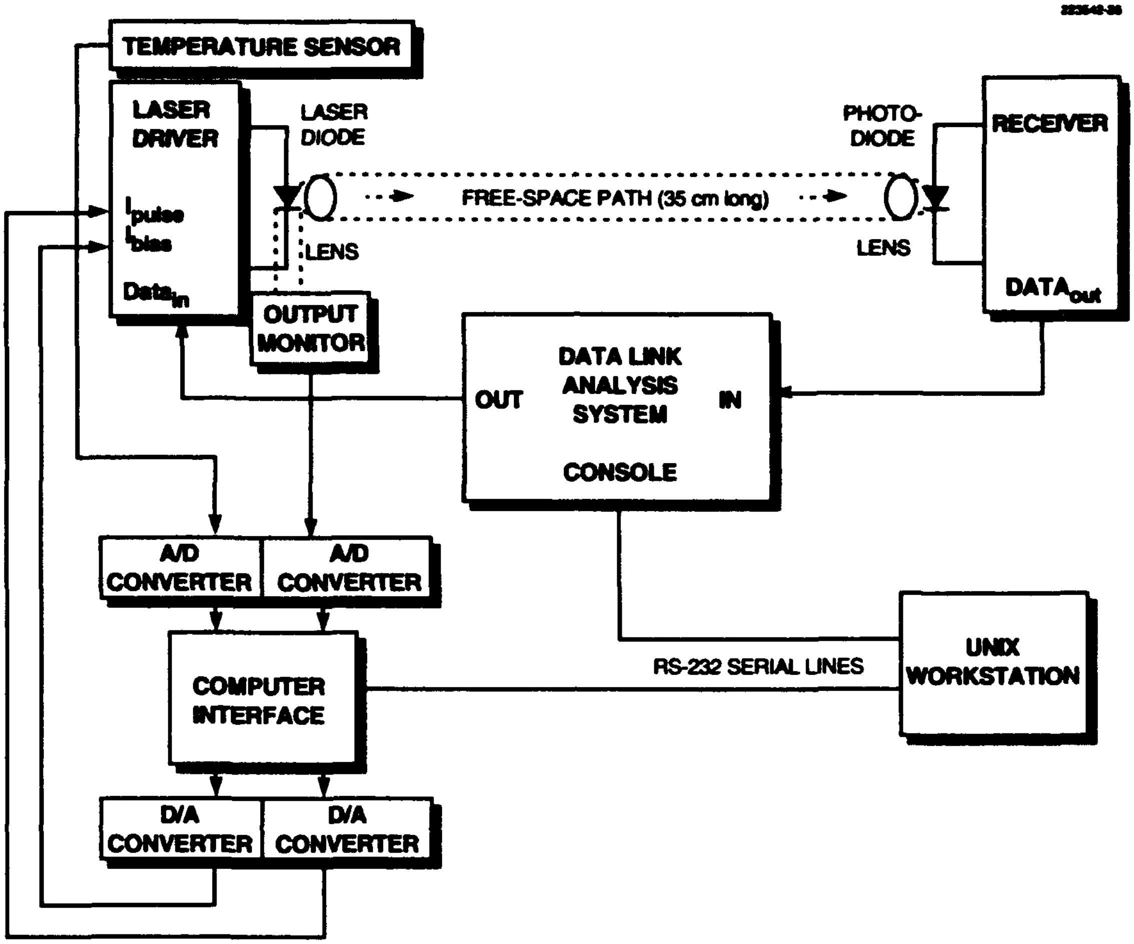

Figure 26. Intelligent laser drive experimental setup.

7.4.1 Experimental Setup

The experimental setup is shown in Figure 26, and a general outline of it is given here. Details of the experimental setup are given in Appendix A.

A laser transmitter board and an optical receiver board are mounted on a light table, at a distance of 35 cm. The laser board holds a laser (with a monitor photodiode), lens, laser driver, and a temperature sensor (mounted in an aluminum block with the laser). The receiver board holds a receiver photodiode, lens, and amplifier.

A data-link analysis system made by Gazelle sends a 1 Gbit/s data stream to the laser driver and receives data from the optical receiver. It compares the two and reports the error results via a low-speed serial interface. The Gazelle system usually communicates with a console terminal, but in this experiment it is connected to a Sun UNIX workstation instead.

A custom-made interface system communicates with the same UNIX workstation, also through a serial link. Via the interface, the workstation can control bias and pulse currents in the laser driver, and it can monitor the laser light output level and the laser case temperature.

Figure 27 is a photograph of the overall experimental setup. The photograph includes a later addition to the experimental setup: an adjustable iris that can block part of the free-space optical path. The optical components are mounted on a bench-top vibration-isolated optical table. From left to right they are: photodiode/receiver board, optical iris, and laser/driver board. Next to the optical table, one can see the computer interface box and the data link analysis board.

7.4.2 Feedback Program

The UNIX workstation runs a program that implements the actual feedback loop. It receives the BER results from the data-link analysis system and controls the laser bias and pulse currents based on the BER results. The program logs the experiment's progress in a disk file, recording BER, bias and pulse current, light output level, and temperature. [Note, however, that the temperature and light output level measurements are strictly for off-line analysis of the experimental results and are not used in the feedback system.]

The program's feedback algorithm is an unsophisticated one and could almost certainly be improved with the benefit of more experience. The algorithm assumes that the open-loop response of the system (that is, drive current → BER) is monotonically decreasing, allowing the use of a simple rule:

- Low error rate ⟶ decrease drive current
- High error rate ⟶ increase drive current

The question of *which* drive current (bias or pulse) to adjust is an interesting one, which the feedback algorithm avoids by treating the two currents equally, as much as possible.

The workstation program is given a BER goal and increases the laser drive currents whenever the observed BER is worse than the goal. When the observed BER is better than the goal, the program tests to see if it can safely reduce the drive or bias current without increasing the BER to above the goal. The algorithm is controlled by a simple finite state machine, the state transition diagram of which is given in Figure 28. The bias current step is 0.12 mA, and the pulse current step is 0.07 mA: in both cases approximately 1% of the normal operating-point value.

When faced with an excessively high error rate, a sophisticated feedback program might try to determine which drive current to increase. Alternatively, the program could initially assume (as

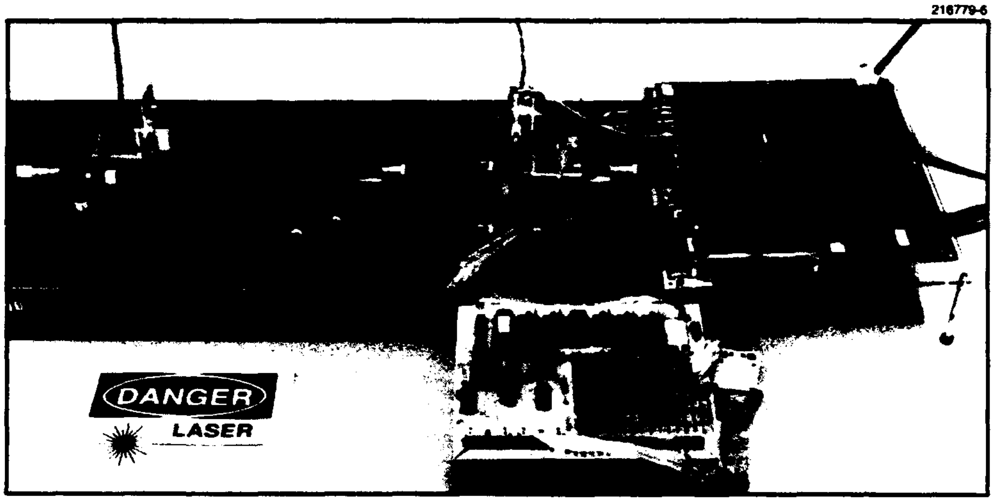

Figure 27. Photograph of laser drive control experimental setup.

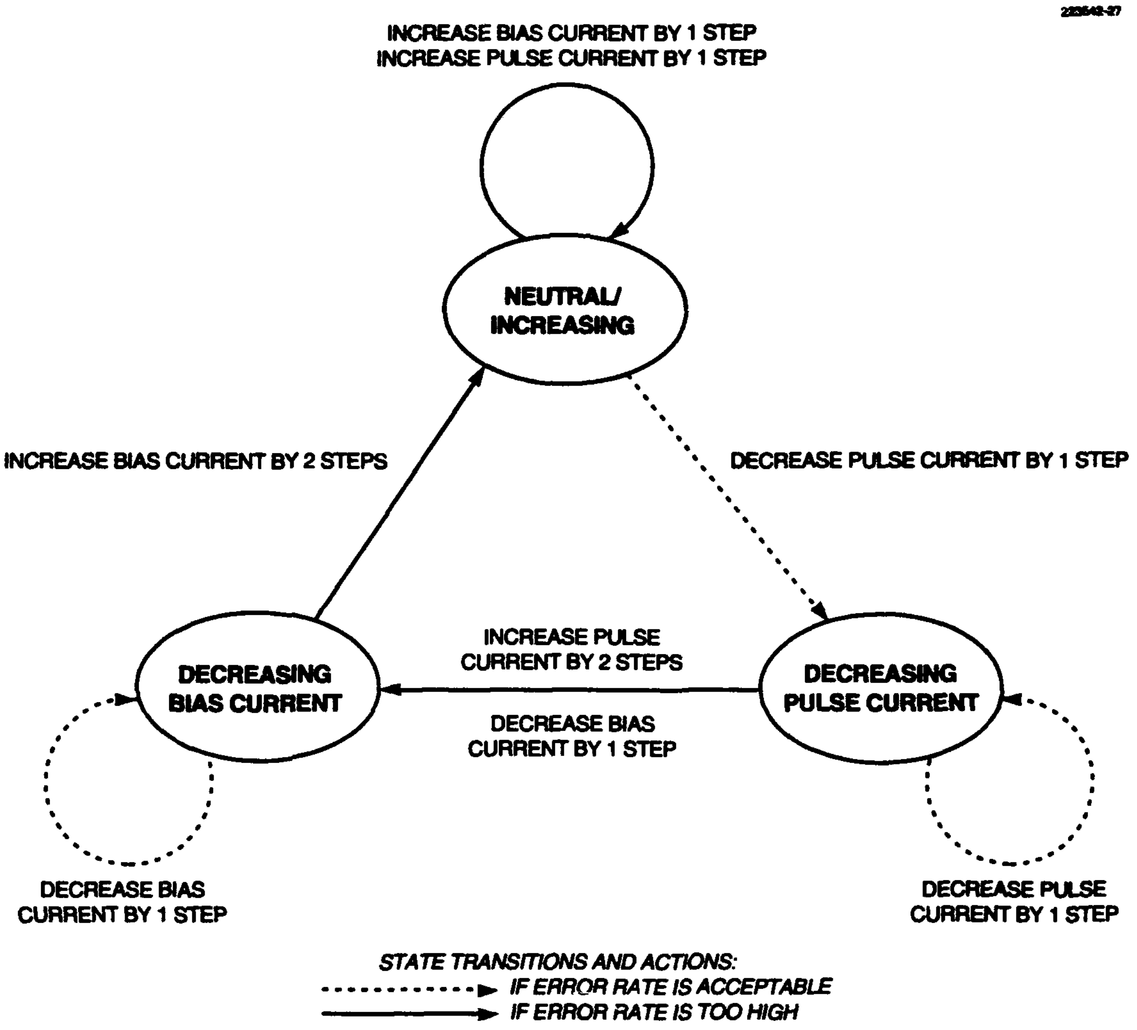
223642-27
INCREASE BIAS CURRENT BY 1 STEP
INCREASE PULSE CURRENT BY 1 STEP
NEUTRAL/ INCREASING
INCREASE BIAS CURRENT BY 2 STEPS
DECREASE PULSE CURRENT BY 1 STEP
INCREASE PULSE CURRENT BY 2 STEPS
DECREASING BIAS CURRENT
DECREASING PULSE CURRENT
DECREASE BIAS CURRENT BY 1 STEP
DECREASE BIAS CURRENT BY 1 STEP
DECREASE PULSE CURRENT BY 1 STEP
STATE TRANSITIONS AND ACTIONS:
IF ERROR RATE IS ACCEPTABLE
IF ERROR RATE IS TOO HIGH

Figure 28. Laser drive control software state diagram.

is most likely) that only the bias current need be increased (only increase the pulse current if the bias current increase did not work).

In Figure 28, the feedback program blindly increases both bias and pulse drive currents until a satisfactory error rate is achieved. This will almost certainly increase one of the currents (bias or pulse) more than actually required. The program then reduces the pulse drive current until the error rate rises again, and then backtracks a small amount. The bias current is then reduced similarly. These reduction mechanisms should eventually eliminate the unneeded drive current produced by the drive-increase code.

7.4.3 Temperature Experiments

It would be impractical to observe the system's control of laser wear-out problems because such wear out occurs over the course of months and years. However, a similar problem occurs much more quickly: temperature variation of laser output. Therefore, the laser drive control system's ability to deal with the temperature variation of laser threshold current is observed, both as a problem in its own right and because of its similarity to laser wear out.

An electric heating element was placed next to the laser board, and the heat output was controlled to vary the laser case temperature between room temperature and 40°C. First, the system was temperature-cycled with the laser bias and pulse currents fixed, then the feedback program (with a BER goal of 2.5×10^{-10}) was enabled on the UNIX workstation and the experiment was repeated.

The results from both experiments are shown in Figure 29. With fixed drive, there is a 50% reduction in light output at higher temperatures, due to increased laser threshold current; this reduction in light output is accompanied by a corresponding increase in BER to over 10^{-3}.

With feedback enabled, even though the light output monitor values played absolutely no part in the feedback loop, and the laser temperature rose even higher than in the fixed-drive experiment, the change in laser light output over temperature is almost imperceptible. The BER is also well-controlled, to less than 10^{-9}, aside from a 4×10^{-7} value at one data point[4].

This result shows that even a very simple BER-based feedback algorithm can adequately control the laser drive level over changes in laser threshold current.

[4] This is most likely due to the artificially rapid temperature rise, and the experimental setup's slow response time because of the low-speed interface with the data link analysis board, which was not designed for this type of application. In an actual network, error rates (when they are *high*) would be available very quickly.

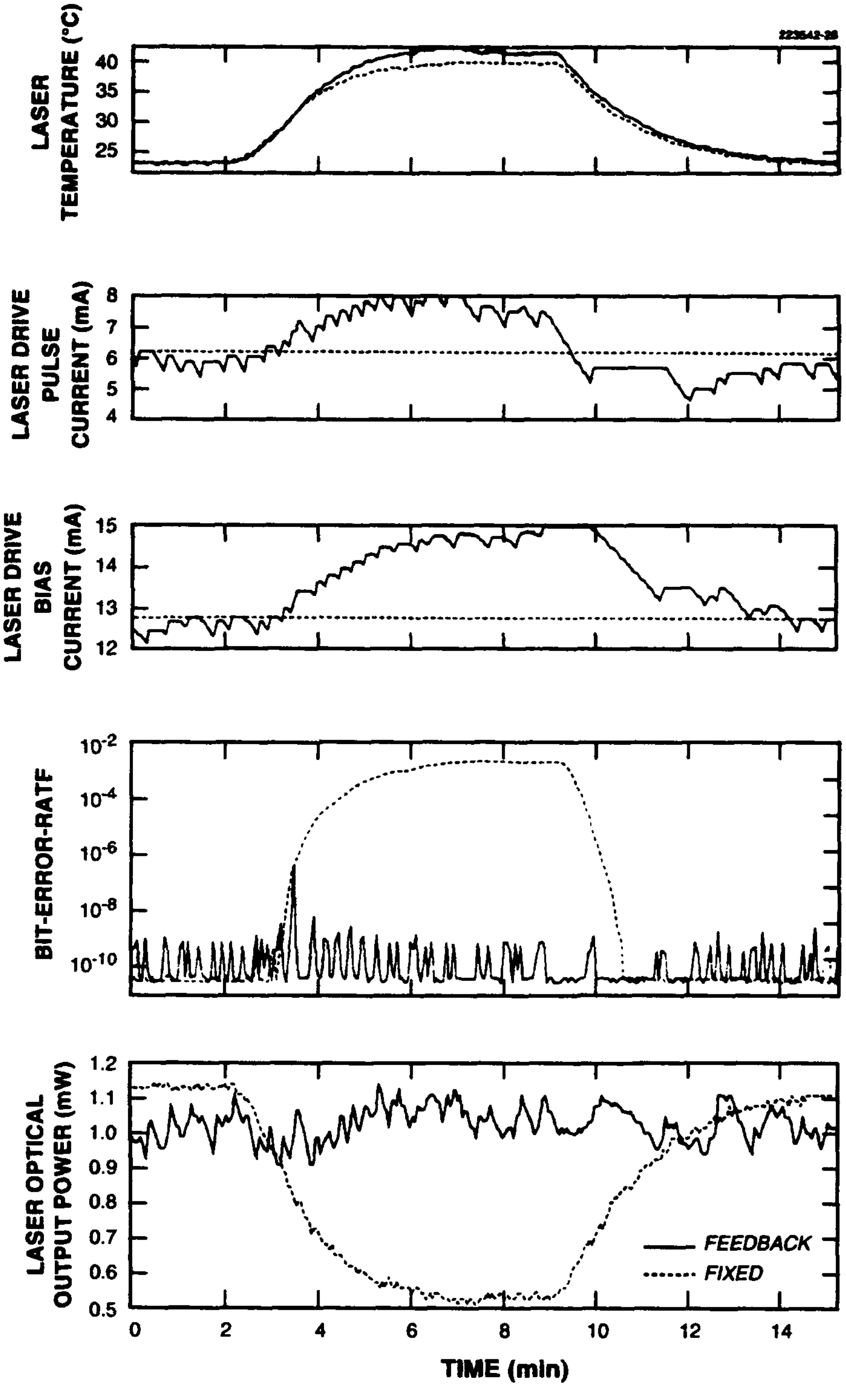

Figure 29. Laser drive control over temperature.

7.4.4 Optical Loss Experiments

This report then examines a problem for which the conventional approach (in Figure 24) would be of no use: optical medium degradation. It could easily happen that the optical medium between laser and receiver could experience an increased optical loss, due to aging, misalignment, damage, or other causes. Because the approach includes the entire communication link in the feedback loop, it should be able to detect and compensate for such a problem, provided that sufficient laser output were available.

The setup was modified in Figure 26 to include an adjustable iris in the free-space light path. When open, the iris had no effect on the light beam. When closed, the iris blocked almost all of the beam, allowing only 2.6% of the light to pass.

The iris was first tried with the laser drive feedback control disabled, and then the experiment was tried with the feedback program running as before. Figure 30 shows the results. When the iris is closed, in both cases the BER immediately increases to more than 10^{-6}. In the fixed-drive case, the BER remains at that level, while in the feedback case it almost as quickly starts falling, as the feedback loop increases the laser drive level.

The effect of the feedback algorithm can be seen between time $t = 2$ and $t = 6$ min, where the bias current has been increased in lock-step with the pulse current. By time $t = 7$ min, the bias has been reduced back to its previous value, while the pulse current remains elevated, as one would expect. After the iris is opened, the pulse current subsides to somewhat higher than its previous level, with the bias somewhat lower. (This is an alternative operating point with approximately the same BER at the one at $t = 0$. Either point is acceptable.)

The laser optical output power readings provided a clue to a shortcoming in the experimental setup. Note that in the fixed-drive case, the power reading falls when the iris is closed, and rises when it is opened again. This cannot have any basis in reality, because (with the drive level fixed) the iris can have no effect on the laser output.

Investigation of this incongruity led to the discovery that the lenses used for the free-space optical path had been specified incorrectly and had an antireflective (AR) coating designed for different wavelength from the one being used. The receiver board lens was reflecting some of the incident light back to the laser, and this reflected light was being detected by the monitor photodiode (behind the rear facet of the laser).

One may also note that, although the optical transmission over the free-space path was reduced by a factor of 38, the laser output power is only increased by about 50%. The most likely explanation for this is that the normal bias point found by the feedback system has nonzero optical output power even in the "off" state, which the monitor photodiode detects in addition to the switched optical power used to transmit data. This would tend to obscure the actual increase in switched optical power.

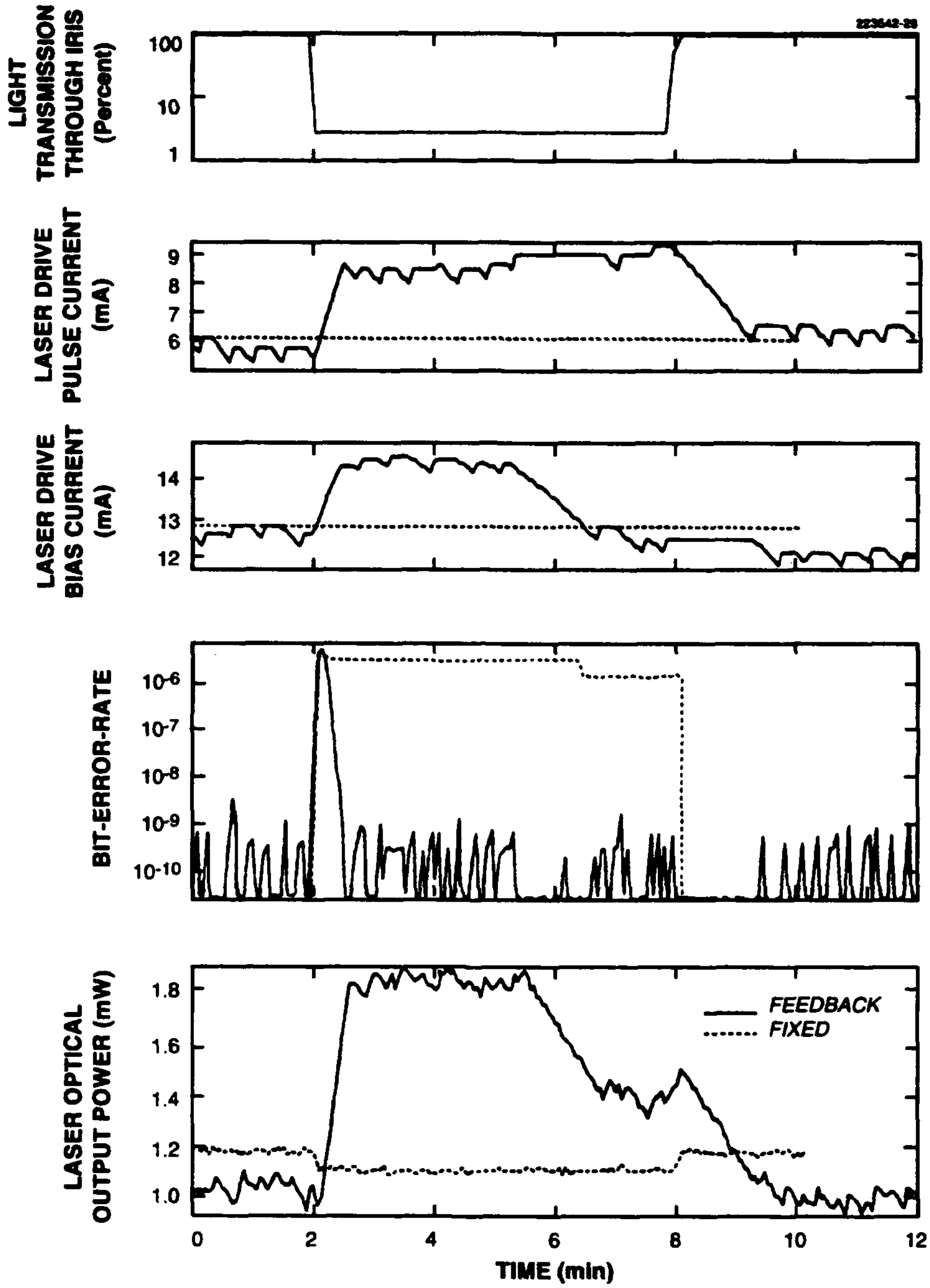

Figure 30. Laser drive control with optical loss.

The intelligent laser drive control system can also maintain operation in spite of optical medium degradation. Because the feedback is in software, a periodic diagnostic program could easily detect the significant increase in pulse current (indicative of such a degradation) and inform the operator so that the need for repair could be evaluated without the urgency of an actual failure.

7.4.5 Feedback Reduction Order

The one asymmetry between bias and pulse drive current in Figure 28 was examined: pulse current is reduced first and then bias current. The software to reverse this situation was reconfigured: reducing bias current first, followed by pulse current. Both the temperature and optical loss experiments in this new configuration were repeated.

Figure 31 shows the results of the temperature experiment, compared with those from the original configuration. Figure 32 is the corresponding plot of optical-loss results.

In both experiments, the new configuration works as well as the original one. The major difference, as one might expect, is that excessive bias current is reduced more quickly in the new configuration and excessive pulse current is reduced more quickly in the original configuration.

The new (bias-first) configuration is shown to be the best advantage in the optical-loss experiment because it involves a significant excess bias current after iris closure. The original configuration deals better with the temperature experiment because it involves an excessive pulse current after the initial heating.

Because threshold current variation is the more likely problem, it would seem that the original state-diagram configuration (in Figure 28) is the better one, if one continued to use this control algorithm. However, either configuration would seem to be adequate.

7.4.6 Bias-Only Drive Control

As indicated in Section 7.2.2, telecommunications systems generally include control of bias current only, keeping pulse current fixed. To test its performance in this mode of operation, the feedback program was reconfigured to keep the pulse drive current fixed; only the bias current was varied.

The temperature experiment was repeated with this new feedback configuration. Figure 33 shows the results, superimposed on the results from the full feedback configuration. Note that the bias-only control does an equally good job. This might be expected from the practice in telecommunications links.

The optical loss experiment was then repeated in the bias-only feedback configuration. Figure 34 shows the results. The feedback initially has some benefit in reducing BER because an increase in the bias current will improve optical efficiency. Unfortunately, the system eventually leaves the region of operation in which the monotonic open-loop response assumption is valid. In a futile attempt to restore normal operation, the feedback program increases bias current so high that the

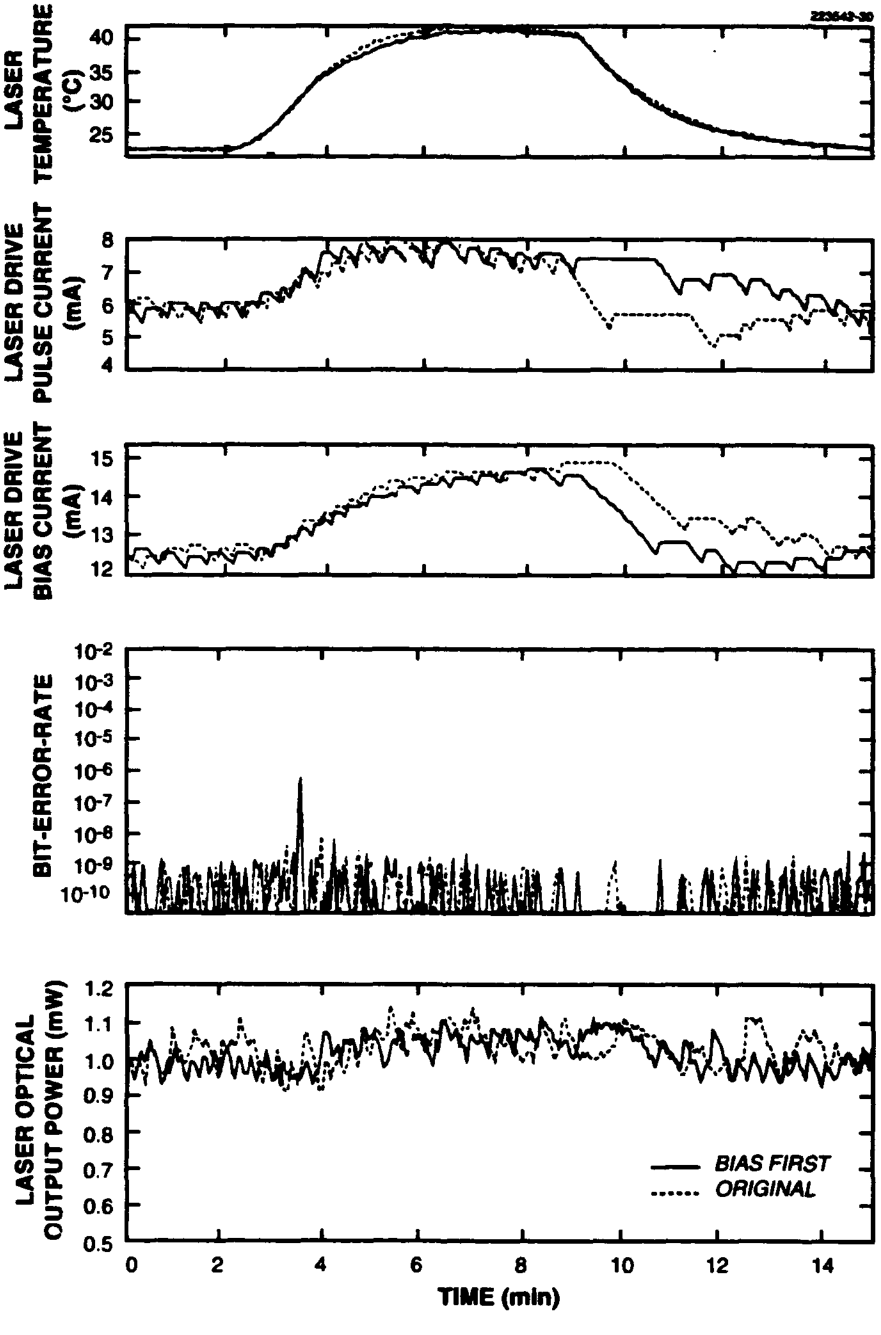

Figure 31. Feedback reduction order (temperature experiment).

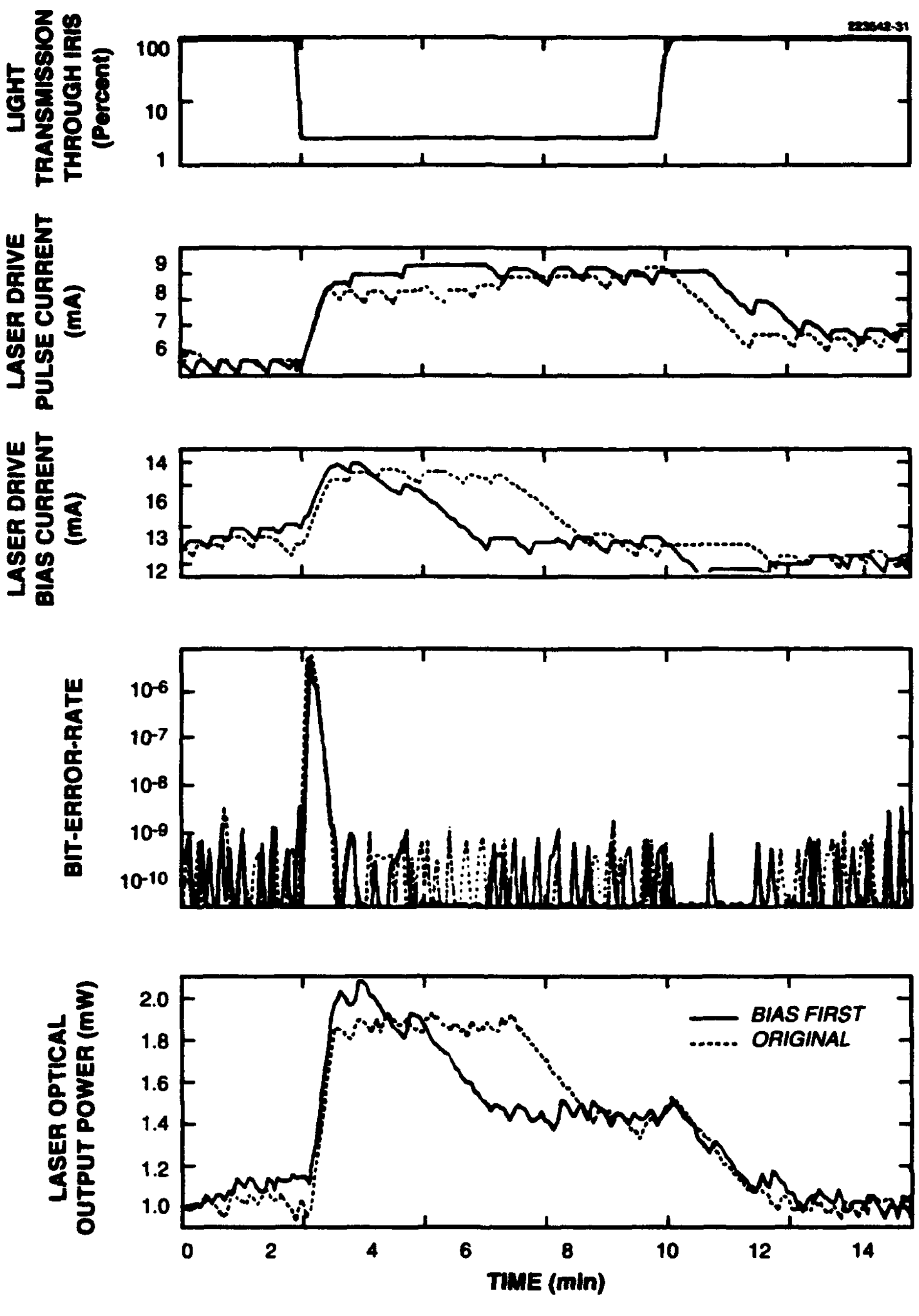

Figure 32. Feedback reduction order (optical loss experiment).

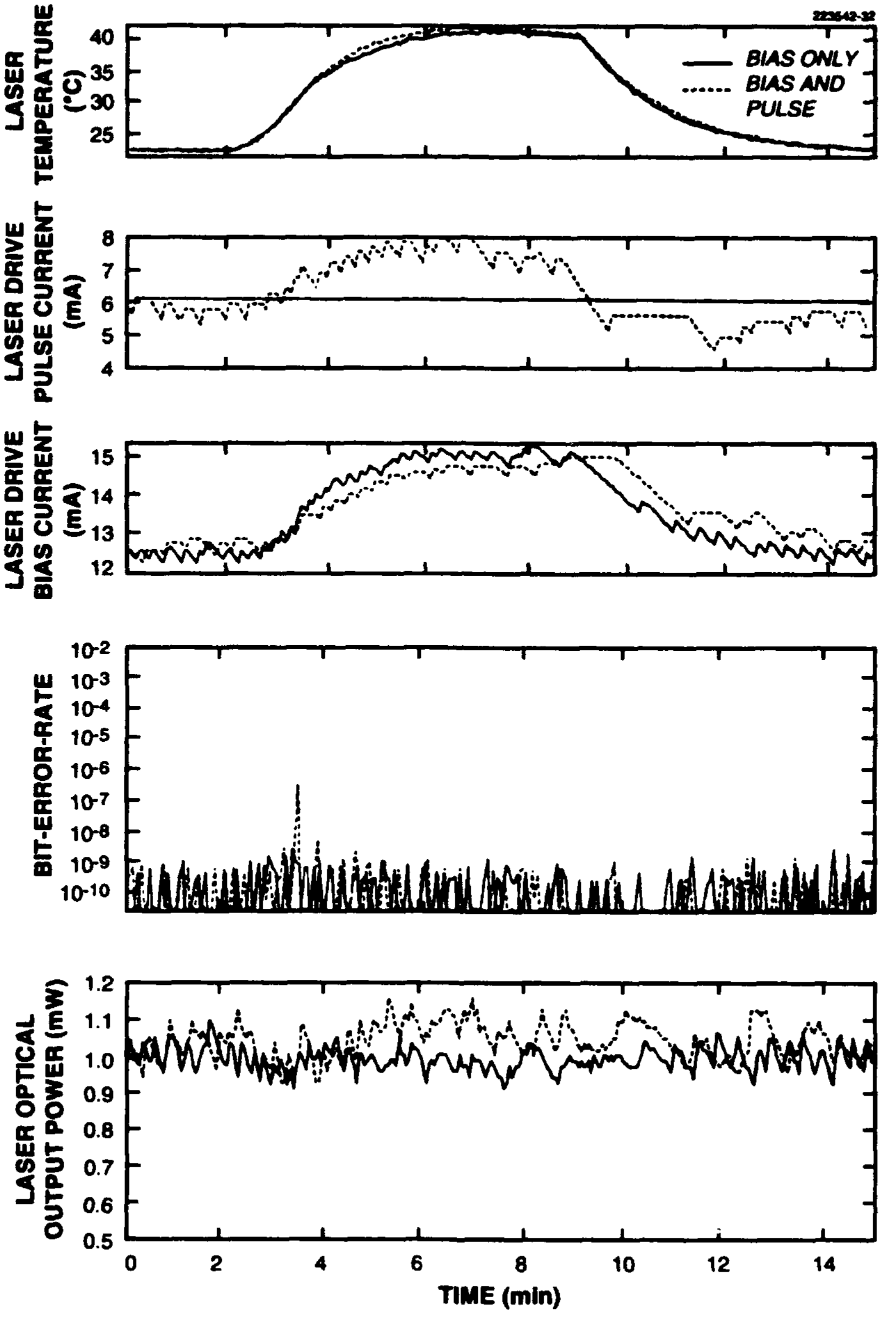

Figure 33. Bias-only feedback (temperature experiment).

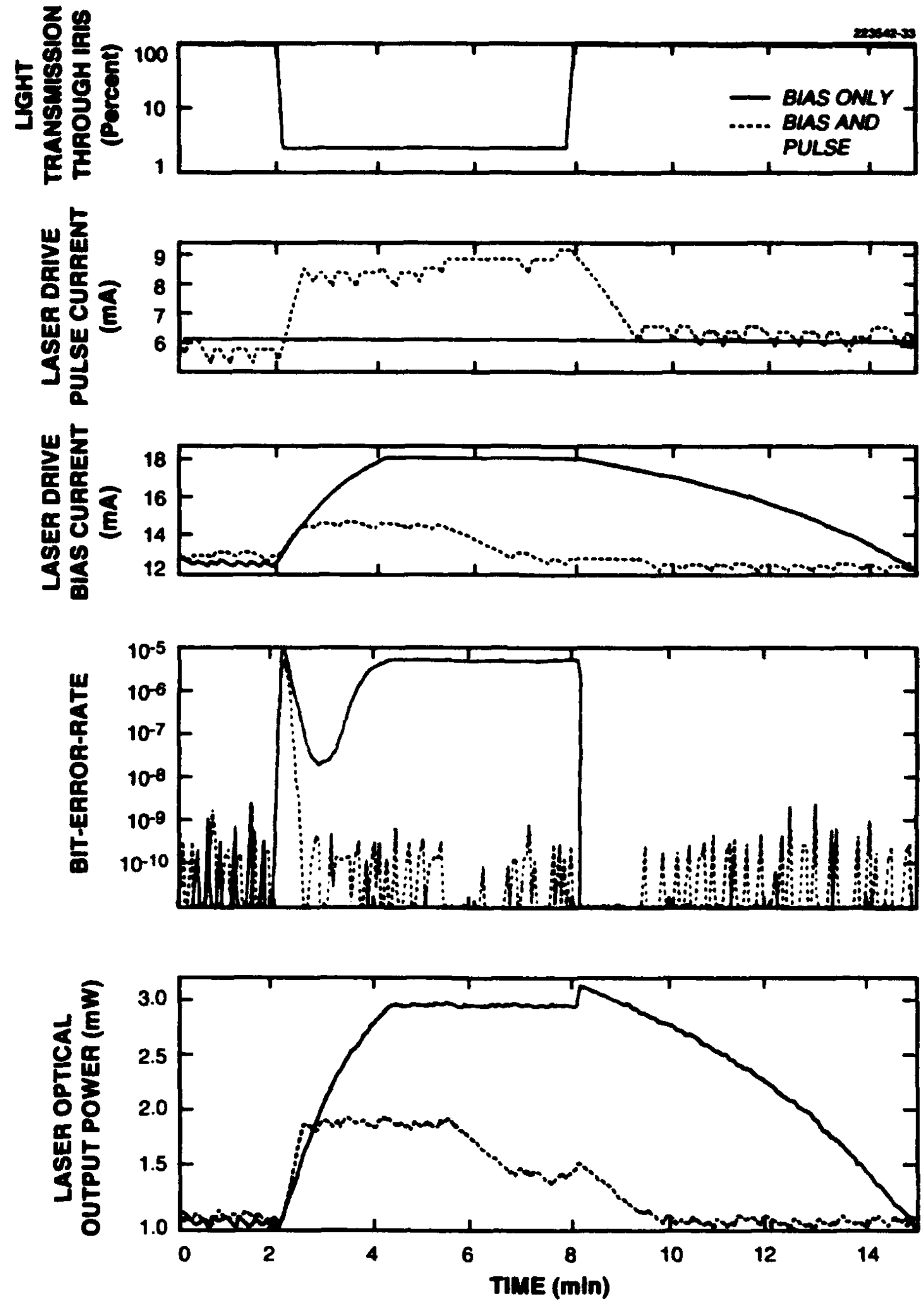

Figure 34. Bias-only feedback (optical loss experiment).

BER rises again. A more sophisticated feedback control program might detect this phenomenon and reduce the bias accordingly. The feedback algorithm continues to raise the bias until it reaches a preset safety limit in the program.

Bias-only intelligent laser drive control, therefore, seems quite adequate to deal with threshold current variation, but does not provide the optical-loss control available with the full feedback system.

7.4.7 DAC Implementation

Now the previous section's assertion (in the description of intelligent laser drive control), that the DACs needed to implement it can be made particularly simple, will be considered.

Some of the DAC requirements can be inferred from the control system structure. Absolute accuracy is irrelevant because the feedback system will inherently compensate for any bias or scale factor errors. Relative accuracy is important to the extent that the input/output function of the DAC must be monotone, else the feedback system might be caught in a suboptimal operating point, because a nonmonotonic output function might create the false appearance of a local minimum in error rate. Even the monotonicity constraint, however, can be relaxed if the DAC is calibrated before use and a calibration table is used to eliminate any kinks in the input/output function.

The required precision of the DAC is not immediately obvious. The experimental setup for the results given above used eight-bit converters, so one can conclude that eight bits will probably suffice. In itself, this is enough to validate the assertion that the control system DAC would be easily implemented. The required precision, however, needs to be established more accurately.

The feedback control program was modified to allow the drive current step sizes (quanta) to be increased, if desired. The original step sizes were about 1% of the normal operating values: the modified program allowed the step sizes to be increased to any multiple of the original step size. Operation with a larger step size is comparable to operation with lower-precision DACs. Specifically, increasing the step size by a factor of 2^n is comparable to reducing the DAC precision by n bits.

A number of bias and pulse current step sizes. To compare the different configurations, the experimental link was operated in normal, steady-state mode (room temperature, iris open), and the average laser power and average error rate for each of the step sizes was recorded.

Figure 35 summarizes the results with equal multiples of bias and pulse current. If M denotes the step multiple (that is, each time the bias or pulse current is changed, the change is M times larger than in the original experiment), then the "effective number of bits" shown in the figure is calculated as $8 - \log_2(M)$. Adequate control is still possible with the equivalent of five-bit DAC for bias and pulse current control. Eight-bit DAC easily implemented in aıı actual network, would provide a substantial margin of additional precision above that which would be strictly required.

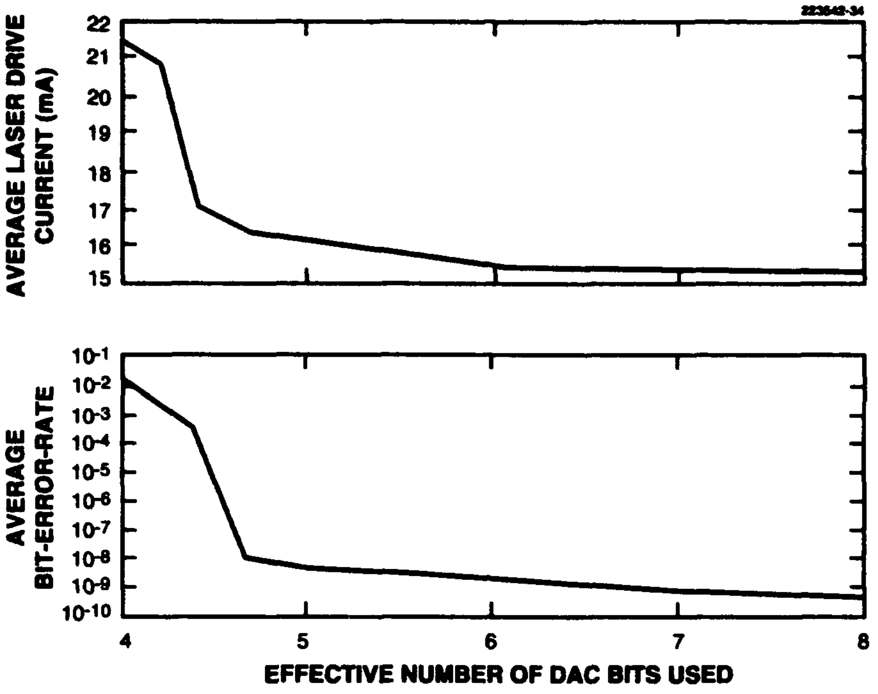

Figure 35. Laser control effectiveness vs DAC precision.

7.4.8 Feedback Stability

Because a feedback control system is proposed here, the stability of the control loop is an important issue.

The open-loop response of the laser-drive/optical-link system is complex. While the BER vs optical power relation given in Figure 8 is simple enough, the relation of bias current, to laser switching speed, to error rate, is not. An accurate mathematical model of the open-loop response function $\mathrm{BER}(I_{\mathrm{bias}}, I_{\mathrm{pulse}})$ would be difficult to find.

Such a formal model, and stability analysis based on it, might be worthwhile subjects for further research. In the absence of such a mathematical analysis, some observations about stability can still be made, based on the experimental data.

Looking at the bias- and pulse-current plots vs time from the experiments described in this section, one can see the system response to (essentially) step inputs of temperature and optical loss. Except for the coupled increase of bias and pulse currents (explained above), the control outputs exhibit little overshoot or ringing, indicating a reasonable stability.

Figure 35 gives a more direct measure of stability. Increasing the step size, in addition to simulating the effect of lower precision, also increased the feedback gain. Because the step size $M = 10$ (4.68 effective bits) still produced acceptable results, one can conclude that the system has a gain margin of at least a factor of 10, that is, 10 dB. (It is unclear if the poor performance at higher multiples was due to oscillation or to other factors.)

7.4.9 Achievable Error Rates

The particular experimental setup was limited to a slow response time because of its unwieldy interface to the data link analysis board. This limitation would not be present in an actual network, but the BER, as a feedback control variable, has an intrinsic time constant (let us call it τ) associated with each error-rate value:

$$\tau = \frac{1}{\text{Bit-Error-Rate} \times \text{Bit-Transmission-Rate}}$$

this time constant τ is the expected interarrival time between errors. For example, in the experiments the transmission rate was 1 Gbit/s, and the desired error rate was 2.5×10^{-10}, so τ was equal to 4 s. The time τ places a floor on the error rate that intelligent laser drive control can achieve, depending on the rate of change in the error signals (temperature, aging, optical loss, etc.) to be controlled.

The feedback system does not, in fact, receive an error rate as input. What it receives is error indications. The system must count the errors (call the count n) occurring a particular interval of time (call it T) to arrive at an estimate of the error rate: n/T.

The longer T is, the better the error-rate estimate will be. From empirical experience in designing the feedback system, acceptable operation will require T long enough so that at least two errors in time T are needed to declare the error rate too high. This implies that $T \geq \tau$. For the experiments shown above, $T = \tau = 4$ s. Any reduction in the error-rate goal would necessitate a proportional increase in T.

This time, however, limitation applies only a *lower* limit to the error-rate estimate. If an excessive number of errors occurs before the period T is complete, one can justifiably suspect that the error rate has become too high.

The experimental program, therefore, does exactly that. The error statistics are read every 1.3 s. If after three such readings there have been less than two errors, the error rate is judged to be acceptable. If after any of the three readings the error total reaches two or more, the error-rate estimation is aborted, and corrective feedback is performed.

Depending on how fast the error signals vary, intelligent laser drive control may be able to achieve error rates better than those in the experimental setup, but the time constants involved quickly become unwieldy as error-rate goals become more stringent. A strong error-control coding system, such as the one described in Section 6, will therefore be required.

These time constants are intrinsic to the error-based feedback system; as with many other theoretical limits they are always valid, but are not necessarily always relevant. Section 7.6 outlines one possible way that this theoretical limit might be sidestepped, to a large extent.

7.5 Benefits of Intelligent Laser Drive

In addition to the direct benefits of threshold current and optical loss variation control, intelligent laser drive control can provide two additional benefits: allowing unbalanced data transmission and permitting laser lifetime monitoring.

7.5.1 Unbalanced Data Transmission

Data communications link designers are usually not at liberty to make assumptions about the nature of the data to be transmitted across the link under design. In particular, they must assume that the data may be heavily unbalanced between the bit values "1" and "0." This can be a problem in an ac-coupled system (such as the one used for the experiments described above), because the data imbalance will produce a dc bias that will eventually be lost through the ac coupling. (This effect is called "baseline wander," because the baseline between "0" and "1" values at the receiver appears to wander.)

An ac-coupled system must therefore use a balanced "line code": a logical transformation of the incoming data stream (which may not have an equal number of zeroes and ones) into another, somewhat longer, data stream that is guaranteed to have equal or nearly equal numbers of ones and zeroes. The experimental setup used an ac-coupled data path, and a "4b5b" line code, that is, every four bits of data generated five bits on the transmission link. Table 2 shows this line code. Note that this code is not perfectly balanced: it could have a duty cycle imbalance of as much as 60%/40% for some input data streams. However the pseudorandom data used in the experiments, both before and after line coding, had equal numbers of zeroes and ones.

The use of a balanced line code can be a problem in systems employing the very wide channels envisioned in this report because line codes depend on a sequence of data being transmitted on each bit; word-by-word transmission might entail high overhead (up to 50%) in implementing the required line code, in addition to the coding and decoding circuitry. Unlike the EDC proposed in the previous section, balanced line codes can obviously not be made in systematic form (i.e., the original data accompanied by separate error check bits), so the line coding and decoding is part of the main thread of data transmission, as opposed to being an activity performed in parallel.

In some systems, therefore, it would be preferable to dispense with line coding, and design dc-coupled links. This could pose two difficulties in relation to laser drive control.

First, unbalanced data transmission makes the standard monitor-based laser feedback system (in Section 7.2.2) unusable. The monitor photodiode signal is not only a function of the laser drive and threshold currents, but also of the laser duty cycle. The feedback system implicitly assumes a fixed and predictable duty cycle (usually 50%) in the transmitted data. (Actually, there are

TABLE 2

4b5b Line Code

Data	Codeword	Data	Codeword
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
0111	01111	1111	11101

monitor feedback systems that compensate for varying duty cycle, but they are complex indeed: quite impractical for this application.) On the other hand, intelligent laser drive control is quite indifferent to duty cycle variations because it concerns itself only with the actual communication performance of the link.

Second, a dc-coupled link will be significantly more sensitive to laser threshold current variation. If, because of threshold current variation, the laser is never completely "off," there will be an abnormal dc bias on photodiode output. An ac-coupling circuit would block such a bias, but a dc-coupled circuit would have difficulty with it.

Such a dc-coupled link would exhibit an error-rate minimum at a particular bias current (or range of currents), with increasing error rates at both higher and lower currents. Simple analog feedback loops (or the unsophisticated program) detailed in this report would have great difficulty with this, but a well-designed intelligent laser drive control program should be able to track such a curve and keep the bias current at or near the error-rate minimum.

Unbalanced data transmission is an interesting design option, which intelligent laser drive control can make more feasible.

7.5.2 Laser Lifetime Monitoring

As indicated in Section 4.2.1, wear out manifests itself as a gradual increase in threshold current. A laser is usually declared worn out when a standard monitor-based control loop has to increase the drive current by a specified amount (usually 50 or 100% of the original drive current). The laser actually fails when the threshold current has increased so much that the laser driver can no longer deliver sufficient current.

A multiprocessor network designer has the luxury of being able to rely on software to manage the system operation. Because laser wear out is a very slow process, if one periodically records the average bias current required by the intelligent laser drive control algorithm, one should be able to perceive the wear-out trend in each laser and therefore predict when it will fail (that is, when it will demand more drive current than is available).

Given this, one should then be able to give the multiprocessor system's owner a reasonably reliable prediction of the network's future reliability and repair needs. For example: "If you repair these lasers at time X, the system will run normally. If you wait until time Y, the system will run at reduced capacity. With no repair the system will probably fail by time Z."

Because most normal-use systems have the option of preventive maintenance open to them, such a monitoring system could significantly increase overall system reliability.

7.6 System Implementation

The intelligent laser drive control system is considerably simpler to implement than a monitor-based feedback system because it requires only one or two simple DACs in place of a monitor photodiode and analog feedback loop.

Compared to a fixed drive system, intelligent control is undeniably more complex, requiring, in addition to the DACs, an adjustable-level driver the fixed drive system has no need for. As demonstrated, however, the DACs are eminently suited to VLSI implementation. Pulse current adjustment might or might not be easily added to a fixed-level laser driver, but bias current adjustment almost certainly could, because it can be applied directly to the laser (with suitable provisions for high-frequency decoupling).

An important aspect to realize is that the system need not run all its links under the full feedback regime at all times. It could instead periodically run the feedback algorithm on each link, to characterize it and check for drift. The link would then be fixed at a known (or predicted) low-error operating point, to be changed only when the link is again rechecked, or if it develops excessive errors.

If only one bit at a time is adjusted, the consequences of a momentarily high error rate are much reduced: if only one bit is affected, the chances of an undetected (multibit) error are still extremely remote. The only consequence would be a drop in link performance, as more retransmissions were required.

The scheduling of such adjustment cycles would depend entirely on how quickly the system characteristics were likely to change. A system might schedule quite frequent adjustments during initial warm-up and gradually taper off as the system reached a steady operating state. Such scheduling would, in effect, be controlled by a higher-level feedback loop, residing entirely in software. Such dynamic adjustment scheduling is an example of the type of flexibility possible in the feedback system because of the intelligent control available to it.

By such techniques, one may be able to sidestep the intrinsic time constants mentioned in Section 7.4.9 and have error rates (except during adjustment cycles) close to the minimum practicable.

7.7 Summary

Intelligent laser drive control seems a viable strategy for multiprocessor optical networks. It can

- Control for temperature-based threshold current variation
- Presumably control for age-based threshold current variation as well
- Detect optical path degradation and restore operation if enough capability remains
- Allow software-based monitoring of optical-link performance
- Make analysis of long-term trends (such as laser aging) possible

An experimental intelligent laser drive control system has been implemented and tested. It performs well under temperature and optical-loss variation. With pulse current fixed, it performs well under temperature variation, but not optical loss. Eight-bit DAC were used, but five bits would have sufficed. Detailed stability analysis was not performed, but the system's gain margin was found to be at least 10 dB.

An error rate around 10^{-9} was achieved and significantly better levels should not be expected on an individual link basis, due to the inherent time scale of the error process. However, overall error performance might be made considerably better by only periodically checking each link for drift and otherwise operating the links at a low-error operating point.

8. REDUNDANT SPARING

This section considers and evaluates a conceptually simple solution for hard failure problems: provision of redundant spares to be used in place of failed channels. It shows how implementation of such a scheme in the type of network described in Section 5 can be achieved via a reconfiguration switch design that seems well suited to VLSI implementation.

Using the failure model developed in Section 4.3.4, the efficacy of the redundant sparing approach for various network sizes is then analyzed. Additional approaches to the hard failure problem are discussed in Section 9.

The approach first deals with acceptable failure levels, which apply to both this section and the next one.

8.1 Acceptable Failure Levels

The question of how many hard failures can be tolerated is rather different from the corresponding question for transient errors because hard errors, while quite inconvenient, are unlikely to occur unnoticed. (See Section 6.3 for a discussion of error detection and diagnosis.)

The main criteria determining the acceptable hard failure rate in a given situation are:

- The tolerance for occasional system unavailability
- The cost and feasibility of repair

Requirements for ultrareliable systems are generally due to one or both of these criteria. For example, an aircraft control computer system can tolerate only a very low failure rate, due to the fact that even occasional unavailability is not tolerable. An example of the other criterion is an undersea fiber-optic transmission cable, which can tolerate very few failures because of the infeasibility of repair.

Such extreme cases have been the driving force for most fault-tolerant system design. A more ordinary, less demanding environment will be assumed, where a low failure rate is desired simply to reduce the inconvenience and expense of frequent downtime and repair. Therefore, the criterion for judging the efficacy of fault-tolerance schemes will be the median time before a system failure requiring repair.

8.2 Redundant Sparing

Redundant sparing is a standard approach to hard failure problems [43]. As the name implies, redundant sparing is the provision of redundant spare optical links, with the ability of the network to switch in a spare link in lieu of a failed one.

As indicated in Section 6.3, hard failures can be easily detected by the error control coding circuitry. Because hard failures occur quite rarely (from a processor-cycle perspective), special hardware need not be provided to control hard-failure diagnosis and network reconfiguration: such tasks can be performed in software, with negligible impact on system performance.

8.3 Redundant Channel Switch

It is easy to see how one might provide spare optical links in a channel, but it is not as obvious how to use these links when they are needed. A telecommunications link where data words are transmitted serially on a single optical link is a simple case: provide one link and m spares and connect the data stream to one of $m + 1$ possible links, depending on which links have failed.

A wide parallel data channel is more complex. It would be far too expensive to provide multiple spares for each bit in the channel, so for an n-bit-wide channel one would provide n regular links and m spares. The problem then becomes one of connecting the data to n of $n + m$ links, depending on which links have failed. This must also be done with the minimum possible switching latency.

This report proposes the use of a substitution switch, capable of switching out failed links and substituting spare links in their place, which seems well suited to high-speed implementation in VLSI.

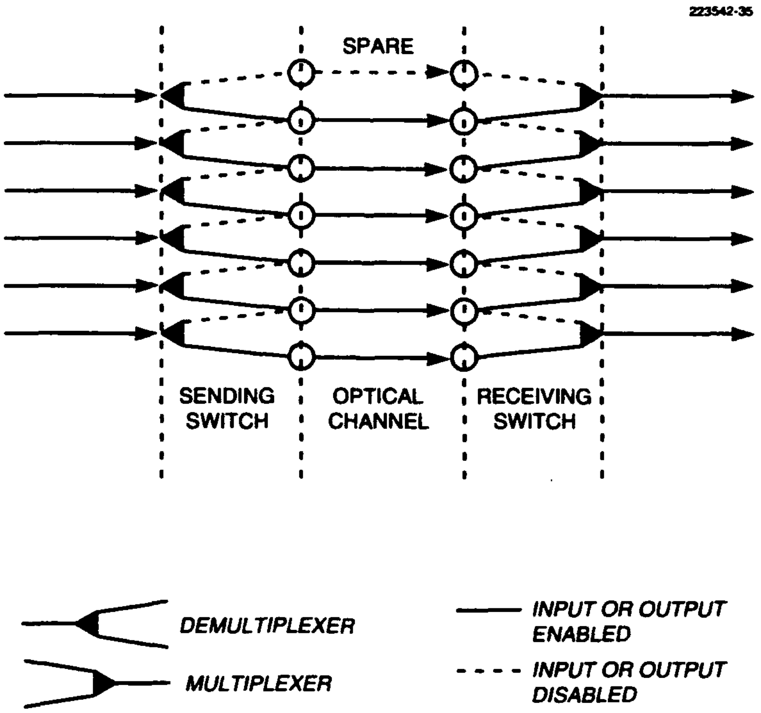

Figure 36. Substitution switch example.

To illustrate the point, let us consider a six-bit channel with one additional spare link. Figure 36 shows a pair of substitution switches (one for sending, and a mirror-image of it for receiving), which will enable us to use the spare link to replace a failed one. Each switch contains one multiplexer (or demultiplexer) per bit. The figure shows the switch in the normal (prefailure) state, with the spare link unused. (The choice of which link is spare is actually an arbitrary one and makes no difference to the fault-tolerance of the switch.)

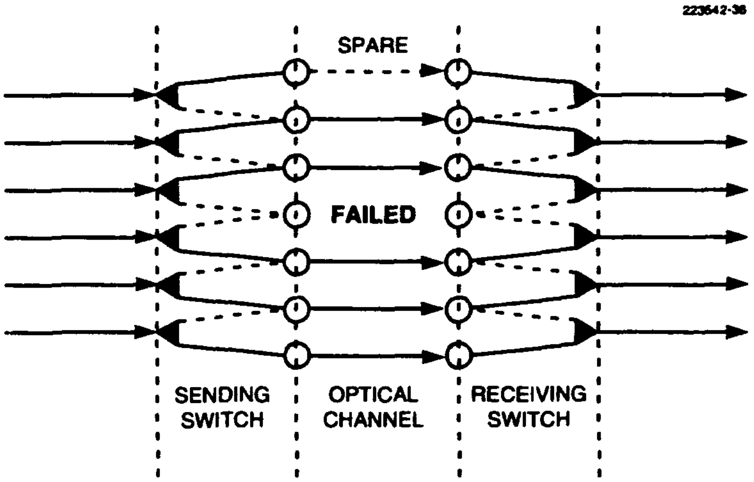

Figure 37. Substitution for one failed link.

Figure 37 shows the same channel after one of the links has failed. Note that upon reconfiguration of the substitution switches (the one switch configured as the mirror-image of the other), the channel can continue operation unimpeded, with the failed link removed from the channel.

The switches in Figures 36 and 37 are adequate for dealing with one error per channel. They can be extended to any number of data bits, but will accommodate only one spare link.

Fortunately, the same concept can be extended to handle additional spare links. Figure 38 shows the same six-bit channel as before, but now with three spare links. There are two mirror-image substitution switches, as before, but now there are two levels of multiplexing/demultiplexing.

Figure 39 shows the channel after three of the links have failed. As before, reconfiguration of the substitution switches (in mirror-image fashion) allows the channel to continue normal operation.

This concept can be extended as far as desired. Figure 40 shows a four-stage substitution switch, which can handle 15 spare links. (To save space, only the sending switch is shown: the receiving switch is, as always, a mirror-image of it.) Figure 41 shows the channel after 15 failures

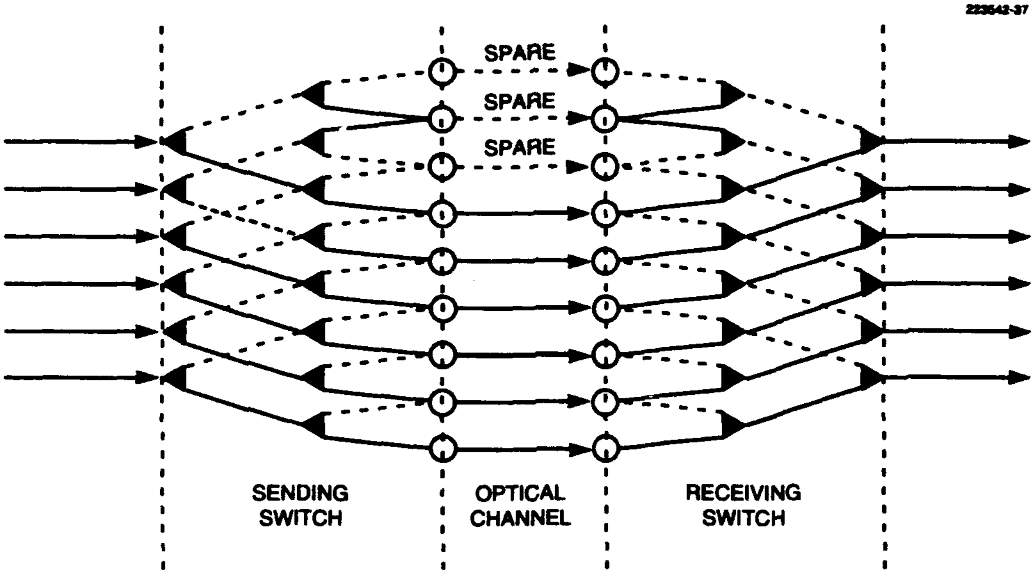

Figure 38. Two-stage substitution switch.

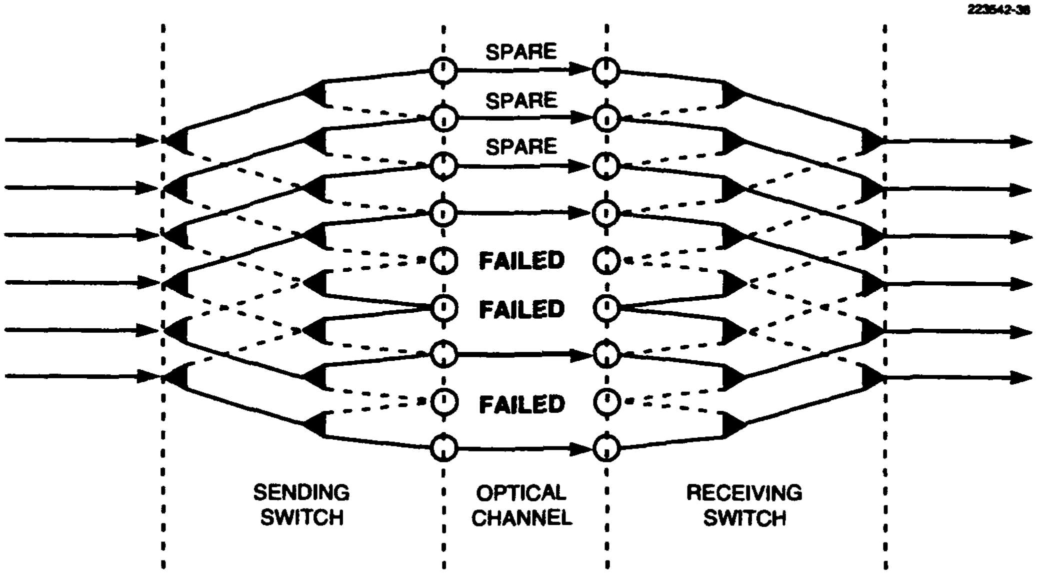

Figure 39. Substitution for three failed links.

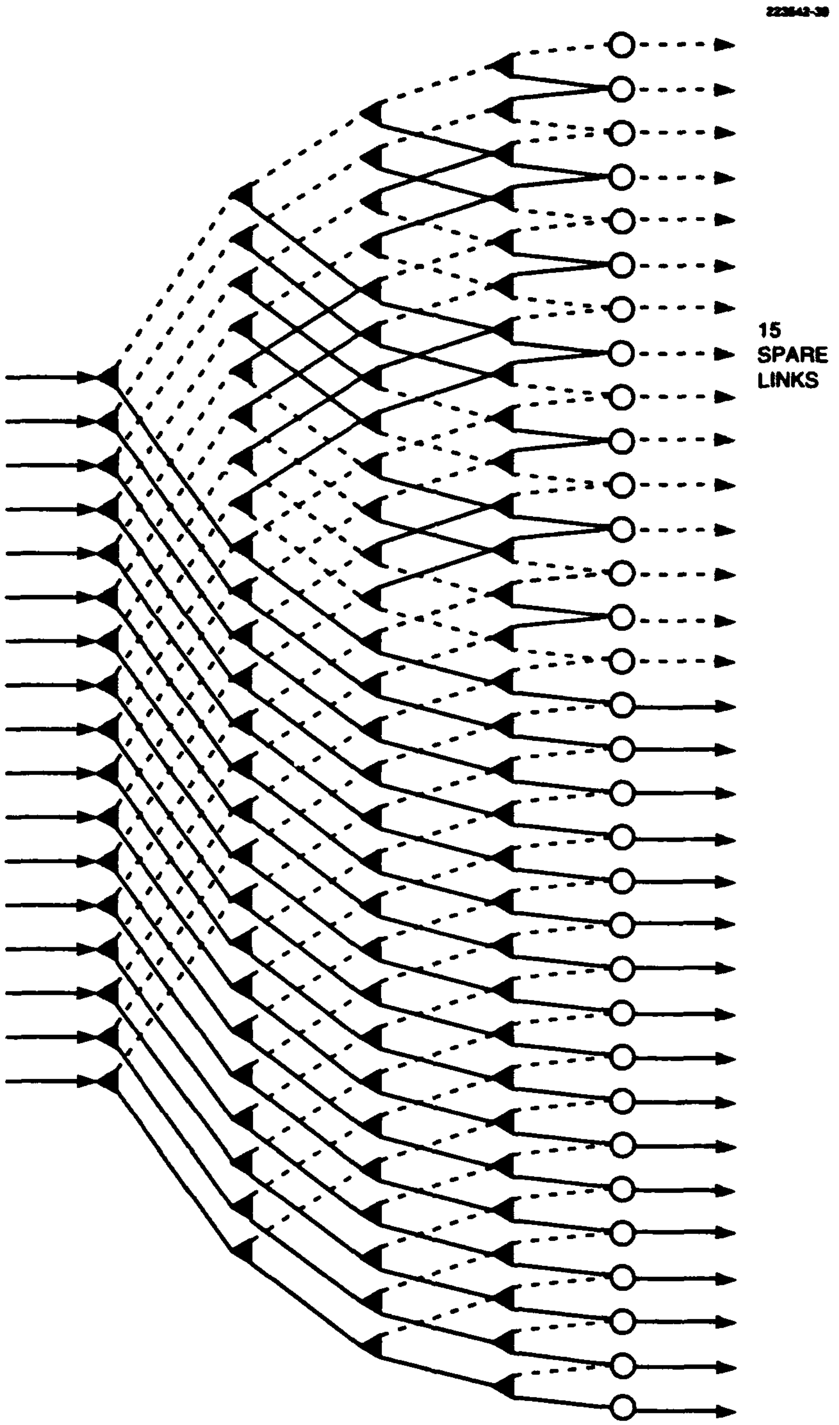

Figure 40. Four-stage substitution switch.

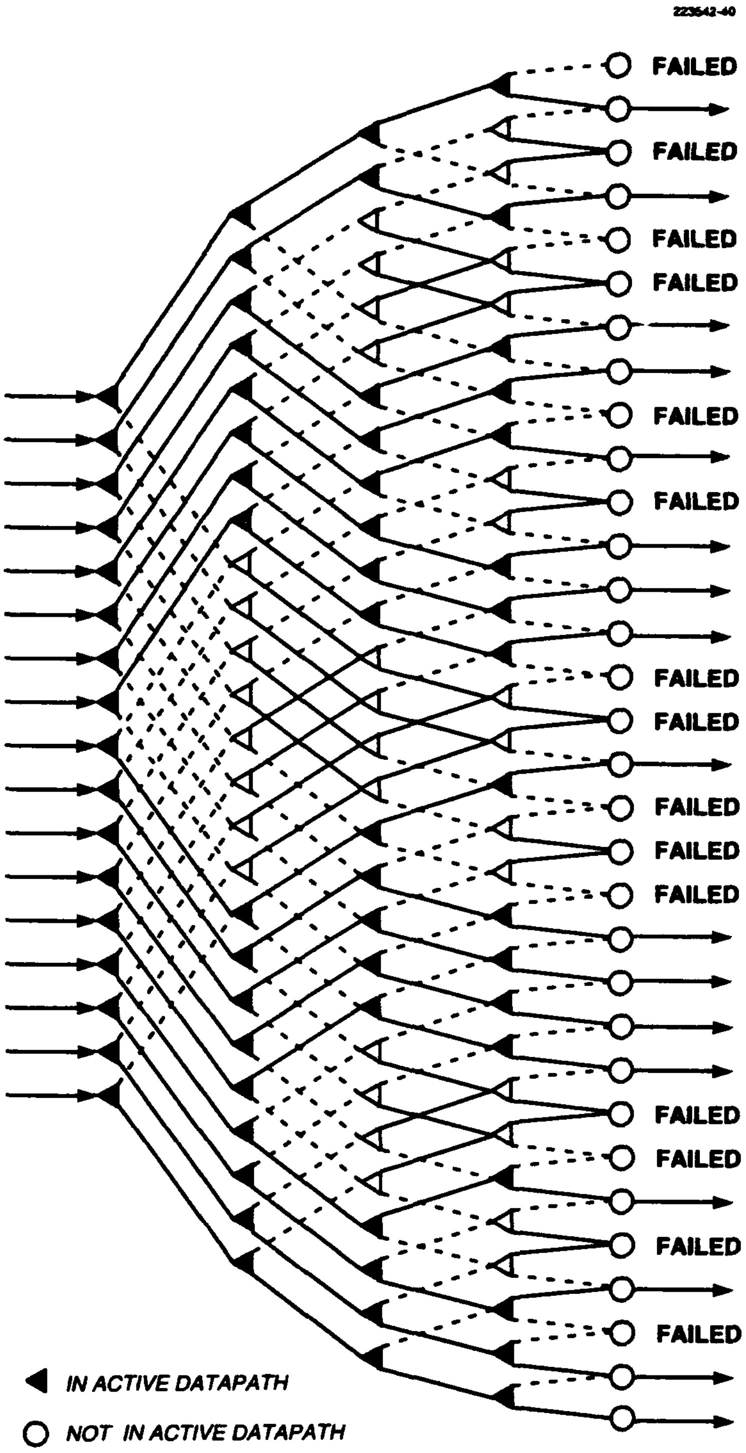

Figure 41. Substitution for 15 failed links.

have occurred: again, the reconfigured switch allows full operation to resume. (Because the switch configuration gets rather complex at this level; Figure 41 distinguishes between the parts of the switch in active use vs those not actually connected for data transmission.)

In fact, an n-level substitution switch of this design can handle up to $2^n - 1$ spare links, and can reconfigure the channel for full operation after any set of $2^n - 1$ links have failed. Let us consider why this is so and what is the proper algorithm for switch reconfiguration given an arbitrary set of failures.

8.3.1 Single-Stage Substitution Switch Theory

Consider the one-stage substitution switch shown in Figure 37. It should be clear that this switch can be reconfigured to eliminate any one failed link. But suppose that there were more than one failure: what could it do? While the one-stage switch can not restore full channel operation when faced with more than one failure, it *can* map the failures to different positions, so that they are more contiguous than before. In particular, given m failed links, the one-stage switch can map every failure into $\lfloor m/2 \rfloor$ contiguous pairs (except that a "pair" at the top of the switch might have only one failure).

Recall that the sending demultiplexers and the receiving multiplexers are configured as mirror images of one another. For convenience, let us call each pair of demultiplexer and corresponding multiplexer a "switchpoint" and call its two switched outputs or inputs "ports."

Let us devise an algorithm to configure the one-stage switch. Starting with the bottom switchpoint, (actually, starting from one end and defining that end as the bottom), a switchpoint is in the "up" state if its upper port is active, and is in the "down" state if its lower port is active. The algorithm is:

- Start with all the switchpoints in the down state
- Scan the links one at a time, from bottom to top. At each link, if the link has failed, invert the state of all switchpoints above that link

[Another way of putting it: if there is an odd number of failed links below a switchpoint, put it in the "up" state; otherwise put it in the "down" state.]

This algorithm assures that all failures (except, perhaps, a failure in the top link) will be mapped into contiguous pairs. To see how this works, consider Figure 42. This shows the two possible switchpoint configurations around a failed link, because the switchpoints above and below it must be in opposite states. In the first case, the failed link has been eliminated from the network; if this is the only failure, then all the switchpoints will be connected to working links, as in Figure 37. Figure 43 shows how two separate failures are mapped to contiguous positions.

The second case in Figure 42 shows what happens if there is an odd number of failures below the one shown. In this case two switchpoints point at the failure and are therefore not usable. One can think of this as effectively moving the "odd" failure from down below to be paired with the

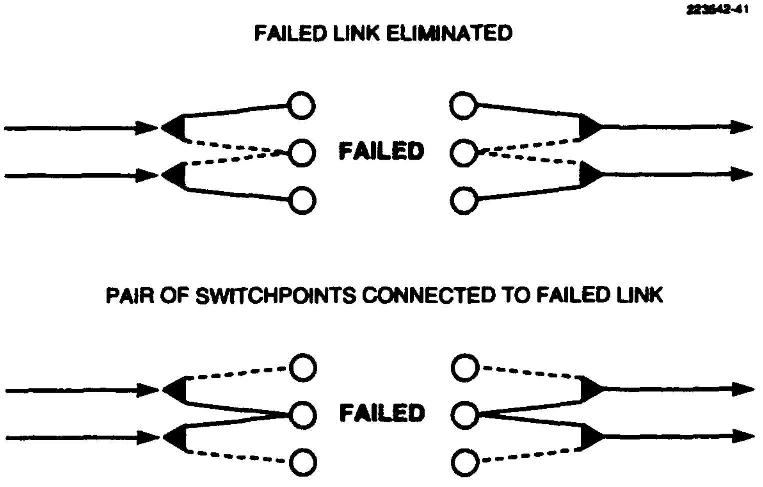

Figure 42. Substitution switch algorithm examples.

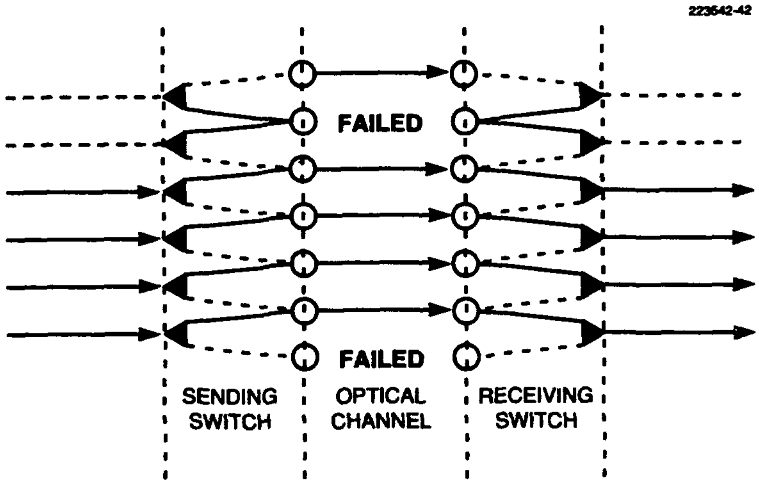

Figure 43. Failure pair mapping.

failure here. Of course, if there were a failure in the top link with an odd number of failures below, then there would be only a single switchpoint to connect to it, so that is the one exception to the rule.

With this algorithm, an isolated link failure will be shifted up until the end of the switch is reached or until another failure is encountered. In the latter case, both failures, and any additional contiguous failures, are then propagated to the switchpoints (that is, the switchpoints are declared "dead"), and the process then repeats. This cannot, therefore, produce isolated "dead" switchpoints (except for the top switchpoint, if the top link has failed).

8.3.2 Multistage Substitution Switch Theory

Given this result, how can the failed channels be eliminated, as opposed to merely being paired? Consider the second-level substitution switch shown in Figure 38. Note that the second level switchpoints are interleaved, spanning *two* positions, instead of one. This can be used to eliminate a *pair* of failed switchpoints, just as the first-level switch could eliminate a single failed link.

To configure the second-level switchpoints, one uses precisely the same algorithm as one used for the first level, except that one scans for switchpoint failure-pairs instead of failed links. So, one first configures the first-level switchpoints. The second-level switchpoints are then configured as follows:

- Start with all the second-level switchpoints in the "down" state
- Scan the first-level failure pairs one at a time, from bottom to top. At each pair, invert the state of all second-level switchpoints above (the center of) the pair

Figure 44 shows the two possible configurations around a failure pair. As with the first-level switches, there are only two possible configurations. The first case shows how a failure pair can be eliminated. The second case shows that when the failures can not be eliminated they will be grouped into quadruplets. If there is a first-level "truncated pair" (that is, a singleton) at the top of the switch, it is handled like any other pair, so there may be a second-level "truncated quadruplet" at the top of the switch also.

If there are p first-level failure pairs, then there will be $\lfloor p/2 \rfloor$ second-level failure quadruplets. With m failed links, $p = \lfloor m/2 \rfloor$, so if $m \leq 3$, the two-level substitution switch will eliminate all the failures.

The switch to n levels can be generalized, with the n-th level switch spanning 2^{n-1} positions and grouping failures into 2^n-tuples. The configuration algorithm given applies to all levels. Because level ℓ can eliminate $2^{\ell-1}$ failures, the multilevel switch can eliminate $\sum_1^n 2^{\ell-1} = 2^n - 1$ failures. A switch to accommodate s spare channels would therefore require $\lceil \log_2(s+1) \rceil$ levels.

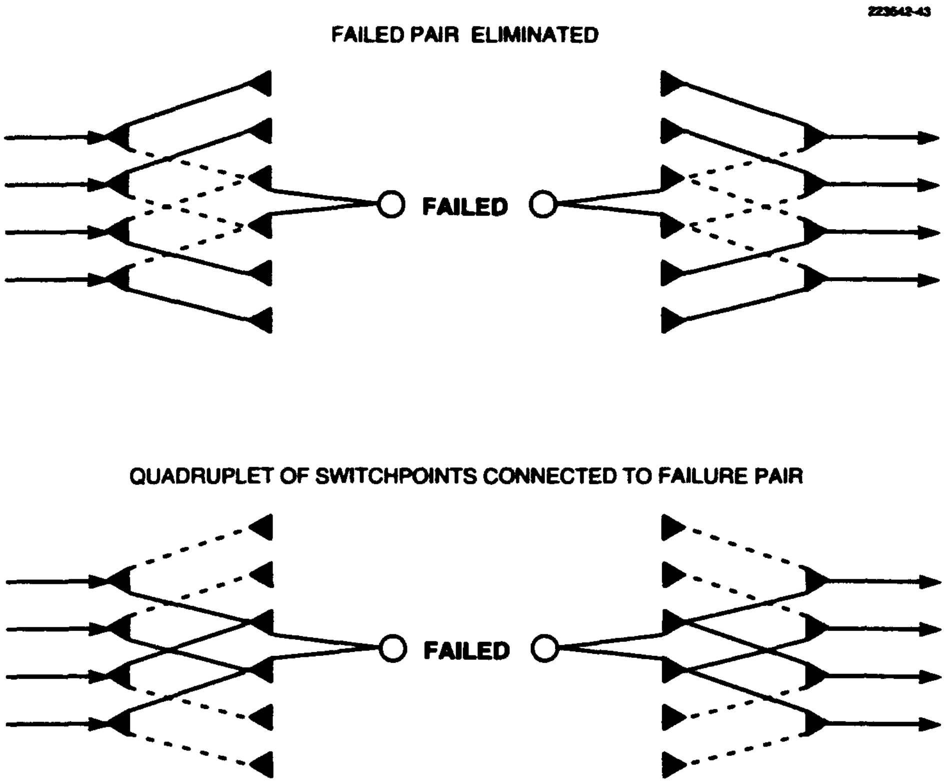

Figure 44. Second-level substitution algorithm examples.

8.3.3 Substitution Switch Implementation

From Figure 41, the substitution switch can appear quite complex. However, the previous section shows that its configuration is actually rather simple. Also, as mentioned in the beginning of the section, new failures are exceedingly rare events on a processor-cycle time scale, so the switch reconfiguration can be handled in software.

While a multistage reconfiguration switch would be complex, it has a number of attributes that would make it well suited for implementation in VLSI:

- It is a quite regular structure with simple control, making the design task straightforward
- Each output drives no more than two input loads, making a high-speed implementation more feasible

- Because reconfiguration would be a rare event, the multiplexer and demultiplexer[5] circuits could be designed, as much as possible, to trade off reconfiguration speed in favor of fast data propagation time

Because the same signals are involved, one VLSI circuit could implement both the substitution switch and the EDC/decoding functions. A monolithic VLSI solution might achieve quite fast switch propagation times because it would avoid chip-to-chip communication delays within the switch. It might also be practical to include all or part of the laser drivers or photodiode amplifiers in such a VLSI circuit.

8.4 Redundant Sparing Lifetime

Let us next consider the likely operating lifetime of a network such as the one described in Section 5, with various levels of redundant sparing (using an appropriate substitution switch) in each optical channel. Let us consider the redundant-sparing network to have failed (i.e., to require repair) when any one of its channels has experienced more failures than it has spare iinks (for example, eight failures in a channel with seven spare links).

Looking at Figure 15, at least until around $t = 8$ years, laser failure can be approximated as a Poisson process, with arrival rate $\lambda \approx 0.009$/year. If each channel is n bits wide, with m spare links (and a substitution switch of $\lceil \log_2(m+1) \rceil$ levels) provided on each channel, and assuming that the spare links suffer no failures when not in use, then the probability that a particular channel has failed by time t can be approximated by the probability that more than m arrivals of a Poisson process of rate $n\lambda$ will occur in a time interval of length t (let $p_c(t)$ denote this probability). Therefore

$$p_c(t) = \sum_{k=m+1}^{\infty} \frac{(\lambda t)^k}{k!} e^{-\lambda t} = P(m+1, n\lambda t) \tag{12}$$

where $P(k, x)$ denotes the incomplete gamma function[44].

If the system has a total of C such channels with independent failures, then the probability that at least one of them has failed by time t (let $p_s(t)$ denote this probability) is given by

$$p_s(t) = 1 - (1 - p_c(t))^C \tag{13}$$

[5] Actually, one might well use just multiplexers (or demultiplexers) exclusively, in both sending and receiving switches, if that were an advantageous design; one can quite easily transform the circuits to use only one or the other. A design is presented with both multiplexers and demultiplexers because it preserves the symmetry of the switch and thereby makes its operation easier to explain.

If, as mentioned in Section 8.1, there is concern regarding the median time before a failure requiring repair, then there is an interest in finding t_m such that $p_s(t_m) = \frac{1}{2}$, so

$$\begin{aligned} p_s(t_m) &= 1-(1-p_c(t_m))^C = \frac{1}{2} \\ (1-p_c(t_m))^C &= \frac{1}{2} \\ p_c(t_m) &= 1-2^{-1/C} \\ P(m+1, n\lambda t_m) &= 1-2^{-1/C} \qquad (14) \end{aligned}$$

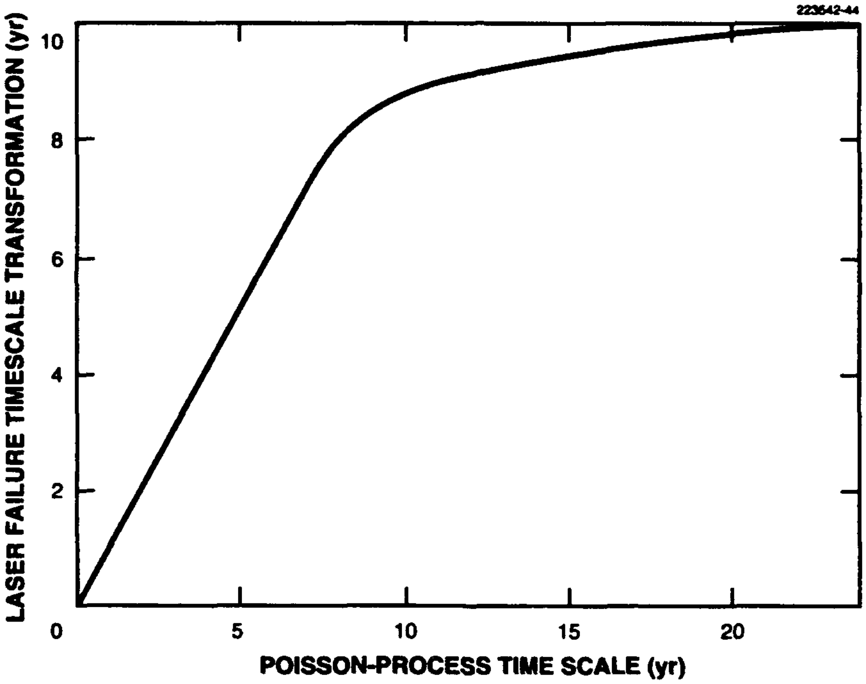

Figure 45. Laser failure time scale transformation.

Actually, the usefulness of the Poisson-process approximation can be extended beyond $t = 8$ years, by stretching out the time axis to make the laser failure probability continue to match that of a Poisson process. This is a sort of Procrustean bed, where the failure model is stretched and squeezed until it fits a desired form. Figure 45 shows a time transformation that makes a Poisson process of $\lambda = 0.009$/year match the failure model given in Figure 15.

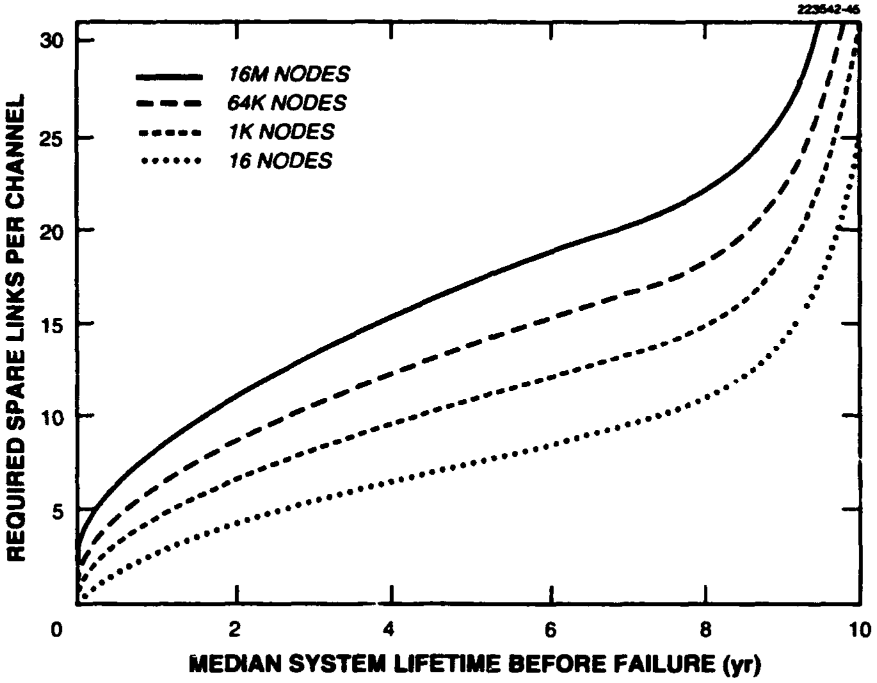

Figure 46. Spares/channel m vs median system lifetime t_m.

Using this method, several numerical solutions of Equation (14) have been derived, each for networks of different sizes. Based on the the network described in Section 5, the charnel width $n = 74$ is set, and the number of channels $C = 4N$, where N denotes the number of nodes in the network.

Figure 46 shows the minimum sparing level m required to achieve different values of the median system lifetime t_m, for mesh networks of 16, 1024, 65,536, and 16,777,216 nodes.

Redundant sparing can handle the random-failure part of the laser lifetime curve, but the required level of spares rapidly becomes impractical to sustain continued operation as the iasers enter the wear-out regime ($t \approx 8 - 10$ years).

[However, these calculations do not include any effect from a laser monitoring and replacement policy of the type suggested in Section 7.5.2. Such a policy might well allow continued operation at times when the system would otherwise be dominated by wear-out failures.]

8.5 Summary

Redundant sparing could suffice to control random laser failures and early wear-out failures, if sufficient spares were provided. A multilevel substitution switch, possibly implemented in a VLSI chip, could make this feasible.

Bandwidth fallback, to be discussed in the next section, is an interesting improvement to this approach. In fact, it can let us omit the spare channels and still offer performance only slightly less than redundant sparing.

9. BANDWIDTH FALLBACK

This section discusses two strategies to supplement redundancy; in particular, the strategies specify courses of action to take when redundant spares run out. By doing this, the level of redundant sparing that was needed in the previous section can be reduced (or eliminated).

The two options discussed here are detour routing and bandwidth fallback. Both seek to continue network operation (at a reduced rate) after failures for which redundancy is inadequate: these are therefore graceful degradation strategies.

9.1 Detour Routing

Detour routing exploits the wormhole-routing concept of virtual channels to produce new (virtual) channels to replace failed ones. Virtual channels is an idea that was actually conceived to enhance the performance of wormhole-routed mesh networks (see Section 5.3 for an explanation of wormhole routing). The basic idea is to take each *physical* channel in a network and establish a number of *virtual* channels within it, that is, to handle the network routing and scheduling as if there were several independent virtual channels wherever there is an actual physical channel. Figure 47 depicts this idea.

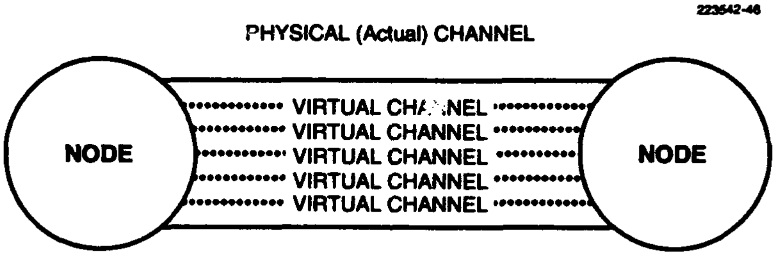

Figure 47. Virtual channels.

Virtual channels are implemented by time-division multiplexing the underlying channel, when required. The additional overhead imposed by a virtual-channels strategy is the additional control and buffering for each additional (virtual) channel created. Wormhole routing therefore makes virtual channels more practical because each channel in a wormhole-routed network requires only minimal buffering.

The use of virtual channels can provide significant performance benefits in a wormhole-routed network as Dally has shown [45], by handling congestion more effectively. However, as Kung suggests [46], it can also provide a measure of fault tolerance.

It can do this by allowing us to construct detour routes, exploiting the existence of redundant paths in the system network topology. The prototype mesh network obviously has such redundancy, as do many other possible topologies (see Section 3.7).

Detour routing can provide an additional backup for a channel that fails because it runs out of spare links. Traffic that would travel the failed link is rerouted on a virtual link created by time-multiplexing a redundant path in the network.

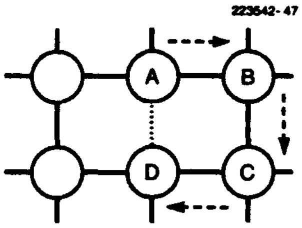

Figure 48. Detour routing example.

As an example, consider Figure 48. If the channel from A to D fails (runs out of spares), virtual channels can be used to replace it. This is done by time-multiplexing the channels A-B, B-C, and C-D, and routing the A-D traffic through these three multiplexed channels. The new channel, so constructed, triples the latency of the original and must share the bandwidth of the multiplexed channels with with regular traffic (which will be impeded as well), but the network *does* function when so reconfigured, in a situation where straight redundant sparing would call for repair. When only small numbers of such detour channels are necessary, they will have minimal impact on system performance, while extending the system MTBF.

Note that the reconfiguration does not vitiate the argument (presented in Section 5) that the mesh network routing will not deadlock. Because the logical topology of the network has been preserved, the nondeadlock arguments still apply; some of the (logical) channels have been slowed down. Steps must be taken to ensure that the regular traffic cannot monopolize the virtual channels to the extent that detour traffic would be blocked: that *would* invalidate the no-deadlock assurance.

This detour routing fault tolerance approach effectively views the links on the alternative channels (such as the channels shown in Figure 48 as another class of spare links, which in this case are not redundant. Especially if a virtual channel scheme would be implemented in any case for performance reasons, the additional overhead in implementing this fault tolerance method is minimal, and it obviously has the potential to transform a must-repair-now situation into an almost-normal (or at worst a repair-soon) situation.

In Section 9.3, the detour routing, by softening the impact of having a channel run out of spares, will reduce the number of spares required in order to maintain a given level of reliability. The following sections consider another approach to reducing the amount of redundancy needed.

9.2 Bandwidth Fallback Concepts

Bandwidth fallback is a different approach from those previously mentioned, in that rather than trying to preserve a fixed network channel width, it seeks to use whatever channel width the network has to offer on the desired path.

Rather than having idle spare links (waiting for an active link to fail), or having entire channels declared dead because they are one or two bits too narrow, bandwidth fallback takes a more flexible policy: when using a given path through the network, set the data width to the width of the narrowest channel on the path.

9.2.1 Basic Operation

The basic idea of bandwidth fallback is one of graceful degradation: when the spare links (if any) within a data channel are exhausted by previous failures and a new failure occurs, one can reconfigure the channel to use only the remaining links, thereby providing a channel that can still operate, albeit with reduced bandwidth. This could be preferable to the alternative: declaring the channel to have failed, and therefore providing zero bandwidth.

Unfortunately, it would be rather difficult to implement bandwidth fallback in exactly this way, because one would have to repackage the data into words of arbitrary size, for example, converting 64-bit words into 51-bit words for transmission. This would require both sender and receiver to have at least a full-width barrel shifter and two full-width holding registers.

By placing a simple constraint on the set of output word widths, however, bandwidth fallback can be implemented in a switch requiring only very simple hardware.

Figure 49 shows the basic circuit for implementing bandwidth fallback. It requires one register and one multiplexer for each bit of a channel. For full bandwidth operation, the registers are bypassed and data transfer occurs normally. For reduced bandwidth operation, a limited set of fractional bandwidths are available: $1 - 2^{-n}$, that is,

$$\frac{1}{2}, \quad \frac{3}{4}, \quad \frac{7}{8}, \quad \frac{15}{16}, \quad \frac{31}{32}, \quad \ldots, \quad 1 \qquad (15)$$

The switch in Figure 49 can implement bandwidths of 1, 7/8, 3/4, and 1/2 times full bandwidth.

Figure 50 shows how the switch can be configured for these different fractions. The fraction 1/2 configuration may be readily understood from the figure: the bits are paired, and the paired bits are sent one after the other.

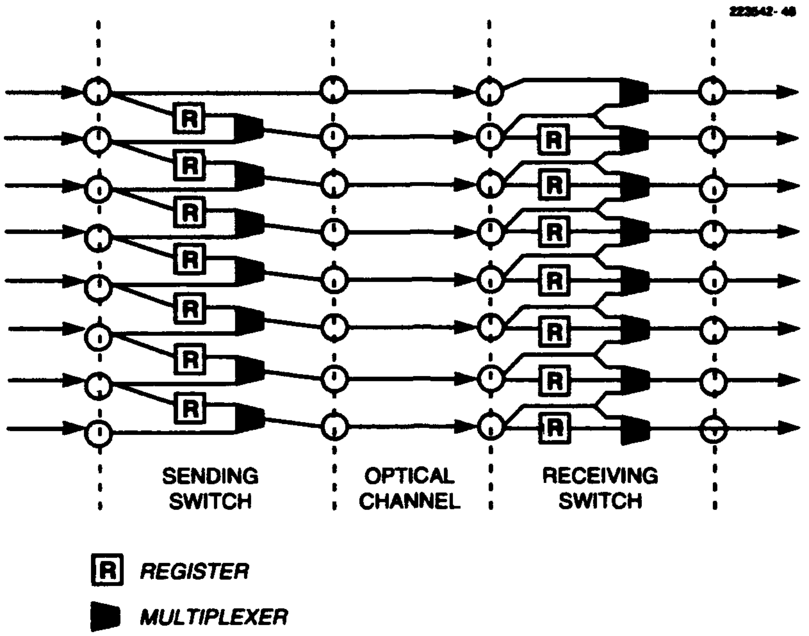

Figure 49. Bandwidth fallback switch.

Switch operation at the higher fractions requires rather more explanation. Looking at Figure 50, consider either of the (now disconnected) four-bit sections in the switch configured for 3/4 bandwidth. Table 3 illustrates the flow of data through such a 75%-bandwidth channel.

The actual operation of the switch is somewhat complex, but its basic concept is simple: for bandwidth $\frac{k-1}{k}$ operation, the switch accepts $k-1$ input words and passes $\frac{k-1}{k}$ of the bits in each one, storing $\frac{1}{k}$ of the bits in the registers. After $k-1$ words have been received, all the registers are full. The input to the switch is then suspended for one cycle and the register contents are output. The same cycle begins again, and continues as long as input is available. The receiver reverses the process using a similar switch. The same basic operation applies to all the possible fractions $(1-2^{-n})$ mentioned above.

While the switch control may be complex, it is fixed and predictable and can therefore be pipelined as much as necessary. The switch control program changes only when the bandwidth fallback fraction must be changed (that is, when too many additional links have failed).

9.2.2 Interaction with Substitution Switches

From Figure 50 some bits in the switch are expendable and some are not. For example, the top channel link is quite expendable, because losing it will only drop the bandwidth to 7/8, while the link next to it is essential, because the fallback switch cannot work around it.

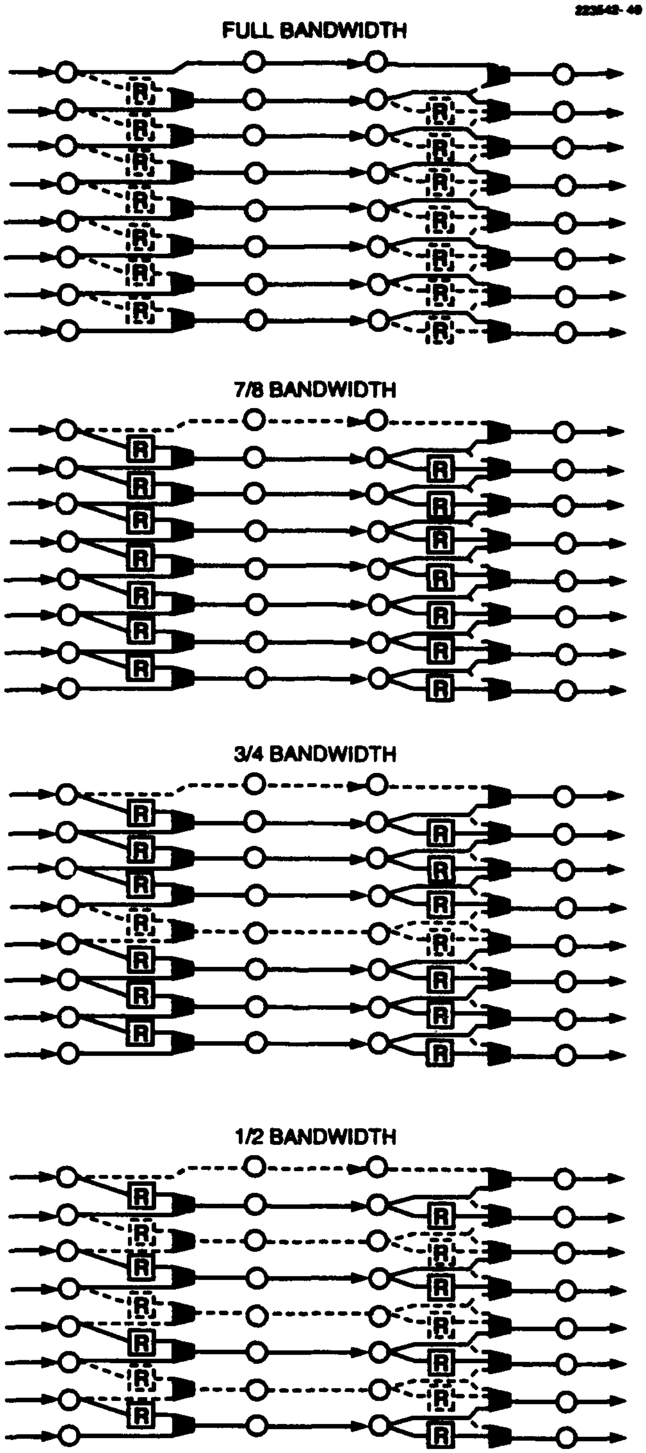

Figure 50. Switch configuration for various fractions of bandwidth.

TABLE 3

Data Transmission Through a 75%-Bandwidth Channel

From Sender				Data Held			75% Channel			Data Held			To Receiver			
A3	A2	A1	A0	-	-	-	A2	A1	A0	-	-	-	-	-	-	-
B3	B2	B1	B0	A3	-	-	A3	B1	B0	A2	A1	A0	A3	A2	A1	A0
C3	C2	C1	C0	B3	B2	-	B3	B2	C0	-	B1	B0	B3	B2	B1	B0
(idle)				C3	C2	C1	C3	C2	C1	-	-	C0	C3	C2	C1	C0
D3	D2	D1	D0	-	-	-	D2	D1	D0	-	-	-	(idle)			
E3	E2	E1	E0	D3	-	-	D3	E1	E0	D2	D1	D0	D3	D2	D1	D0
	⋮				⋮			⋮			⋮			⋮		

This may seem to be a problem, but there is a simple solution: ensure that the expendable bits are routed to the outside channels of the redundancy substitution switch and the essential channels are routed to the inside. If the switch is not completely full (that is, it has less than $2^{\ell} - 1$ spares, where ℓ is the number of levels), then the bandwidth fallback will essentially create new spare channels from the bandwidth it stops using.

Figure 51 demonstrates this. The first diagram shows that although there is a two-level substitution switch (that can accommodate three spares), there is only one spare link. The second diagram shows that three links have failed, but the channel can still operate because the bandwidth fallback scheme has stopped using two of the bits in the channel, which effectively makes three spare links available, where before there was only one.

Therefore, bandwidth fallback allows us to view the regular communication links as if they were redundant spares, but unlike redundant spares, can make full use of them can be made both before and after failures occur.

9.3 Bandwidth Fallback Simulations

Simulation programs have been written to estimate the impact of using detour routing and bandwidth fallback in a multiprocessor network of the type described in Section 5. Details of the simulation are given in Appendix B, while results are given here.

Results of simulating the operation of a 10 × 10 mesh network with varying levels of redundant sparing are shown here. Figures 52 and 53 show how performance evolves over time for both detour routing and bandwidth fallback (using the failure model depicted in Figure 15) with 5 and 10 spare links/channel, respectively. (For these bandwidth fallback simulations, the report assumes the use of five-level substitution switches.)

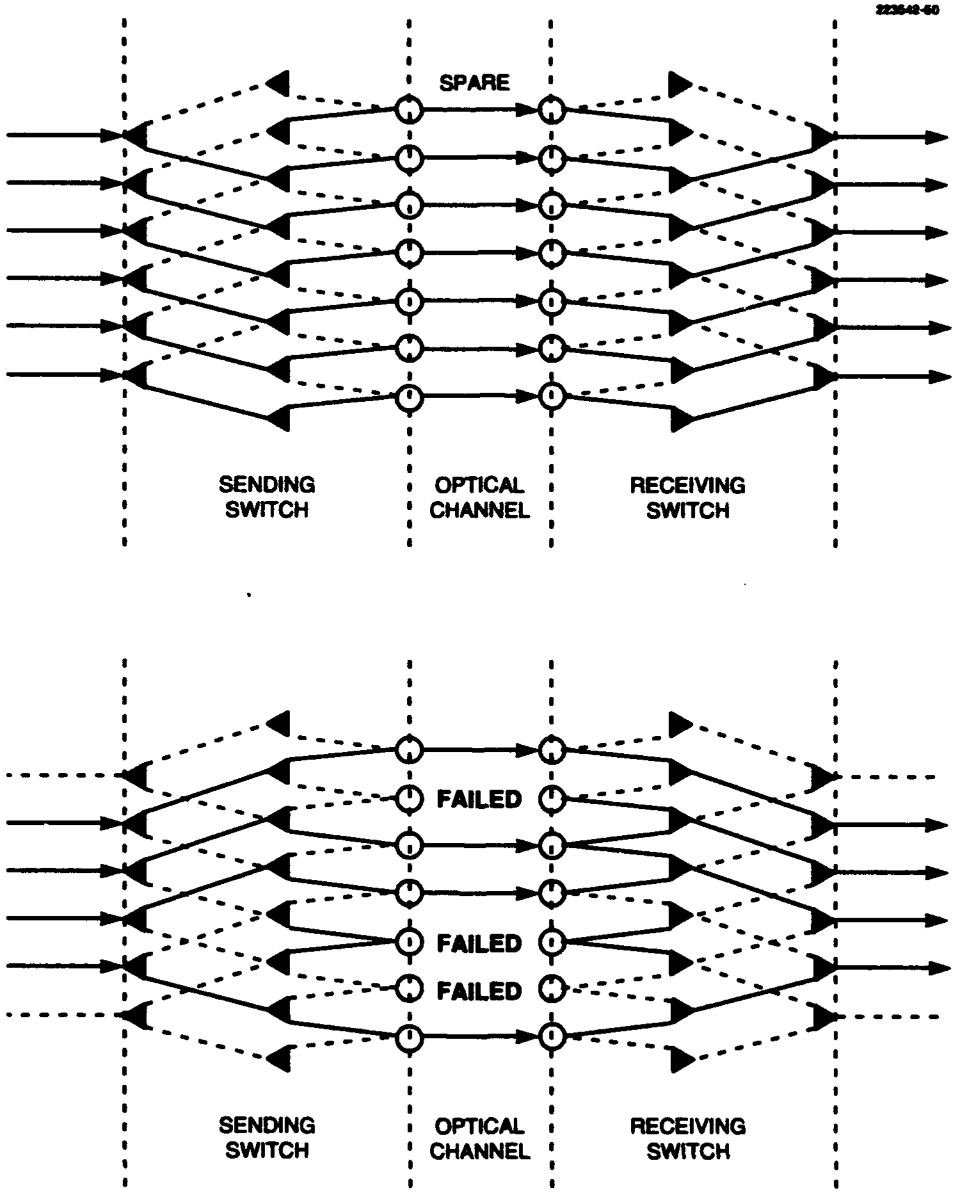

Figure 51. Substitution switch with bandwidth fallback.

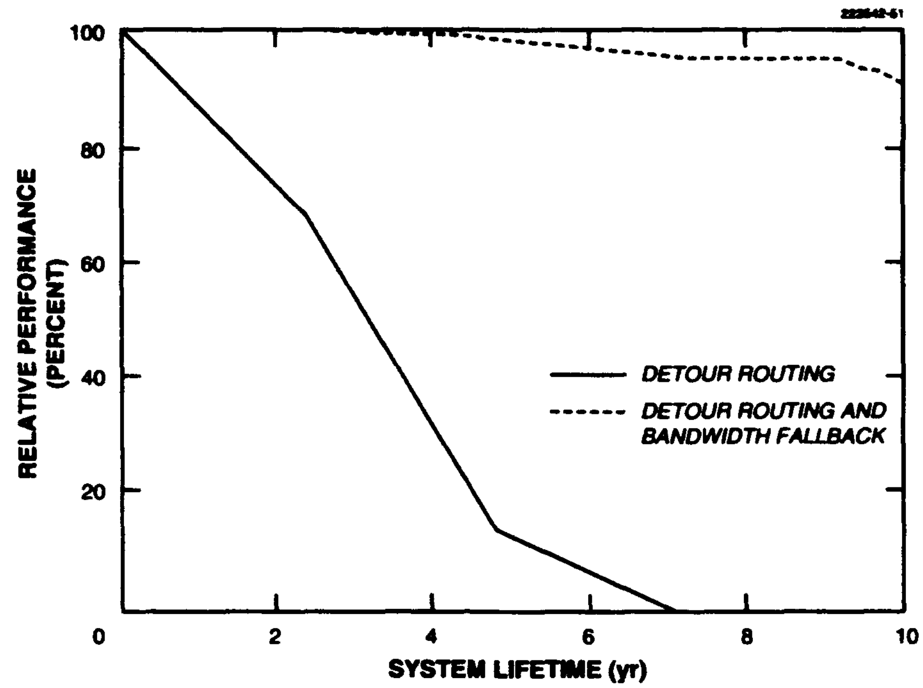

Figure 52. Performance vs time with 5 spares/channel.

The detour routing approach is acceptable, especially with the higher level of sparing, but the bandwidth fallback results are almost perfect. (Note that straight redundant sparing would do worse than detour routing because it would just give up completely when a channel ran out of nodes.)

Let us see if the limits of this phenomenally good bandwidth fallback performance can be found. Figure 54 shows a configuration where even with detour routing the system fails almost immediately: *no spare links at all.* Bandwidth fallback, on the other hand, enables the system to continue operation smoothly, with only modest performance loss.

9.4 Conclusions

This report has presented two failure-tolerance schemes: detour routing and bandwidth fallback. When implemented together, they provide superb error tolerance, with performance loss of only a few percent. If adequate substitution switching is provided, bandwidth fallback can provide continued system operation well into the laser wear-out region of system lifetime, while allowing us to completely dispense with the provision of any spare links.

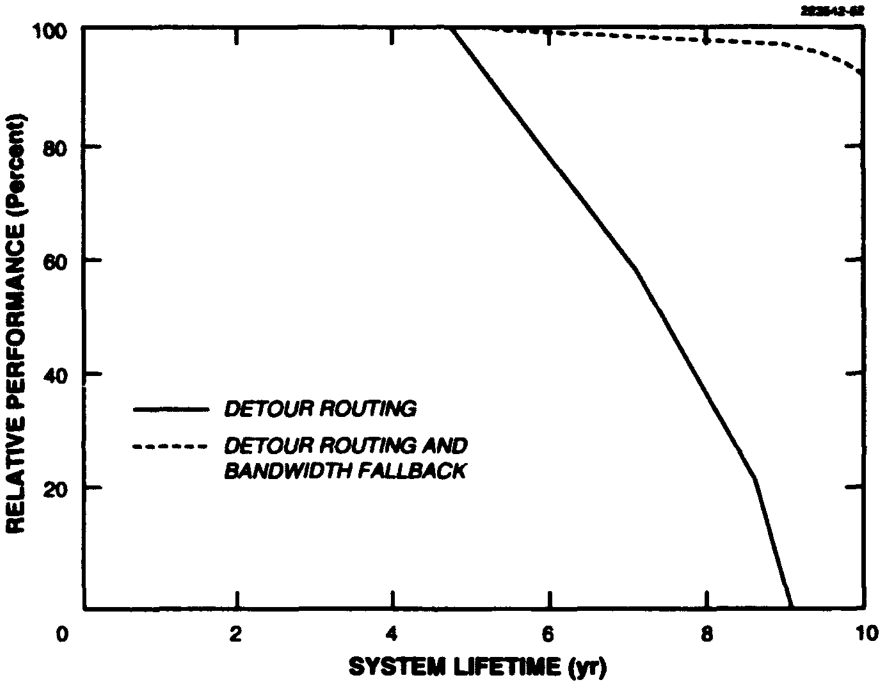

Figure 53. Performance time with 10 spares/channel.

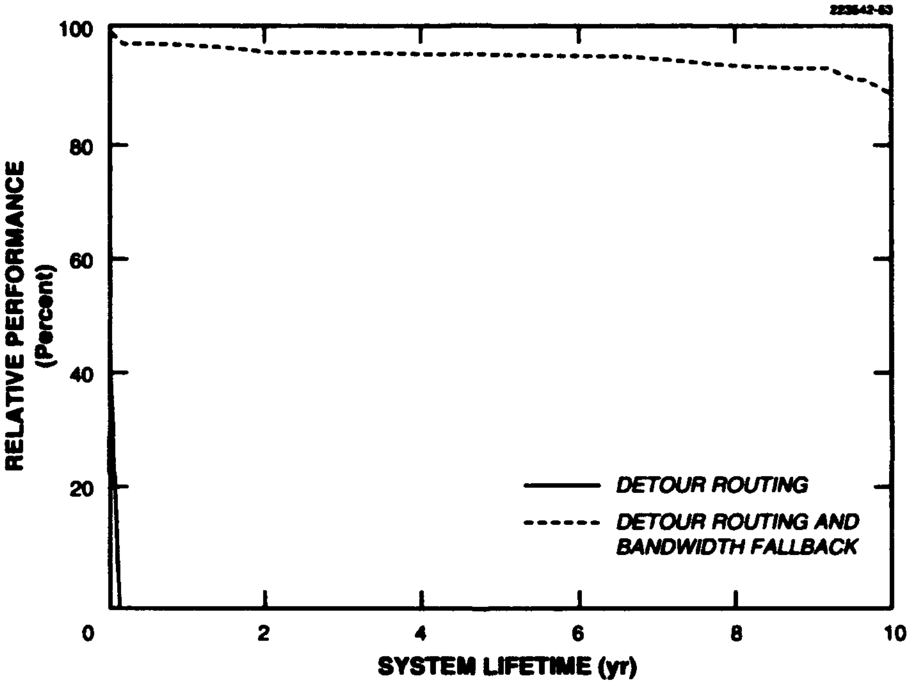

Figure 54. Bandwidth fallback performance with no spares.

10. CONCLUSIONS

10.1 Results of the Research

This report demonstrates a number of promising solutions to the problems of reliability and control of semiconductor lasers in large-scale multiprocessor networks.

For transient errors, an EDC is preferred over an error-correcting one, because the additional circuit complexity for the error correction is not warranted by any significant system-level benefit. An error-detecting Hsiao code, implemented at a bit overhead of around 12%, will suffice to relax the BER requirement to a very tractable level, such as 10^{-7} or 10^{-8}. This is a systematic code that can be coded and decoded in parallel with the actual data transmission.

Given such leeway in the permissible error rate, my experiments have shown that it is feasible to use the link BER as a feedback variable to control the laser drive current level. The experimental feedback system is stable, with a gain margin of at least 10 dB. Such a feedback system can control both for threshold current variation (that is, temperature or age) but also for optical medium degradation, as long as the feedback system is given control of laser pulse current as well as the bias level. The intelligent drive control system also offers the possibility of controlling laser wear-out failures by tracking laser wear-out trends and scheduling timely replacement.

For the problem of laser failures, this report has shown a switch design that enables the use of redundant spare optical links to replace failed ones. The substitution switch seems well suited to VLSI implementation. Provision of 10 to 15 spare links per channel seems to suffice to allow continued system operation until the effects of laser wear out begin to assert themselves.

The use of detour routing to exploit redundant paths in the network topology offers some fault-tolerance benefit, but a new approach, called bandwidth fallback, offers a dramatically better improvement.

Bandwidth fallback allows the use of partially-failed channels via the provision of simple switching hardware that resizes the transmitted data to match the remaining data channel width. If an appropriate substitution switch is provided, bandwidth fallback can provide better fault tolerance than redundant sparing, even with no spare links at all. Even as it enters the laser wear-out time frame, such a system continues operation with a performance loss of only a few percent.

10.2 Topics for Further Study

The research presented here opens up several topics for further study. A few of the possibilities are listed below. The list is roughly ordered according to a judgement of their breadth: the first topics are potentially suitable for Masters-level research, followed by larger topics potentially suitable for doctoral theses, and ending with broad research fields, encompassing many different research opportunities.

1. It would be worthwhile to try VLSI implementations of the substitution switches and the bandwidth fallback switches from Section 9. Can these be implemented reasonably in VLSI?

2. Is VLSI implementation of large numbers of DACs for intelligent laser drive control as straightforward as suggested in Section 7?

3. How practical is VLSI implementation of the Hsiao-code-based EDC system described in Section 6? Combined with a substitution switch? A bandwidth fallback switch?

4. Section 6 assumes that transient errors on different channels would be independent of each other. This assumption is critical to the conclusion about the efficacy of EDC. Is it valid on an actual parallel-path optical link? On a link under intelligent laser drive control?

5. As mentioned in Section 4.3.5, it is assumed that laser failures will be independent. For lasers implemented together in an array, this assumption is patently false: failures of such lasers are obviously correlated. How does this alter the failure control conclusions presented here?

6. The within-array correlation mentioned above offers an opportunity to simplify the intelligent laser drive control system: rather than control individual lasers, one could use the same control signal for all the lasers in an array. Does this yield adequate control? What impact does this have on the transient-error-control performance?

7. In Section 7.5.2, a number of possible benefits were suggested from a laser monitoring and replacement program based on laser wear-out tracking. Based on a reasonable cost model, what benefits could be expected from such a monitoring program? Would it be worthwhile?

8. The VLSI devices suggested in items 1 to 3 above, along with the laser drivers and optical receivers, form the elements of an interface between the processors and the optical components (laser and receiver arrays). How closely can these elements be integrated? What role does electromagnetic interference in the receivers play?

9. The substitution and bandwidth-fallback switches proposed here are not specific to optical networks, but could be applied to electrical (wired) networks as well. Based on a reasonable model of electrical network failures, do the proposed solutions make sense in such a context? What are the critical differences between electrical and optical networks in this regard?

10. Increasing the laser output power makes the drive circuit more complex and shortens laser life, but it simplifies receiver design. What impact do the solutions proposed here (EDC, laser drive control, bandwidth fallback) have on the best choice of laser output power level?

11. What impact do the proposed solutions have on optimal laser array parameters: threshold current, reliability, and array size?

12. Information has been discovered about the stability of the feedback system, it was only by the "brute-force" method of increasing the feedback gain. What is an appropriate mathematical model of the control system and its stability?

13. More generally, the laser drive system controls two variables (bias and pulse current) based on the arrival times of link errors. What are the limits of such a control system? What is the best control strategy? Is there a theoretically optimal control algorithm?

14. The multiprocessor is considered to have failed when there is insufficient connectivity to any node. Much work is now being done on fault-tolerant multiprocessing via reallocation of work from failed nodes to working ones. How do these higher-level methods interact with the methods suggested here? Is there a synergism from combining them?

15. Data in a processor can be conveyed either electrically or optically. As optical interconnection is made cheaper and more reliable, where will electrical signaling continue to be superior?

10.3 Feasibility of Optical Multiprocessor Networks

With the use of the techniques described here: EDC, intelligent laser drive control, redundant-spare substitution switching, and bandwidth fallback (especially if the circuits involved can be effectively implemented in VLSI), semiconductor laser reliability and control should not bar their use in large-scale multiprocessor networks.

APPENDIX A
LASER DRIVE CONTROL EXPERIMENTAL SETUP

The setup for the laser drive control experiments discussed in Section 7 is here described in detail.

A.1 Hardware

The overall expermental setup is shown in Figure 26. Not shown in the figure is an adjustable iris in the free-space optical path, which offers no obstacle to the light beam when open, and blocks 97.4% of the light beam when closed. Figure 27 is a photograph of the overall experimental setup. The optical components are mounted on a benchtop vibration-isolated optical table. From left to right they are: photodiode/receiver board, optical iris, and laser/driver board. Next to the optical table, one can see the computer interface box and the data link analysis board.

A.1.1 Laser/Driver Board

The laser/driver board is shown in Figures A-1 and A-2. On the front of the board, there is an aluminum block, holding a Corning 350110 aspheric lens, a Mitsubishi ML7761 laser diode (1300-nm wavelength), and an Analog Devices AD590KH temperature transducer. (Note that this particular lens was actually ill-suited to this application, having an AR coating designed for a wavelength of 775 nm, instead of 1300 nm. This helps explain some of the anomalous light output readings seen in Section 7.4.4.)

The aluminum block and the laser diode were fixed in position when the board was constructed. The lens was then inserted and its position adjusted for optimal collimation of the output beam. The lens was then fixed in position with a nylon setscrew.

In Figure A-1, there can also be seen a large power resistor mounted just in front of the laser/driver board, This was connected to line power through a Variac and was used to control the laser temperature in the temperature-based tests.

On the rear of the board, there is an NEL NL4512-2 laser driver circuit, a Motorola MC33074 operational amplifier, and several discrete components. A simplified schematic of the board is given in Figure A-3, omitting power bypass capacitors and similar details. All control inputs (bias current and pulse current) and monitoring outputs (temperature and light output) are current based, to avoid ground-loop problems and to ease the interconnection between the −4.5 V-powered driver board and the +5 V-powered computer interface.

Control of the NEL driver circuit was rather difficult to understand, and it required some experimentation to arrive at a working design. The data sheet (apparently translated from the Japanese) stated that the V_{CSDC} pin controlled the bias current and the V_{CSAC} pin controlled the pulse current. Connected this way, the circuit did not work correctly. Eventually, the circuit shown in Figure A-3 was arrived at. The pulse and bias level current commands from the computer

216779-2

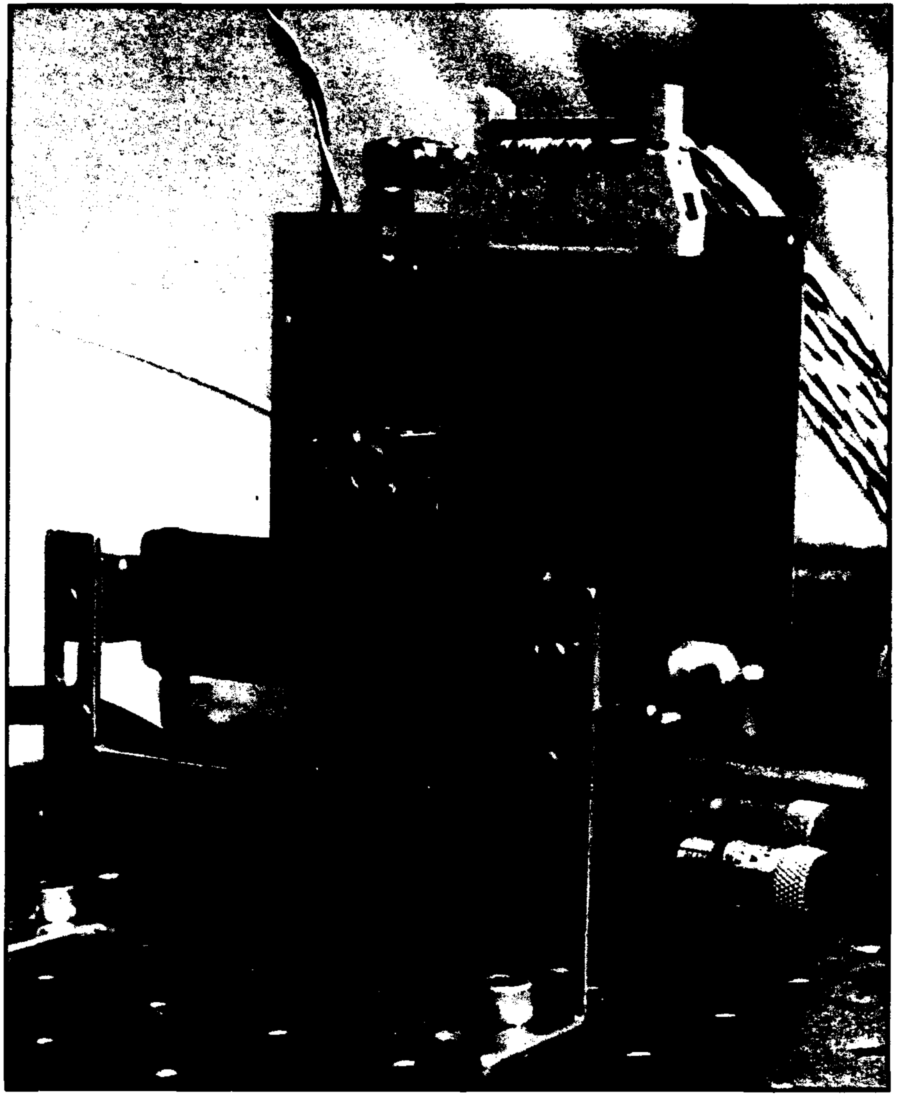

Figure A-1. Laser/driver board, front view.

216779-1

Figure A-2. Laser/driver board, rear view.

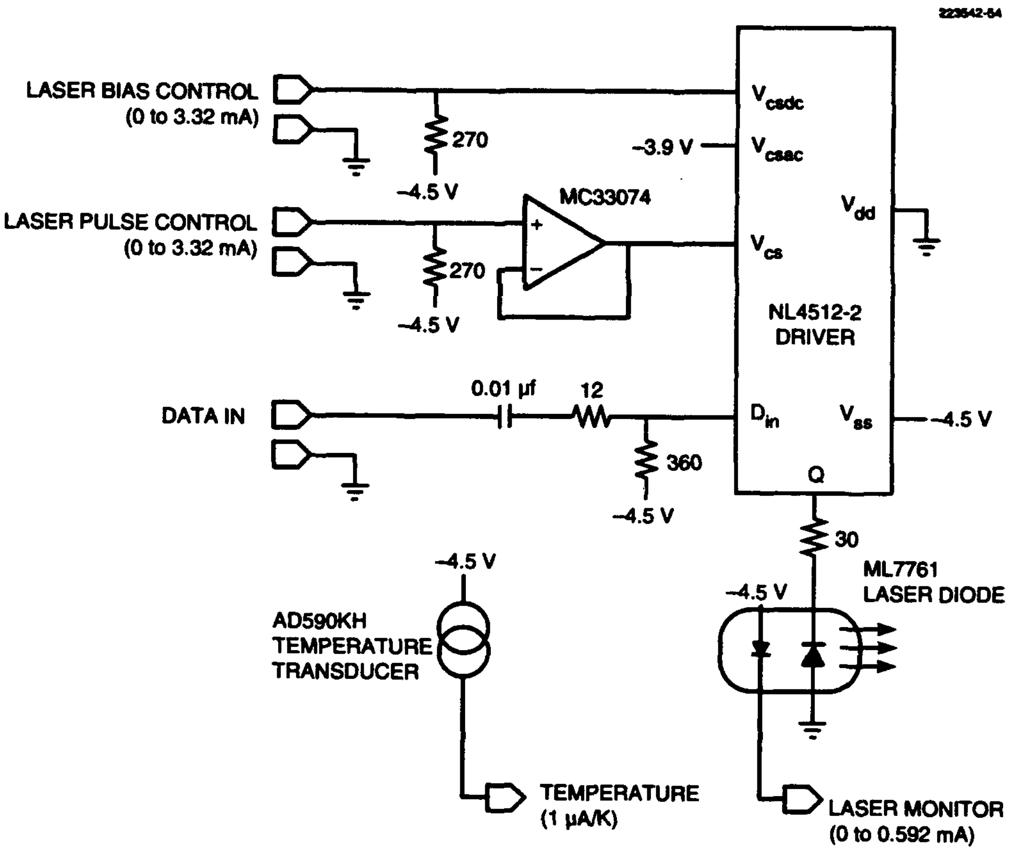

Figure A-3. Simplified laser/driver board schematic.

interface are applied to 270Ω resistors to V_{SS}, to develop the required control voltages (V_{SS} to $V_{SS} + 0.7$). The V_{CS} input presented an unexpected nonlinear load, so that the addition of a buffer amplifier was needed to control it adequately.

The laser driver input is ac-coupled to simplify level-shifting between the Gazelle data link analysis board output and the NEL laser driver circuit. Because the Gazelle board uses a balanced (zero-dc-bias) line code on its data output, ac coupling can be used without fear of baseline wander.

A.1.2 Photodiode/Receiver Board

Figure A-4 shows the photodiode/receiver board and the Newport optical iris used in the experiments. The photodiode/receiver board is much simpler than the laser/driver board, because it has no control inputs or monitoring outputs. Another Corning 350110 lens (also incorrectly AR coated) is mounted with a GE C30617 PIN photodiode. The lens/photodiode assembly is mounted and collimated in the same manner as the laser/driver board.

The photodiode current is amplified by an Avantek MSA-0370 amplifier before output to the Gazelle data link analysis board. As with the laser driver input, the photodiode output is ac-coupled.

A.1.3 Computer Interface

The computer interface box, shown in Figure A-5, provides a means for a UNIX workstation to control the laser drive level and to read the temperature and laser monitor outputs for experiment logging. The interface is based on a Motorola MC68HC11E2 microcontroller, which includes an eight-channel A/D converter on-chip.

The schematic of the analog section of the interface is shown in Figure A-6. Two Analog Devices AD558 eight-bit DACs develop output voltages between 0 and 2.56 V on command from the microcontroller. Each of these voltages is then converted to current by an op-amp and field effect transistor. As noted above, all signals between the computer interface and the laser/driver board are current-mode signals. The laser monitor and temperature current signals are converted to voltage and sent to the on-chip A/D converters.

The microcontroller communicates with the UNIX workstation via a 9600-baud RS-232 serial interface. It was originally connected directly tu a workstation, but the connection was later transferred to an Annex terminal server, which communicates with the workstation via an Ethernet connection.

A.1.4 Data Link Analysis Board

The data link analysis board is a Gazelle HOT ROD Development System (HRDS), shown in Figure A-7. In these experiments, it generates a test data pattern and transmits it at a 1 Gbit/s line rate, using a dc-balanced line code, to the laser/driver board. It receives the resulting data stream

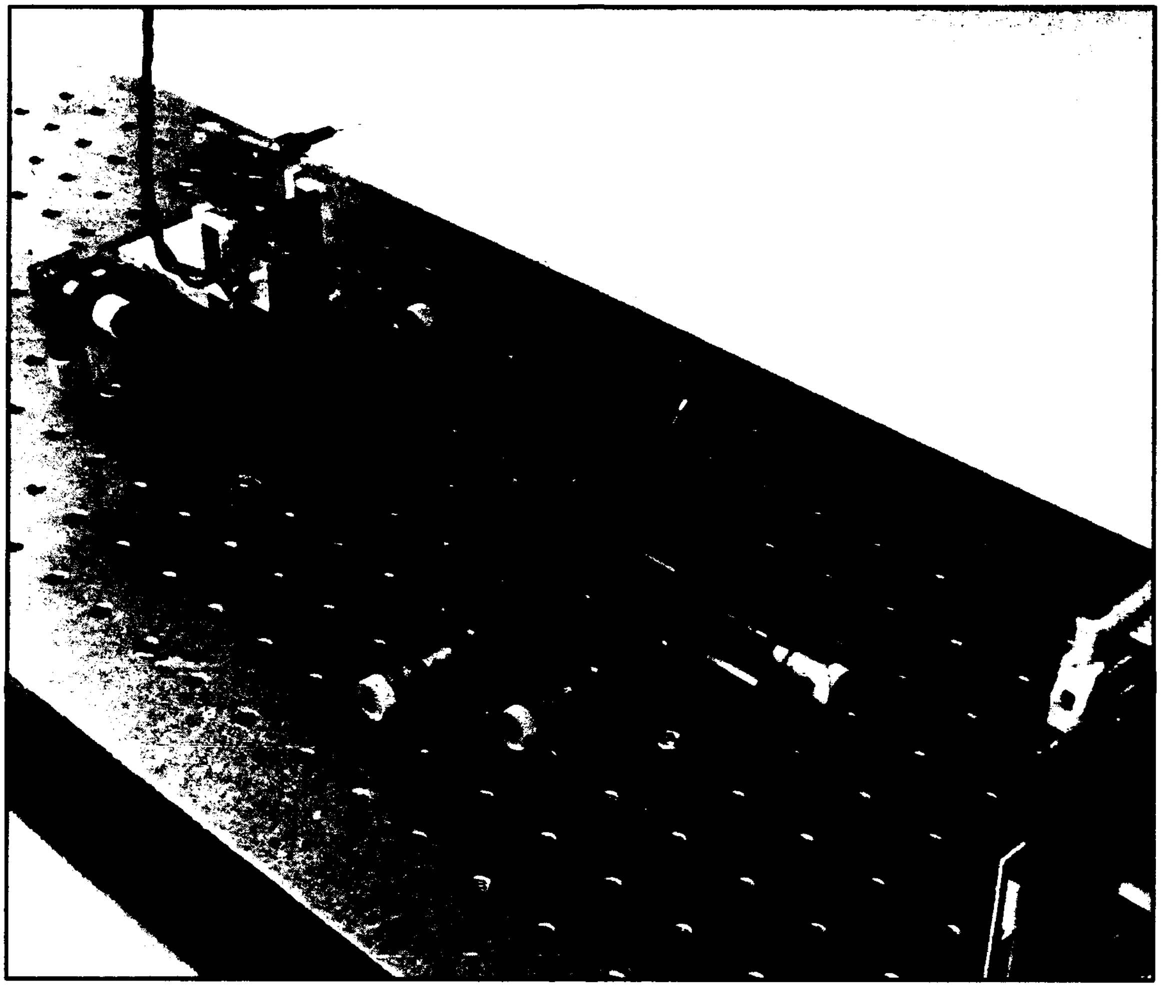

Figure A-4. Photodiode/receiver board.

216779-5

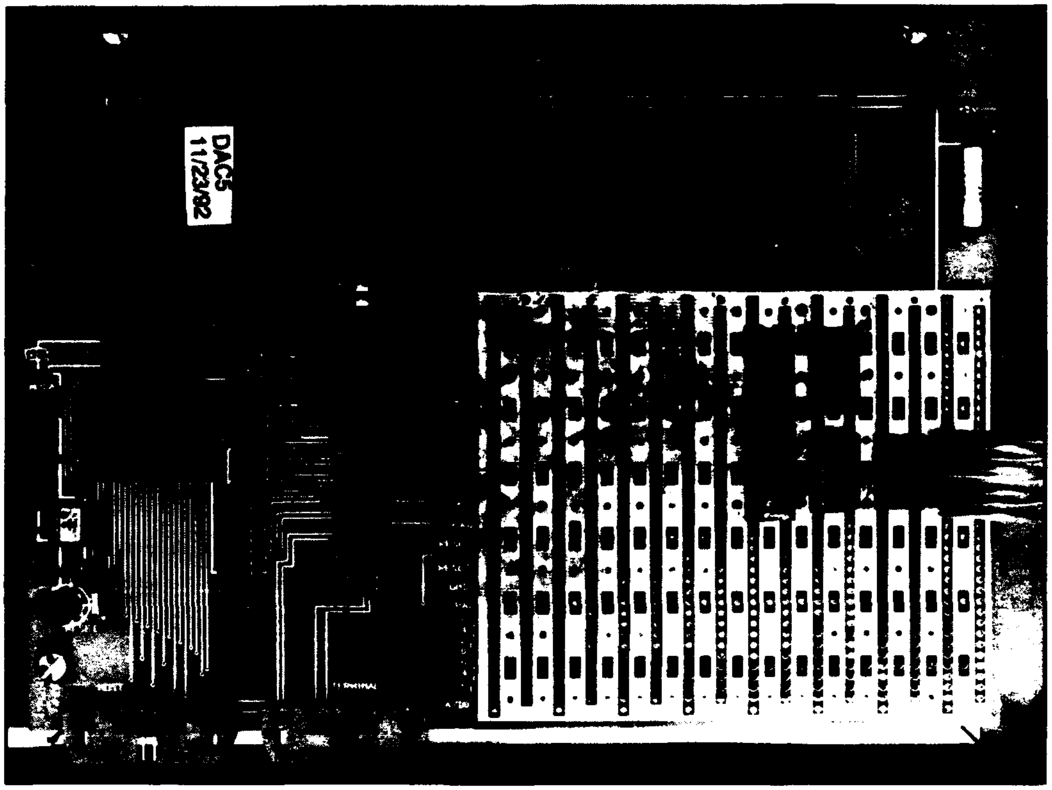

Figure A-5. Computer interface.

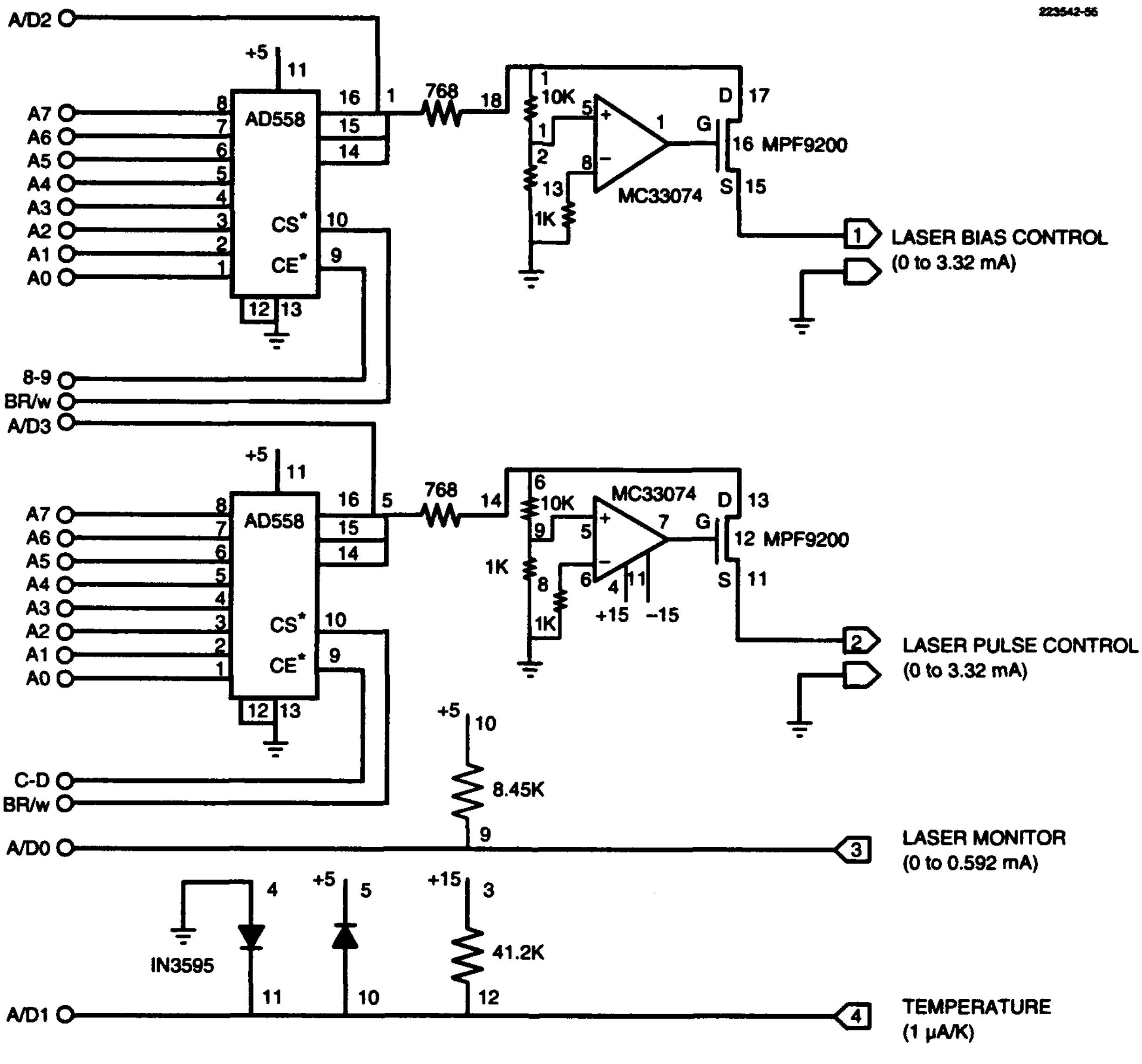

Figure A-6. Computer interface schematic (analog section).

220101-1

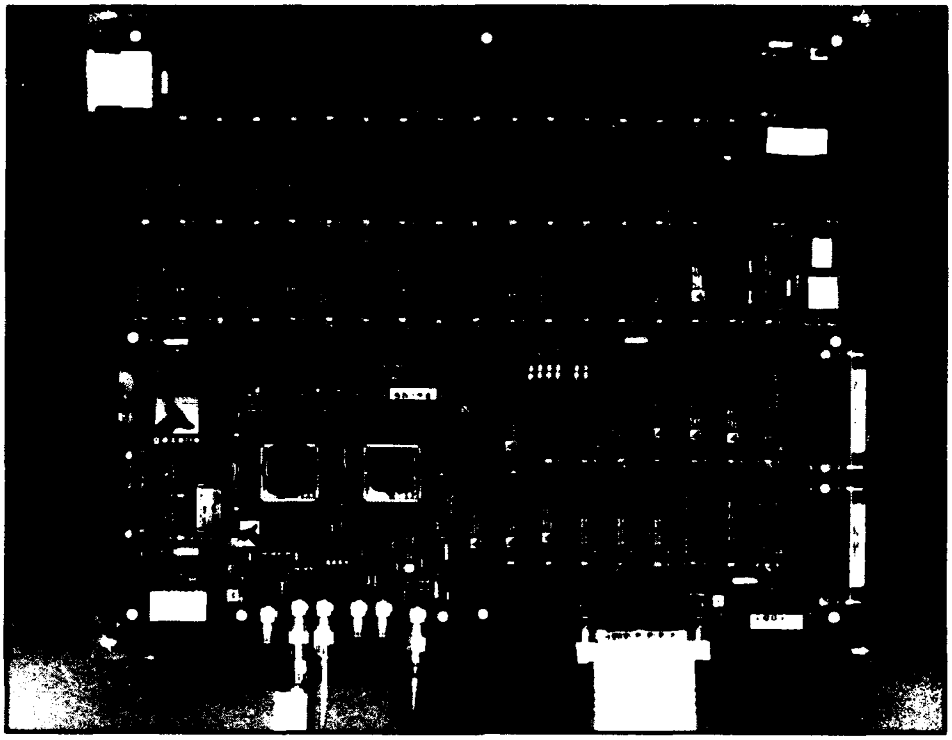

Figure A-7. Data link analysis board.

from the photodiode/receiver board and com ares the received and transmitted data, keeping a running count of errors.

The analysis board is actually intended to facilitate design of systems using the Gazelle "HOT ROD" communication boards, one of which can be seen in Figure A-7, mounted as a daughterboard on one corner of the main Gazelle HRDS board. The HRDS board actually generates and receives data in 40-bit-wide words at 1/40th of the line rate, via the HOT ROD. The HOT ROD performs parallel-to-serial and serial-to-parallel conversions with some of Gazelle's GaAs integrated circuits.

The Gazelle board has a 4800-baud RS-232 serial interface for connection to a terminal, through which the operator can start and stop the data gathering and get error reports. Instead of a terminal, this serial interface was connected to the UNIX workstation, through the Annex terminal server mentioned above. The software described in the next section had therefore to emulate a person sitting at a terminal in order to control the Gazelle board.

A.2 Software

The laser drive control experimental software is written in C, and runs on a UNIX workstation. A Sun Sparcstation was used, but any comparable UNIX computer would have sufficed.

The first problem to be solved was interfacing with the experimental equipment. The custom-made computer interface was not a problem, because its communication protocol could be changed at will by modifying the microcontroller firmware. The Gazelle board was another matter. Its protocol (assuming a human operator) is fixed into proprietary firmware, the source code of which is unavailable. The UNIX workstation is made to emulate a person.

Fortunately, tools for doing this are readily available. The "expect" library of Libes [47] was used (itself based on the "TCL" library of Ousterhout [48]), which was expressly written for this sort of operation. The feedback control program handles all the hardware interface chores via "expect" library routines.

The feedback program's strategy is implemented via a finite-state machine (FSM), described in Section 7.4.2 and diagramed in Figure 28. The state machine's basic timing was limited by the slow interface to the Gazelle board. Because the board's control firmware insists on sending two screenfuls of text, at 4800 baud, in response to every error-status query, the error status could only be obtained every 1.3 s.

In Section 7.4.9, the basic cycle of the FSM entails gathering three error measurements, which will take 3.9 s. If the number of errors during that time does not exceed a user-specified threshold (in the experiments shown in Section 7, the threshold was one error), then the error rate is judged to be acceptable. If, after any of the three error readings the error threshold is exceeded, the FSM cycle ends immediately, and the error rate is judged to be too high.

Every 3 s, the feedback program writes an entry into an experiment log file. The entry gives the following parameters:

- Laser bias current
- Laser pulse current
- BER
- Temperature reading
- Laser output monitor reading

The experimental data plots in Section 7 are derived from these log files.

APPENDIX B
NETWORK SIMULATION SOFTWARE

The bandwidth-fallback simulation results presented in Section 9 were produced by a suite of simulation programs written in C, running on a UNIX workstation.

The simulation task was separated into stochastic and deterministic components and different programs were written for each.

The stochastic tasks were performed by two relatively simple programs: a laser failure simulator and a packet generator. The laser failure simulator took as parameters the laser failure probability, channel width, and network mesh size, and generated a file indicating how many lasers had failed in each channel by a simple Bernoulli process.

The packet generator took as parameters the network mesh size, number of packets for each node to send, and the maximum packet size. It then created descriptions of the desired number of packets for each node, specifying the destination node and the packet length (in flits). The length was chosen as a uniformly distributed random variable between 1 and the maximum packet length. The choice of destination node was a little more constrained. Rather than assign destinations completely at random, the packet generator ensured that each node received and sent exactly the same number of packets. It therefore constructed a pool of packets, with each node being the destination of an equal number of packets. The pool was then shuffled and dealt out to the nodes; this process determined which node would be the sender of the packet. All the packet information was then written out onto a file.

The rest of the simulator was deterministic. It received as parameters an ensemble of laser-failure and packet-assignment files, performed simulations on each combination of them, and reported the mean and deviation of the performance results from the various simulation runs. For each run, the simulation result was the estimated number of cycle required for all the nodes to transfer all their data packets.

The structure of the simulation program might be more easily understood if some of the major data structures were examined.

```
struct node_struct {
    int sleep_time,xfers_left;
    unsigned char x,y;
    struct node_struct *next;     /* linked list for events */
    struct channel_struct ch[4];
    /* the following data describe the packet
       originating at this node */
    struct xfer_struct *my_xfer;
    enum {PK_READY,PK_OPENING,PK_XFERING,PK_CLOSING} state;
    int min_fract;
    struct channel_struct *next_chan,**detour_chan;
    struct vc_struct *head_vc;
};
```

The node structure is the most important one in the program. The structures describing the four channels leaving this node reside here. Because each node can have at most one active packet at a time; the packet information is kept here too. The packets go through three phases: pushing through the network (establishing a path), transferring data, and tearing down the path. The packet also keeps track of the smallest fallback fraction on its path, as this affects the packet's throughput. This is an event-driven simulator, so the nodes are kept on a linked list that functions as an event queue.

```
/* virtual channel structure */
struct vc_struct {
  struct node_struct *user;
  struct vc_struct *prev;
  struct channel_struct *c;
};

struct channel_struct {
  int fraction,n_vc,n_users;
  struct vc_struct vcs[MAX_VC];
  struct node_struct *src,*dest;
  struct channel_struct *prev,**detour;
};
```

The channel structure defines the basic topology of the network via its src and dest pointers, although this might be overkill for a simple mesh network. Each channel is associated with four virtual channels, so a packet is refused entry to an actual channel only when all four of its virtual channels are used up. However, special provisions for the Virtual Channel detours around failed channels. If a channel forms a part of such a detour path, then that detour path has one of the virtual channels dedicated to it exclusively. This is necessary to avoid deadlock: if normal transactions could freeze out a detour path, then the detour would no longer be logically equivalent to the channel it replaces.

At the start of the simulation, one node structure is created for each node in the network to be simulated. The packet data is read from the packet-generator output file, and the channel capacity is read from the laser failure file. Each channel is checked for adequate remaining capacity; its capacity is inadequate (and the network can't fix it with the given network parameters), then the simulation run is declared "failed" and aborted. In the results given in Section 9, such runs were counted as zero performance and averaged with the performance of the nonfailed runs, if any. The performance of normal runs was calculated as the normal (no-failure) execution time divided by the actual execution time with failures.

The simulation results were based on runs of a 10×10 mesh network, with 64-bit-wide paths, 16-flit maximum packet length, and 32 packet transfers per node. The simulations were performed for 11 values of the laser failure probability $P(t)$, from 0 to 0.2 in steps of 0.02. These results were then transformed into system lifetimes using the model given in Section 4.3.4.

REFERENCES

1. S. Ryu et al., "Field demonstration of 195-km-long coherent unrepeatered submarine cable system using optical booster amplifier," *Electron. Lett.* 28(21), 1965-1967 (1992).

2. M. Fukuda, ***Reliability and Degradation of Semiconductor Lasers and LEDs***, Norwood, Mass.: Artech House (1991).

3. A. Guha et al., "Optical interconnections for massively parallel architectures," *Appl. Opt.* 29(8), 1077-1093 (1990).

4. D.H. Hartman, "Digital high speed interconnects: A study of the optical alternative," *Opt. Eng.* 25(10), 1086-1102 (1986).

5. P.R. Haugen, S. Rychovsky, and A. Husain, "Optical interconnects for high speed computing," *Opt. Eng.* 25(10), 1076-1984 (1986).

6. M.R. Feldman et al., "Comparison between optical and electrical interconnects based on power and speed considerations," *Appl. Opt.* 29(9), 1742-1751 (1990).

7. T.F. Knight, Jr., and A. Krymm, "A self-terminating low-voltage swing CMOS output driver," *IEEE J. Solid State Circuits* 23(2), 457-464 (1988).

8. I.M. Ross, "Electronics/photonics technology: Vision and reality," in *Proc. of the SPIE*, Volume 1389 (1990) pp. 2-26.

9. M. Nakamura and M. Obara, "Advanced semiconductor devices for optical communication systems," *Toshiba Rev.* 44(7), (1989).

10. T.F. Knight, Jr., "Technologies for low latency interconnection switches," in *Proc. 1989 ACM Symposium on Parallel Algorithms and Architectures* (1989), pp. 351-358.

11. R.W. Keyes, "Physical limits in digital electronics," in *Proc. of the IEEE* 63(5) (1975), pp. 740-767.

12. D.A.B. Miller, "Optics for low-energy communication inside digital processors: Quantum detectors, sources, and modulators as efficient impedance converters," *Opt. Lett.* 14(2), (1989).

13. D.A.B. Miller, "Quantum-well devices for optics in digital systems," in *Proc. of the SPIE*, Volume 1389 (1990), pp. 496-502.

14. J.L. Bufton, "Experimental study of soliton transmission over many thousands of kilometers in fiber with loss periodically compensated by Raman gain," in *Summaries of Papers Presented at the Conference on Lasers and Electrooptics*," Optical Society of America (1989), pp. 150-151.

15. ANSI standard X3.166-1990, fibre data distributed interface, (FDDI)–token ring layer medium dependent (PMD), American National Standards Institute (1990).

REFERENCES
(Continued)

16. F.E. Ross, "Overview of FDDI: The fiber distributed data interface," *IEEE J. Selec. Areas Commun.* 7(7), 1043-1051 (1989).

17. A.G. Dickinson et al., "Free-space optical interconnects," in *Proc. of the SPIE*, Volume 1389 (1990), pp. 503-514.

18. D.Z. Tsang, "Free optical interconnects," in *Proc. of the SPIE*, Vol. 994 (1988), pp. 73-76.

19. L.A. Bregman et al., "Holographic Optical interconnects for VLSI," *Opt. Eng.* 24(10), 1109-1118 (1986).

20. C.T. Sullivan and A. Husain, "Guided-wave optical interconnects for VLSI systems," in *Proc. of the SPIE*, Volume 881 (1988), pp. 172-176.

21. F. Lin et al., "Optical multiplanar VLSI interconnects based on multiplexed waveguide holograms," *Appl. Opt.* 29(8), 1126-1133 (1990).

22. E.E.E. Frietman et al., "Parallel optical interconnects: Implementation of optoelectronics in multiprocessor architectures," *Appl. Opt.* 29(8), 1161-1177 (1990).

23. H. Kobayashi, H. Kanbara, and K. Kubodera, "Optical gating performance using a semiconductor-doped glass etalon," *IEEE Photonics Tech. Lett.* 2(4), 268-270 (1990).

24. D.A.B. Miller et al., "Field-effect transistor self-electrooptic device: Integrated photodiode, quantum well modulator and transistor," *IEEE Photonics Tech. Lett.* 1(3), 62-64 (1989).

25. W.J. Dally, *A VLSI Architecture for Concurrent Data Structure*, Norwell, Mass.: Kluwer Academic Publishers (1987).

26. A. Agarwal, "Limits of interconnection network performance," *IEEE Trans. on Parallel and Distributed Sys.* 2(4), 398-412 (1991).

27. A.W. Wilson, Jr., "Hierarchical cache/bus structure for shared memory multiprocessors," in *Conf. Proc.-Annu. Sympos. on Comput. Architec.* (1987), pp. 244-252.

28. M.R. Feldman and C.C. Guest, "Nested crossbar connection networks for optically interconnected processor arrays for vector-matrix multiplication,"*Appl. Opt.* 29(8), 1068-1076 (1990).

29. W. Crowther et al., "Performance measurements on a 128-node butterfly parallel processor," in *Proc. of the 1985 Int. Conf. on Parallel Processing* (1985), pp. 531-540.

30. T. Leighton, "The role of randomness in the design of parallel architectures," in *Advanced Research in VLSI: Proc. of the Sixth MIT Conf.*, ed. W.J. Dally, Cambridge, Mass: MIT Press (1990), pp. 177-178.

31. A. Yariv, *Optical Electronics*, NY, NY: Holt, Rinehart, and Winston (1985).

REFERENCES
(Continued)

32. O. Fujita, Y. Nakano, and G. Iwane, " Screening by aging test for highly reliable laser diodes," *Electron. Lett.* 21(24), 1172-1173 (1985).

33. S.P. Sim et al., "High reliability InGaAsP/InP buried heterostructure lasers grown entirely by atmospheric MOVPE," in *Fourteenth European Conf. on Opt. Commun. (ECOC)*, IEEE Conf. Pub. 292, Part 1 (1988).

34. W.C. Athas and C.L. Seitz, "Multicomputers: Message-passing concurrent computers," *Computer* 21(8), 9-25 (1988).

35. D. Chaiken et al., "Directory-based cache coherence in large-scale multiprocessors," *Computer* 23(6) 49-59 (1990).

36. W.J. Dally and C.L. Seitz, "Deadlock-free massage routing in multiprocessor interconnection networks," *IEEE Trans. on Comput.*, 36(5), 547-553 (1987).

37. H. Sullivan and T.R. Bashkow, "A large scale, homogeneous, fully distributed parallel machine," in *Conf. Proc. - Annu. Sympos. on Comput. Architec.*, IEEE/ACM, 105-117 (1977).

38. C.J. Glass and L.M. Ni, "The turn model for adaptive routing," in *Conf. Proc.-Annu. Sympos. on Comput. Architec.*, IEEE/ACM, 278-287 (1992).

39. S. Lin and D.J. Costello, *Error control coding: fundamentals and applications*, Englewood Cliffs, NJ: Prentice-Hall (1989).

40. T.R.N. Rao and E. Fujiwara, *Error-Control Coding for Computer Systems*, Englewood Cliffs, NJ: Prentice-Hall (1989).

41. M.Y. Hsiao, "A class of optimal minimum odd-weight-column SEC-DED codes," *IBM J. of Res. and Dev.* 14, 395-401 (1970).

42. P.W. Shumate et al., "GaAlAs laser transmitter for lightwave transmission systems," *Bell Sys. Tech. J.* 57(6), 1823-1836 (1979).

43. D.P. Siewiorek and R.S. Swarz, *The Theory and Practice of Reliable System Design*, Chapter 3, Bedford, Mass.: Digital Press (1982).

44. W.H. Press et al., *Numerical Recipes in C*, Cambridge, Mass.: Cambridge University Press (1988), p. 176.

45. W.J. Dally, "Virtual-channel flow control," in *Conf. Proc.-17th Annual Int. Sympos. on Comput. Architecture*, IEEE Comput. Soc. (1990), pp. 60-68.

46. H.T. Kung and O. Menzilcioglu, "Virtual channels for fault-tolerant programmable two-dimensional processor arrays," Carnegie-Mellon University, Computer Science Dept., Technical Report CMU-CS-87-171 (1986).

REFERENCES
(Continued)

47. D. Libes, "Expect: Curing those uncontrollable fits of interaction," in *Proc. of the Summer 1990 USENIX Conf.*, USENIX Association, Berkeley, CA (1990).

48. J.K. Ousterhout, "TCL: An embeddable command language," in *Proc. of the Winter 1990 USENIX Conf.*, USENIX Association, Berkeley, CA (1990).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)

2. REPORT DATE
22 December 1993

3. REPORT TYPE AND DATES COVERED
Technical Report

4. TITLE AND SUBTITLE
Control and Reliability of Optical Networks in Multiprocessors

5. FUNDING NUMBERS
C — F19628-90-C-0002

6. AUTHOR(S)
James J. Olsen

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Lincoln Laboratory, MIT
P.O. Box 73
Lexington, MA 02173-9108

8. PERFORMING ORGANIZATION REPORT NUMBER
TR-985

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

HQ Electronic Systems Center ESC/ENKL
Hanscom, AFB, MA 01730-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
ESC-TR-93-237

11. SUPPLEMENTARY NOTES
None

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

Optical communication links have great potential to improve the performance of interconnection networks within large parallel multiprocessors, but semiconductor laser drive control and reliability problems inhibit their wide use. This report describes a number of system-level solutions to these problems. The solutions are simple and inexpensive enough to be practical for implementation in the thousands of optical links that might be used in a multiprocessor.

Semiconductor laser reliability problems are divided into two classes: transient errors and hard failures. It is found that for transient errors, the computer system might require a very low bit-error-rate (BER), such as 10^{-23}, without error control. Optical links cannot achieve such rates directly, but a much higher link-level BER (such as 10^{-7}) would be acceptable with simple error detection coding. A feedback system is proposed that will enable lasers to achieve these error levels even when laser threshold current varies. Instead of conventional techniques using laser output monitors, a software-based feedback system can use BER levels for laser drive control. Experiments demonstrate that this method is feasible and has other benefits such as laser wearout tracking and optical loss compensation.

For hard failures, one can provide redundant spare optical links to replace failed ones. Unfortunately, this involves the inclusion of many extra, otherwise unneeded optical links. A new approach, called "bandwidth fallback," is presented that allows continued use of partially failed channels while still accepting full-width data inputs, providing high reliability without any spare links.

It is concluded that the drive control and reliability problems of semiconductor lasers should not bar their use in large scale multiprocessors, because inexpensive system-level solutions to them are possible.

14. SUBJECT TERMS

redundancy	optical communication	laser drive control
feedback	interconnection	semiconductor laser
threshold current	switch network	bit error rate

15. NUMBER OF PAGES
142

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Same as Report