

UNIVERSITY OF CALIFORNIA

SANTA CRUZ

**COLLISION AVOIDANCE TECHNIQUES FOR PACKET-RADIO
NETWORKS**

A dissertation submitted in partial satisfaction
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER ENGINEERING

by

Chane Lee Fullmer

June 1998

The dissertation of Chane Lee Fullmer is approved:

Professor J. J. Garcia-Luna-Aceves, Chair

Professor Patrick Mantey

Professor Charlie McDowell

Dean of Graduate Studies

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE JUN 1998		2. REPORT TYPE		3. DATES COVERED 00-06-1998 to 00-06-1998	
4. TITLE AND SUBTITLE Collision Avoidance Techniques for Packet-Radio Networks				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of California at Santa Cruz, Department of Computer Engineering, Santa Cruz, CA, 95064				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 172	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

Copyright © by
Chane Lee Fullmer
June 1998

Contents

Abstract	viii
1 Introduction	1
2 FAMA for Fully Connected Networks	7
2.1 FAMA Protocols	8
2.1.1 MACA	10
2.1.2 FAMA-NTR (Non-persistent Transmit Request)	11
2.2 Floor Assignment in FAMA along a Single Hop	14
2.3 Approximate Throughput Analysis	19
2.3.1 Assumptions and Notations	20
2.3.2 FAMA-NTR	22
2.3.3 Slotted FAMA-NTR	25
2.3.4 MACA	27
2.3.5 Slotted MACA	32
2.3.6 Performance Comparison	35
2.4 Related Work	41
2.5 Conclusions	42
3 FAMA Emulation of Collision Detection	44
3.1 FAMA-PJ	46
3.2 Floor Assignment in FAMA-PJ	50
3.3 Approximate Throughput Analysis	53
3.3.1 Assumptions and Notations	54
3.3.2 FAMA-PJ	56
3.3.3 Slotted FAMA-PJ	58
3.3.4 Performance Comparison	60
3.4 Average Delay	63
3.4.1 FAMA-PJ	66
3.4.2 Slotted FAMA-PJ	67
3.4.3 Throughput-Delay Characteristics	69
3.5 Conclusions	70

4	FAMA for Ad-Hoc Networks	71
4.1	FAMA Protocols	75
4.1.1	Overview	75
4.1.2	FAMA-NCS	78
4.1.3	FAMA-NBR	82
4.1.4	FAMA-NPS	83
4.2	Correct Floor Acquisition	85
4.2.1	Using Carrier Sensing	85
4.2.1.1	FAMA-NBR	89
4.2.2	Using Packet Sensing	90
4.3	Comparative Throughput Analysis	92
4.3.1	Assumptions and Notations	92
4.3.2	Throughput in Fully Connected Networks	94
4.3.3	Throughput in Wireless LANs	98
4.3.4	Throughput in Ad-Hoc Networks	104
4.4	Simulation Results	112
4.5	Related Work	117
4.6	Concluding Remarks	118
5	Collision Avoidance for Multichannel Networks	120
5.1	FAMA-MC Protocol	123
5.2	Correct Floor Acquisition in fully connected multichannel networks	125
5.3	Approximate Throughput Analysis	127
5.4	Simulation Results	130
5.5	Conclusion	133
6	Implementation: WINGs for Internets	135
6.1	WING Protocol Architecture	137
6.2	FAMA-NCS	140
6.3	WIRP	142
6.3.1	Reliable Transmission of Updates	143
6.3.2	Neighbor Discovery Mechanism	143
6.3.3	Wireless Internet Routing	144
6.4	Simulation Experiments	146
6.5	Implementation Experience	150
6.5.1	WING Software Approach	150
6.5.2	WING Hardware Configuration	152
6.5.3	Wireless Internetworking Demonstrations	152
6.6	Conclusions	154
7	Conclusion	156
	Bibliography	158

List of Figures

2.1	MACA Specification	10
2.2	Unsafe MACA transmission period	11
2.3	FAMA-NTR Specification	12
2.4	MACA RTS/CTS collision	18
2.5	FAMA-NTR transmission periods	23
2.6	Slotted FAMA-NTR transmission periods	26
2.7	MACA transmission periods	28
2.8	A failed RTS transmission period in MACA	30
2.9	Slotted MACA transmission periods	32
2.10	Slotted MACA Transmission Periods	34
2.11	Throughput of FAMA protocols in a low-speed network.	37
2.12	Throughput of FAMA protocols in a high-speed network.	37
2.13	Throughput of FAMA-NTR protocols for different values of a using a 10 packet train in a high-speed network.	40
2.14	Throughput of FAMA-NTR versus traffic for different values of b in a high speed network.	40
3.1	FAMA-PJ Specification	48
3.2	FAMA-PJ active jamming failure under conditions where $\varepsilon > \tau$	50
3.3	Passive jamming periods	54
3.4	FAMA-PJ transmission periods	57
3.5	Slotted FAMA-PJ transmission periods	58
3.6	Analytical performance comparison of the FAMA-PJ protocol in a high speed network.	62
3.7	Throughput of FAMA-PJ protocols using a 10 packet train in a high-speed network.	62
3.8	Throughput of FAMA-PJ versus b in a high speed network.	63
3.9	Markov chain defining FAMA-PJ delay characteristics.	64
3.10	Delay performance of FAMA-PJ and slotted FAMA-PJ.	69
4.1	CTS Dominance in FAMA-NCS	77
4.2	FAMA-NCS Specification	81
4.3	FAMA-NPS Specification	84

4.4	Stations involved in interference of the exchange between S and R	87
4.5	HT NPS FAIL	91
4.6	FAMA-NCS transmission periods	95
4.7	Throughput of FAMA-NCS in fully connected networks	98
4.8	Throughput of FAMA-NCS for different values of b	98
4.9	Throughput of FAMA-NCS with independent groups	104
4.10	Throughput of FAMA-NCS in complimentary couple	104
4.11	Multihop Throughput vs Q	110
4.12	Multihop Throughput	111
4.13	Configurations	113
5.1	FAMA-MC Specification	122
5.2	FAMA-MC transmission periods	123
5.3	FAMA Multichannel Markov Process for data channel	128
5.4	Throughput of FAMA-MC using carrier sensing	130
5.5	Throughput of FAMA-MC using packet sensing	131
5.6	Maximum throughput per station of FAMA-MC in simulations	132
5.7	Multihop topology for FAMA-MC simulations	133
5.8	Comparison of Throughput for FAMA-MC in multihop topology	134
6.1	WINGS Software Architecture	138
6.2	WIRP routes at startup	146
6.3	WIRP routes after link failure	146
6.4	Sequence of packets in WIRP simulation	147
6.5	Throughput in Los Nettos with ideal radios	149
6.6	Throughput in Los Nettos with Utilicom radios	149
6.7	Average delay in Los Nettos tests	150

List of Tables

2.1	Throughput Equations for CSMA and FAMA protocols	35
2.2	FAMA protocol variables	36
3.1	Throughput Equations for CSMA and FAMA-PJ protocols	61
3.2	FAMA-PJ protocol variables	61
4.1	Throughput results for various configurations	114
4.2	Throughput comprison of FAMA-NCS, IEEE 802.11 and MACAW	116
5.1	Results for FAMA-NCS in an ad-hoc network	133
6.1	WING I Hardware Specifications	153

Abstract

Collision Avoidance Techniques for Packet-Radio Networks

by

Chane Lee Fullmer

Doctor of Philosophy in Computer Engineering

University of California at Santa Cruz

Professor J. J. Garcia-Luna-Aceves, Chair

Medium access control for devices that share a particular medium is a fundamental problem in communications networks. We present a new protocol for medium access control called *floor acquisition multiple access* (FAMA). Floor acquisition protocols guarantee data packets are received without collisions from other packets. We present FAMA protocols for both single and multiple channel devices operating in ad-hoc packet radio networks. We present analytical and simulation results for FAMA protocols.

This is dedicated
to the many friends, family and faculty
who so encouraged me along the way.

Acknowledgements

The text of this dissertation includes material that has been previously published [FG95a, FG95b, FG97a, FG97b] and [GBF⁺97] or accepted for future publication [GF98]. Professor J. J. Garcia-Luna-Aceves, listed as co-author in these publications, directed and supervised the research which forms the basis for this dissertation.

This work was supported in part by the Office of Naval Research (ONR) under Grants N00014-92-J-1807 and N00014-94-1-0688, and by the Defense Advanced Research Projects Agency under Grants DAAB07-95-C-D157 and DAAH04-96-1-0210.

Chapter 1

Introduction

With the increasing affordability of computers, society is fast approaching an era of “ubiquitous computing.”¹ Laptop computers, personal digital assistants and pagers are but a few examples of computing devices commercially available today. One aspect of ubiquitous computing is that people are now starting to carry their computers with them wherever they go, and use the computers to access information remotely as they move.

In the past, computing devices were generally large, stationary and connected to a wired network in most cases. However, computers now are often mobile, or temporal in their location, which requires communication networks that offer more flexibility than is available from a pre-configured wired infrastructure. Multihop packet-radio networks (or ad-hoc networks) are an ideal technology to extend the wired infrastructure to mobile users, to establish an “instant communication infrastructure” for military and civilian applications (e.g., ad-hoc networks in disaster areas resulting from flood, earthquake, hurricane, or fire);

¹Weiser [Wei93] describes ubiquitous computing as making many computers available throughout the physical environment, while making them effectively invisible.

a rapid infrastructures for remote or developing regions—or even ad-hoc regional networks for schools or communities [BVG97]. The key differences between ad-hoc and traditional wired infrastructures are as follows:

- Ad-hoc networks have smaller available bandwidths than wired networks.
- The topologies of ad-hoc networks are much more dynamic than wired networks due to host and router mobility and the characteristics of the network medium.
- To communicate among themselves, routers in ad-hoc networks must use a common transmission medium instead of multiple point-to-point interfaces.

The topology dynamics of ad-hoc networks coupled with the use of a shared transmission medium brings up the problem that, in some cases, a node may receive the concurrent transmissions of multiple neighbors that cannot hear one another. We call these nodes *hidden* from each other. If two or more of these hidden nodes transmit packets that would overlap in the same time period to a common receiver, their transmissions will typically fail at the intended receiver. This scenario has been called the “hidden-terminal problem”, as originally discussed by Tobagi and Kleinrock [TK75].

The sharing of a common transmission medium or channel by multiple nodes is determined by a medium access control (MAC) protocol. In an ad-hoc network, this medium or channel is a scarce resource, and as such, designing MAC protocols that are efficient even when hidden terminals are present is a fundamental problem in ad-hoc networks.

The first MAC protocol for packet-radio was ALOHA [Abr70]. In this protocol, radios did not sense or “listen” to the channel prior to transmitting. As such, the vulnerability period (the period of time during which a packet may be interfered with at its

intended destination) is twice the size of the packet transmission time. As a result the maximum channel throughput is very low (18% of the channel capacity).

To mitigate the high-loss rate of packets in ALOHA, Tobagi and Kleinrock introduced the carrier sense multiple access (CSMA) protocols [KT75]. In CSMA, the nodes sense the channel (i.e., listen) before transmitting a packet, and defer transmission if the channel is found to be busy. This limits the vulnerability period of a transmission to the time it takes for the radio signal to propagate across the channel. As the transmission time of a packet is usually much larger than the propagation delay of the channel, a packet's vulnerability period in CSMA is much shorter than in ALOHA. Accordingly, CSMA performs markedly better than ALOHA in a fully connected network (its maximum throughput is 80%, instead of 18% of the channel capacity). However, in an ad-hoc network with hidden terminals, sensing the channel activity at the transmitter does not offer any information as to the state of the channel at the intended receiver. This implies that the vulnerability period of a packet at the receiver becomes twice the transmission time of the packet, and CSMA's performance in ad-hoc networks degrades to that of ALOHA.

To solve the hidden-terminal degradation of CSMA, Tobagi and Kleinrock introduced the busy tone multiple access (BTMA) protocol [TK75]. In BTMA, a receiver transmits a separate tone on a special channel whenever it detects carrier on its data channel. Any sender first listens to the busy tone channel and defers transmission if this channel is busy. BTMA achieves a maximum throughput in a network with hidden terminals that is comparable to CSMA's performance in a fully-connected network. However, BTMA requires two sets of radio devices at each node, and is too costly to implement in most commercial applications.

The use of a request-to-send/clear-to-send (RTS/CTS) dialogue was first introduced for packet-radio networks in the split-channel reservation multiple access (SRMA) protocol [TK76]. The RTS/CTS dialogue was used to add a control mechanism to the channel access to allow the source and destination to schedule data transmissions more efficiently. SRMA uses three channels, one each for transmission of request (RTS) packets, answer (CTS) packets, and data packets, respectively. Because control packets are sent in separate channels they do not interfere in the data channel, and in a network with a central station and a population of user terminals, the data packets are sent free of interference from other packets, even when the terminals are hidden from each other. However, in a multihop network, there is no guarantee that a node will always hear a neighbors' CTS and data packets can interfere with each other, degrading performance. Additionally, requiring three sets of radio transceivers at each node is impractical commercially.

Multiplexing of RTS/CTS control packets and data packets over the same channel has been shown in carrier sense multiple access with collision avoidance (CSMA/CA) [Col83], which was originally designed for wired local area networks.

The multiple access collision avoidance (MACA) protocol [Kar90] and its modified version MACAW (MACA for Wireless) [BDSZ94] were proposed to operate with hidden terminal using a simple three-way dialogue without using carrier sensing at the transmitter. These protocols use what we call *packet sensing* (with packet sensing a station only reacts to complete, interference free packets, and does not detect any other type of activity on the channel). However, the performance of MACA and MACAW degrade to ALOHA's under high traffic conditions in networks with hidden terminals. An extension to MACAW called BAPU (Basic Access Protocol solUtions for wireless) [Vad95] uses separate channels

for control and data packets. Because all nodes use a common data channel and control packets may go unheard in BAPU, collisions of data packets are still possible which limits its maximum throughput.

The goal of our research is to introduce, analyze and compare new channel access methods that provide better throughput performance in ad-hoc packet-radio networks than has previously been shown. Our first contribution consists of showing the importance of carrier sensing in single-channel networks. We demonstrate that carrier sensing provides higher throughput than packet sensing in fully-connected networks. Our second contribution consists of showing carrier sensing, together with a CTS packet that overlaps RTS packets (which we call *CTS dominance*) is sufficient to provide collision avoidance in ad-hoc networks based on a single channel and in which nodes transmit packets asynchronously. We describe a new method of channel access control: *Floor Acquisition Multiple Access (FAMA)*. The use of carrier sensing along with the dominant CTS packet allows FAMA to perform a simple “BTMA-like” protocol in a single channel, using only one radio per node. We provide the first formal verification of collision avoidance techniques in ad-hoc networks to date. Our third contribution is the first average throughput analysis of MAC protocols based on “floor acquisition” in both fully-connected and multihop networks. True collision detection is not practical in single-channel radio devices and as such, our fourth contribution is the design and analysis of a new channel access technique we call FAMA with passive jamming (FAMA-PJ) that emulates collision detection in fully-connected packet-radio networks. Our last contribution consists of introducing a new channel access technique for multiple-channel devices called FAMA-MC which amounts to providing one floor for each network channel. The rest of this thesis is organized as follows:

Chapter 2 discusses FAMA protocols for fully-connected networks. In this chapter we describe and analyze FAMA non-persistent transmit-request (FAMA-NTR). FAMA-NTR operates in a similar fashion to the IEEE 802.11 standard for wireless local area networks. We also analyze MACA and compare it with FAMA-NTR, showing the importance carrier sensing plays in increasing throughput over simple packet-sensing.

In Chapter 3 we discuss and analyze the operation of FAMA with passive jamming (FAMA-PJ). Passive jamming is a technique whereby at least one station that is central to all others transmits a jamming signal in the channel when it detects that control packets have failed. We show that FAMA-PJ emulates collision detection while providing for data packets to be sent without interference.

Chapter 4 presents FAMA non-persistent carrier-sense (FAMA-NCS). We show the importance of combining carrier sensing with the dominance of the CTS in providing interference-free data transmissions in the presence of hidden terminals in a single-channel. In addition, we show how packet-sensing fails in the presence of hidden-terminals in networks with a single channel and asynchronous transmissions.

Chapter 5 is a discussion of FAMA for networks with multiple channels (FAMA-MC). We present our analysis for fully-connected networks and provide simulation results for operation in both fully-connected and ad-hoc networks.

The FAMA-NCS protocol has been implemented and tested in Wireless Internet Gateways (WINGs) which are wireless routers running the IP protocol stack on top of FAMA. Chapter 6 discusses the details of this implementation.

Chapter 7 summarizes the work presented here, and raises some points of interest and directions for future work.

Chapter 2

FAMA for Fully Connected Networks

In this chapter, we unify the basic schemes used in many MAC protocols for carrier sensing and collision avoidance into a new channel access discipline that we call FAMA (floor acquisition multiple access). The objective of a FAMA protocol is for a station that has data to send to acquire control of the channel (which we call the floor) before sending any data packet, and to ensure that no data packet collides with any other packet. We show that the MACA protocol and its derivatives (e.g., MACAW [BDSZ94]) become a variant of FAMA protocols when RTS and CTS transmissions last long enough. We also show that, contrary to what some previous approaches have suggested [BDSZ94, Kar90], contention avoidance should be done at both sender and receiver, and that combining carrier sensing and the RTS-CTS exchange provides a very efficient MAC protocol that performs as well as MACA under the hidden-terminal situation, and as well as CSMA otherwise.

Section 2.1 introduces two variants of FAMA protocols (MACA and FAMA-NTR). Section 2.2 demonstrates that both variants correctly enforce floor acquisition provided that RTSs and CTSs are of at least a certain minimum length. Section 2.3 analyzes the throughput of such FAMA protocols and compares them against the throughput of non-persistent CSMA. Section 2.4 discusses other related work, and Section 2.5 provides our conclusions.

2.1 FAMA Protocols

The objective of a FAMA protocol is to allow a station to acquire control of the channel (the floor) dynamically, and in such a way that no data packets ever collide with any other packet. This can be viewed as a form of dynamic reservations; however, in contrast to prior approaches to dynamic reservations, which are also called collision avoidance schemes (e.g., SRAM [TK76], MSAP [KS80] and BRAM [CFL79]), the FAMA protocols presented in this chapter do not require separate control sub-channels or preambles to reserve the channel. Instead, a FAMA protocol requires a station who wishes to send one or more packets to acquire the floor before transmitting the packet train. The floor is acquired using control packets that are multiplexed together with the data packets in the same channel in such a way that, although control packets may collide with others, data packets are always sent free of collisions.

There are many different schemes with which stations can acquire the floor, and any single-channel MAC protocol that does not require a station to sense the channel while it is transmitting can be adapted to support floor acquisition for our purposes.

A floor acquisition strategy based on an RTS-CTS exchange is particularly attractive in the control of packet-radio networks because it provides a building block to solve the hidden-terminal problem that arises in CSMA [TK75]. Within the context of using an RTS-CTS exchange for floor acquisition, there are many ways in which such control packets can be transmitted. We address only two variants in this chapter.

- RTS-CTS exchange with no carrier sensing.
- RTS-CTS exchange with non-persistent carrier sensing.

The first variant corresponds to using the ALOHA protocol for the transmission of RTS packets; the second consists of using the non-persistent CSMA protocol to transmit RTS packets. We choose to consider non-persistent carrier sensing over persistent carrier sensing, because the throughput of non-persistent CSMA is much higher under high load and only slightly lower under low load than the throughput of p-persistent CSMA [KT75].

As we show in Chapter 4 the RTS-CTS dialogue can be used as the building block to eliminate the hidden-terminal problem; however, in this chapter we focus on using such a dialogue to establish a floor acquisition discipline, and focus on single-hop networks in which no hidden terminals exist. The design of FAMA protocols for multihop packet-radio networks is addressed in Chapter 4; the basis for such protocols is the use of additional feedback from the receiver, in the form of CTSs and partial acknowledgments to packet trains.

```

Variable Definitions
   $T_{PROP}$  = Maximum channel propagation delay
   $T_{RTS}$  = Transmission time of an RTS packet
   $T_{CTS}$  = Transmission time of a CTS packet
   $T_{DATA}$  = Transmission time of a DATA packet
   $T_{TR}$  = Time to transition from transmit to receive
   $T_{MBT}$  = Minimum backoff time

Procedure START()
Begin
  call PASSIVE()
End

Procedure PASSIVE()
Begin
  While(No Packet Received  $\wedge$  No Local Packet) wait
  If(Packet Received) Then call REMOTE(received packet)
  Else call RTS()
End

Procedure RTS()
Begin
  Transmit RTS
   $Timer \leftarrow T_{CTS} + T_{TR} + 2T_{PROP}$ 
  While(Timer not expired  $\wedge$  No Packet Received) wait
  If(Timer expired) Then call BACKOFF()
  Else DO CASE of (received packet type)
  Begin
    Local CTS: call XMIT()
    Default: call REMOTE(received packet)
  End
End

Procedure BACKOFF()
Begin
  Retransmit Timer  $\leftarrow 2 \times$  Retransmit Timer
   $Timer \leftarrow RANDOM(0, Retransmit Timer)$ 
  While(Timer not expired  $\wedge$  No Packet Received) wait
  If(Timer expired) Then call PASSIVE()
  Else call REMOTE(received packet)
End

Procedure XMIT()
Begin
  Wait  $T_{TR}$ 
  Transmit Data Packet
  Retransmit Timer  $\leftarrow T_{MBT}$ 
  call PASSIVE()
End

Procedure REMOTE(packet)
Begin
  DO CASE of (packet type)
  Begin
    Local RTS:
      Wait  $T_{TR}$ 
      Transmit CTS
       $timer \leftarrow T_{DATA} + T_{TR} + 2T_{PROP}$ 
    Other:
       $timer \leftarrow T_{CTS} + T_{TR} + 2T_{PROP}$ 
    CTS:
       $timer \leftarrow T_{DATA} + T_{TR} + 2T_{PROP}$ 
    DATA:
      If(Local DATA) Then pass packet to upper layer
      call PASSIVE()
  End
  While(Timer not expired  $\wedge$  No Packet Received) wait
  If(Timer expired) Then call PASSIVE()
  Else call REMOTE(received packet)
End

```

Figure 2.1: MACA Specification

2.1.1 MACA

The first variant of FAMA that we address has been recently proposed by Karn [Kar90] and has been called MACA (Multiple Access Collision Avoidance). According to MACA, a station that has a data packet to send first transmits a request-to-send packet (RTS) to the receiver. A station that receives a complete RTS that it can understand defers transmission for an amount of time specified in the RTS; upon reception of a correct RTS that is understood by the intended receiver, the receiver sends a clear-to-send packet (CTS) and waits long enough for the data packet to arrive from the sender. Figure 2.1 specifies MACA in detail, following from Karn's original description [Kar90]. MACA and improvements over it are also discussed in detail by Bharghavan, et al. [BDSZ94]. The key aspect of this variant of FAMA protocols that is important to highlight is that, as specified

by Bharghavan, et al. [BDSZ94] and Karn [Kar90], stations do not sense the channel before transmissions. A station defers its transmission only *after* it has received and understood a *complete* RTS or CTS (just as the ALOHA protocol permits a station to send a data packet whenever it is ready). As Figure 2.2 illustrates, without proper precautions, data packets can collide with RTSs. Section 2.2 demonstrates that the duration of an RTS must be at least twice the maximum channel propagation delay in order for MACA to ensure that data packets do not collide with RTS or CTS transmissions. MACA can also be modified to permit the transmission of packet bursts by enforcing waiting periods on stations proportional to the channel propagation time; these changes are straightforward and can be derived from the specification of FAMA-NTR, described next.

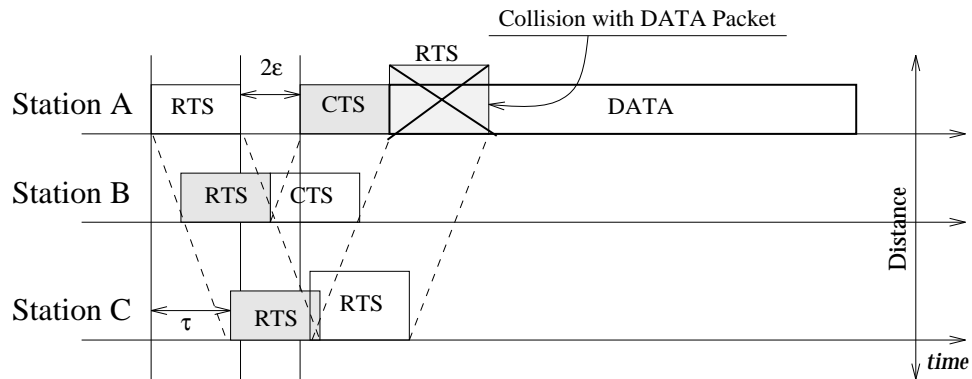


Figure 2.2: MACA unsafe transmission:

An RTS from *C* collides with *A*'s data packet due to differences in propagation time from *A* to *B* and from *A* to *C* and the length of RTS and CTS packets.

2.1.2 FAMA-NTR (Non-persistent Transmit Request)

The second variant of FAMA, which we call FAMA-NTR (Non-persistent Transmit Request) combines non-persistent carrier sensing with the RTS-CTS exchange of MACA.

```

Variable Definitions
  CD = Carrier Detected
  TPROP = Maximum channel propagation delay
  TPROC = Processing time for carrier detection
  TTR = Transmit to receive turn-around time
  Burst = Number of packets to send in a burst

Procedure START()
Begin
  Timer ← 2 × TPROP
  While( $\overline{CD}$  ∧ Timer not expired) wait
  If (CD) Then call REMOTE(2 × TPROP + TTR + TPROC)
  Else call PASSIVE()
End

Procedure PASSIVE()
Begin
  While( $\overline{CD}$  ∧ No Local Packet) wait
  If (CD) Then call REMOTE(2 × TPROP + TTR + TPROC)
  Else call RTS(2 × TPROP + TTR + TPROC)
End

Procedure RTS(Tσ)
Begin
  Transmit RTS Packet
  Timer ← Tσ
  While( $\overline{CD}$  ∧ Timer not expired) wait
  If (Timer Expired) Then call BACKOFF()
  Else Begin
    Receive Packet
    DO CASE of (received packet type)
    Begin
      CTS: call XMIT()
      Default: BACKOFF()
    End
  End
End

Procedure BACKOFF()
Begin
  Timer ← RANDOM(0,10 × TRTS)
  While( $\overline{CD}$  ∧ Timer not expired) wait
  If (CD) Then call REMOTE(2 × TPROP + TTR + TPROC)
  Else call RTS(2 × TPROP + TTR + TPROC)
End

Procedure XMIT()
Begin
  Burst ← maximum burst
  Wait TTR
  While ((Burst > 0) ∧ Local Packet)
  Do Begin
    Transmit Data Packet
    Burst ← Burst - 1
  End
  Timer ← TPROP + TTR
  While(Timer not expired) wait
  If (Local Packet) Then call BACKOFF()
  Else call PASSIVE()
End

Procedure REMOTE(Tσ)
Begin
  Timer ← Tσ
  While( $\overline{CD}$  ∧ Timer not expired) wait
  If (Timer Expired)
  Then Begin
    If (Local Packet) Then call BACKOFF()
    Else call PASSIVE()
  End
  Else Begin
    Receive Packet
    DO CASE of (received packet type)
    Begin
      RTS:
        If(Destination ID = Local ID)
        Then Begin
          Wait TTR
          Transmit CTS Packet
        End
        call REMOTE(2 × TPROP + TTR + TPROC)
      CTS: call REMOTE(2 × TPROP + TTR + TPROC)
      DATA:
        If(Destination ID = Local ID)
        Then pass packet to upper layer
        call REMOTE(TPROP + TTR + TPROC)
      ERROR: REMOTE(2 × TPROP + TTR + TPROC)
    End
  End
End

```

Figure 2.3: FAMA-NTR Specification

Figure 2.3 specifies FAMA-NTR in detail. When a station has one or multiple packets to deliver, it first listens to the channel. If the channel is busy, the station backs off and tries to retransmit at a later time using a random value for the backoff time; if the channel is clear (i.e., no carrier is detected), the station transmits an RTS. The sender listens to the channel for one round-trip time plus the time needed for the destination to send a CTS. If the CTS packet is corrupted or is not received within the time limit, the sender goes into the backoff state and tries to retransmit at a later time. When the originator receives the CTS from the destination, it begins its transmission of the data packet burst. The burst is limited to a maximum number of data packets, after which the station must release the

channel and contend to re-acquire the floor. This variant of FAMA is similar to the protocol proposed for IEEE 802.11 [Bib92], and Apple's Local Talk Link Access protocol [SAO90].

FAMA-NTR enforces a waiting period on stations at strategic points in the operation of the protocol. Receiving stations (those stations in the REMOTE state) have a required waiting period of τ seconds after processing a data packet, to allow the current transmitting station the capability to send a burst of packets once it acquires the floor. A receiving stations' waiting period for any control packet is 2τ seconds; this is done to allow the RTS-CTS exchange to take place (see the timing for Station A during the successful transmission period in Figure 2.5).

Transmitting stations in the RTS state require a waiting period of 2τ seconds after transmitting their RTS to allow the destination to receive the RTS and transmit the corresponding CTS. A sending station must also use a waiting period of τ seconds after a final data packet to allow the destination to receive the complete packet and to account for the enforced waiting time at the destination. After the waiting period expires (assuming no further transmission on the channel, specifically, after a transmission period) all stations transition either to the PASSIVE state (if they have no packets pending) or the BACKOFF state (if a local packet is pending delivery). The channel becomes idle when all stations are in either the PASSIVE or BACKOFF state. The next access to the channel is driven by the arrival of new packets to the network and retransmission of packets that have been backed off.

Different backoff strategies can be adopted in the versions of FAMA addressed in this chapter (e.g., see the one proposed for MACAW [BDSZ94]). Exact distributions of the retransmission times are not necessary in our throughput analysis, which simply assumes

that retransmissions are, on the average, long enough to make them independent of the original arrival of packets for transmission. Furthermore, stability and optimization of the channel (e.g, [ML87]) are not addressed in this chapter.

The three-way handshake (i.e., RTS-CTS exchange followed by data packets) assumed in FAMA-NTR can also be extended to include an acknowledgment by the receiver after processing the last packet in the packet train. This four-way handshake is part of IEEE 802.11 and has also been proposed in MACAW.

2.2 Floor Assignment in FAMA along a Single Hop

For FAMA protocols to work correctly, they must ensure that all data packets delivered to the channel reach their proper destination without collisions. Theorems 1 and 2 below show this under the following assumptions:

- A1) The maximum end-to-end propagation time in the channel is $\tau < \infty$.
- A2) A packet sent over the channel that does not collide with other transmissions gets delivered free of errors to a station with probability $p > 0$.
- A3) A station transmits an RTS that does not collide with other transmission with probability $q > 0$.
- A4) All stations are within one maximum propagation delay (τ) of all other stations, i.e., there are no hidden terminals.
- A5) All stations execute a FAMA protocol correctly.

A6) The transmission time of an RTS or CTS packet is γ , the transmission time of a data packet is δ , and the processing time is t_p , where $\gamma \leq \delta < \infty$, and $t_p < \infty$.

Theorem 1 *FAMA-NTR ensures that each new data packet, or any of its retransmissions, is sent to the channel within a finite time after it becomes ready for transmission, and that a data packet does not collide with any other transmission, provided that $\tau < \gamma < \infty$.*

Proof: By this theorem's assumption, an RTS lasts longer than the channel propagation time. Therefore, if an arbitrary station A is able to send its RTS to station B without colliding with other transmissions, all other stations must detect carrier before A ends transmitting its RTS and must enter the remote state, which forces them to enter a waiting period of longer than 2τ seconds (2 times the propagation time plus a maximum processing time, t_p) after detecting the end of station A 's RTS transmission. Because the maximum channel propagation time is τ , a station other than B can receive A 's RTS at most τ seconds before B does.

Therefore, given that station B 's CTS can take at most τ seconds to reach all stations, the backoff time used in the remote state is long enough to make every station backoff again for station B 's CTS, allowing only station A to use the channel if it receives station B 's CTS with no errors. Accordingly, it follows that if an arbitrary station A sends a packet i to the channel, such a packet can collide with no other packet.

Let t_0 be the time when, in order to transmit packet i (which can be a new data packet or a retransmission), an arbitrary station A sends an RTS to station B . From our assumptions there must be a time t_1 such that $t_0 \leq t_1 < \infty$ when A sends an RTS that

- (a) forces all stations other than A to enter the remote state by time $t_2 = t_1 + \tau < \infty$ after detecting a carrier in the channel,
- (b) causes station B to send a CTS to station A by time $t_3 = t_2 + \gamma < \infty$,
- (c) makes station A start processing a CTS from B by time $t_4 = t_3 + \tau < \infty$ and send packet i by time $t_5 = t_4 + \gamma + t_p < \infty$.

It follows that, for any given packet, any station takes a finite time to send the packet in the channel, and that such packet does not collide with any other packets. \square

In MACA a station must understand a packet before deferring transmissions and it takes up to τ seconds for a transmission to reach all stations. Therefore, a station (call it C) may begin an RTS up to τ seconds after another station (call it A) has finished sending its RTS request intended for another station (call it B). In addition, the beginning of the RTS transmission from station C can take up to τ seconds to reach station A . Therefore, there is a maximum period of 2τ seconds between the end of stations A 's RTS and the beginning of an RTS from C . If station B is very close to station A , it will respond with its corresponding CTS in a very short time ($\epsilon \ll \tau$) after the complete reception and processing of the RTS from A ; in turn, this CTS will arrive at station A in ϵ seconds and the data packet from A will begin immediately after the processing of the CTS from B . As $\epsilon \rightarrow 0$, if $\gamma \leq 2\tau$, it is possible for station A to receive a correct CTS from B and send a data packet within 2τ seconds after the end of its RTS. This data packet collides with the RTS from C , which does not arrive at A until 2τ seconds after the end of A 's RTS. Figure 2.2 illustrates this situation.

Theorem 2 *MACA ensures that data packets do not collide with any other transmissions, provided that $2\tau < \gamma < \infty$.*

Proof: Given a fully connected network of stations, consider a station A sending data to station B , and an interfering station C . If $\gamma > 2\tau$ (as shown in Figure 2.4), it is guaranteed that, at station A , the CTS sent by B to A will collide with station C 's RTS. Here, stations A and B are close neighbors (B receives A 's complete RTS in ϵ seconds, with $\epsilon \rightarrow 0$), and station C receives A 's RTS in exactly τ seconds and B 's transmission in at most τ seconds. After station A completes its clear transmission of an RTS to station B , B receives the entire RTS in ϵ more seconds, when it sends its CTS. The end of the CTS from B reaches A ϵ seconds after B stops its transmission. For station C to be able to begin transmitting its own RTS after A has started its RTS, station C must transmit in at most τ seconds after the completion of A 's RTS, just before understanding A 's RTS. The RTS from C reaches A in at most τ seconds (2τ seconds after the completion of A 's RTS) and must collide with the CTS from B – even if $\epsilon = 0$ – because $\gamma > 2\tau$, causing the RTS-CTS exchange between A and B to fail and A to backoff and retry later. It follows that, if $\gamma > 2\tau$, station A cannot send a data packet if any other station starts an RTS within τ seconds of the end of A 's RTS. Furthermore, every station must understand A 's RTS in at most τ seconds if no other station sends an RTS before that time. Therefore, the theorem is true. \square

Under the conditions demonstrated in Theorems 1 and 2, both MACA and FAMA-NTR assign the channel dynamically to different stations in such a way that data packets are always sent in the clear. However, as the next section shows, using carrier sensing together with the RTS-CTS exchange provides substantial performance improvements over

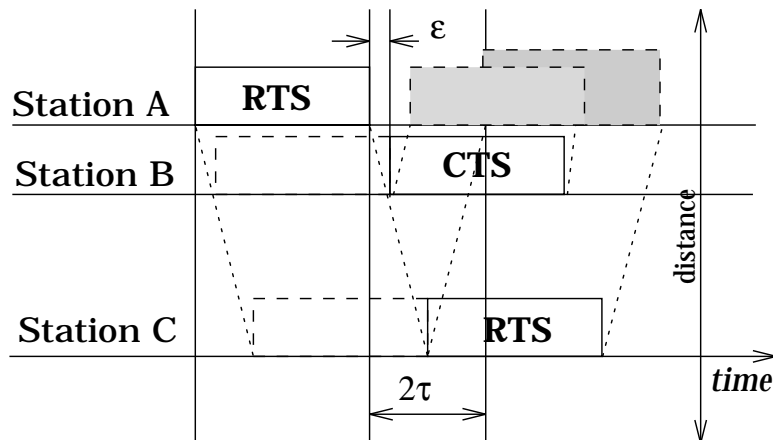


Figure 2.4: MACA RTS/CTS collision

the basic MACA scheme.

Theorems 1 and 2 apply to the case in which stations transmit asynchronously. Alternatively, a global clock can be used to force all stations to start packet transmissions at the beginning of time slots.

In slotted MACA, the duration of a time slot should equal one RTS duration plus one maximum propagation delay, with all transmissions being of lengths that are multiples of such a slot time. With such slotting, any control packet transmitted at the beginning of a given slot is received in its entirety before another station is allowed to start transmitting any packet it has scheduled for transmission during the same slot. Therefore, if a station A sends an RTS during slot i , any other station scheduling an RTS transmission for slot $i + 1$ must defer its transmission after receiving the intended RTS from A by the end of slot i . Accordingly, collisions of data packets and RTSs cannot occur in slotted MACA, and slotted MACA constitutes a variant of FAMA.

Slotting can also be applied in FAMA-NTR; i.e., in this case, the duration of a slot equals the maximum propagation delay and all packets have a duration that is a multiple of a slot duration. Therefore, if a station sends an RTS at the beginning of slot i , any station scheduling an RTS transmission for slot $i + 1$ must detect carrier by the beginning of that slot and defer transmission. Accordingly, even if an RTS lasts τ seconds, data packets cannot collide with RTSs.

The above shows that the size of the RTS and CTS packets in relation to the data packets is critical to the efficient operation of a FAMA protocol. If the size of RTS and CTS packets approaches the size of the data packets, the overhead of the contention period will degrade the performance considerably. Therefore, RTS and CTS packets must be kept as small as possible compared to the size of data packets, while ensuring that RTS and CTS packets last longer than the maximum propagation time across the network when no slotting is used.

2.3 Approximate Throughput Analysis

We present an approximate throughput analysis that assumes the same traffic model first introduced by Tobagi and Kleinrock [KT75] to analyze the throughput of CSMA protocols, and the conditions for floor acquisition derived in Section 2.2. The protocols we analyze are non-persistent CSMA, MACA, FAMA-NTR, and the slotted versions of these FAMA protocols. The throughput of non-persistent CSMA used in this analysis has been previously reported [KT75].

2.3.1 Assumptions and Notations

There is an infinite number of stations that constitute a Poisson source sending RTS packets (for the case of FAMA), or new or retransmitted data packets (for the case of CSMA) to the channel with an aggregate mean generation rate of λ packets.

Each station is assumed to have at most one data block to be sent at any time. In all protocols, a station transmits the entire data block as a single packet (which is the case of CSMA and MACA [Kar90]) or as multiple packets (which is the case of FAMA-NTR). The average transmission time of a data block is δ seconds. RTS and CTS packets are of size γ seconds, and the maximum end-to-end propagation delay of the channel is τ seconds. Collisions (e.g., RTS packets in FAMA-NTR, data packets in CSMA) can occur in the channel, and we assume that, when a station has to retransmit a packet, it does so after a random retransmission delay that is much larger than δ on the average. The average channel utilization is given by [KT75]

$$S = \frac{\bar{U}}{\bar{B} + \bar{I}} \quad (2.1)$$

where \bar{B} is the expected duration of a busy period, defined to be a period of time during which the channel is being utilized; \bar{I} is the expected duration of an idle period, defined as the time interval between two consecutive busy periods; and \bar{U} is the time during a busy period that the channel is used for transmitting user data successfully.

The channel is assumed to introduce no errors, so packet collisions are the only source of errors, and stations detect such collisions perfectly. To further simplify the problem, we assume that any station can listen to the transmissions of any other station, that two or more transmissions that overlap in time in the channel must all be retransmitted, and

that a packet propagates to all stations in exactly τ seconds [KT75]. The later assumption provides a lower bound on the performance of the protocols we analyze.

The time stations take to transition from transmit to listening mode and from listening to transmit mode is assumed to be negligible. This assumption is in agreement with implementation parameters in IEEE 802.11 [IEE97]. When such turn-around times are not negligible, it is easy to show that the only impact on our approximated model is an increase in the effective duration of transmissions in the channel.

Of course, this model is only a rough approximation of the real case, in which a finite number of stations access the same channel, some stations may not be able to hear some other stations' transmissions, stations can queue multiple packets for transmission, and the stations' transmissions and retransmissions (of RTS or data packets) are highly correlated (e.g., a failed RTS is followed by another RTS within a bounded time, and a data packet is always preceded by a successful RTS). However, our analysis helps to understand why it is beneficial to listen for any type of channel activity, rather than for specific packet types, and provides additional insight on the performance of the FAMA protocols and the impact of channel speed, propagation delay, and hidden terminals on the floor acquisition technique. Our analysis favors CSMA and MACA, in that we assume that the applications accessing the channel can efficiently use data packets that are much longer than an RTS.

Insofar as the hidden-terminal problem is concerned, our analysis provides only an approximation of the performance that a modified FAMA-NTR would have in the two extreme cases in which either all or none of the sender-receiver interactions are affected by it. More specifically, we assume that every station can listen to the transmissions of any other station and our analysis of FAMA-NTR corresponds to the case in which no

hidden terminals exist. However, when the sender of an RTS is unable to sense another station's RTS, a modified FAMA-NTR that can provide floor acquisition over multiple hops should behave much like MACA with no hidden terminals; therefore, MACA's throughput represents the worst case of a modified FAMA-NTR when all senders experience hidden-terminal problems.

2.3.2 FAMA-NTR

Figure 2.5 shows the transmission periods of FAMA-NTR. A transmission period begins with a source station transmitting an RTS at some time t_0 . The transmission is vulnerable for a period of τ seconds, during which another RTS from some other station may collide with it, causing the transmissions to fail. After the vulnerability period, if no other station has transmitted, all other stations will sense the channel busy, defer their transmissions, and the RTS transmission will be successful. According to FAMA-NTR, the RTS is followed by the CTS response from the destination and the data packet(s) from the source. As Figure 2.5 illustrates, because of the enforced waiting times and idle periods discussed in Section 2.1.2, a FAMA-NTR busy period is exactly one transmission period in length, either a successful or failed transmission, followed by an idle period.

Theorem 3 *The throughput of FAMA-NTR is given by*

$$S = \frac{\delta}{\gamma + \delta + \frac{(2 - e^{-\tau\lambda})}{\lambda} + e^{\tau\lambda}(\gamma + 4\tau)} \quad (2.2)$$

Proof: A successful transmission consists of an RTS with one propagation delay to the intended recipient, a CTS and propagation delay back to the sender, and the data packet

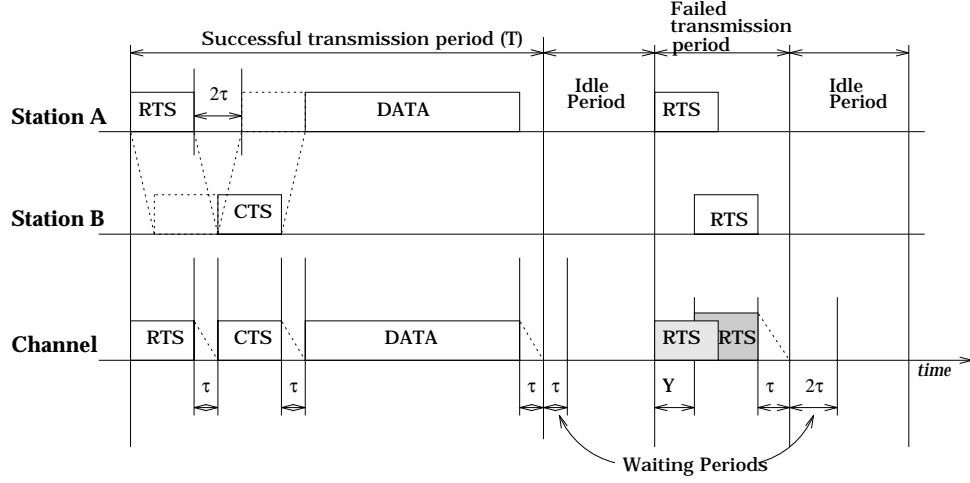


Figure 2.5: FAMA-NTR transmission periods

followed by a propagation delay. The time for a successful transmission, T , is then

$$T = 2\gamma + 3\tau + \delta \quad (2.3)$$

Because FAMA-NTR guarantees that data packets sent after a successful RTS will not collide with any other packet (Theorem 1), an unsuccessful transmission will consist of one RTS being sent to the channel at time t_0 followed by one or more RTSs transmitted by other stations within time Y (see Figure 2.5), where $0 \leq Y \leq \tau$, plus one final propagation delay. Therefore, as in non-persistent CSMA [KT75], the duration of the average failed transmission period is given by

$$T_{FAIL} = \gamma + \tau + \bar{Y} \quad (2.4)$$

The cumulative distribution function for Y is the probability that no arrivals occur in the interval of length $\tau - y$ and equals $F_Y(y) = e^{-\lambda(\tau-y)}$ [KT75] (where $y \leq \tau$); therefore, the expected value of Y is

$$\bar{Y} = \tau - \frac{(1 - e^{-\tau\lambda})}{\lambda} \quad (2.5)$$

Substituting \bar{Y} in (2.4), we obtain

$$T_{FAIL} = \gamma + 2\tau - \frac{(1 - e^{-\tau\lambda})}{\lambda} \quad (2.6)$$

The probability of success for an RTS equals the probability that no arrivals occur in τ seconds, because there is a delay across the channel of τ seconds before all the other stations in the network detect the carrier signal. After this vulnerability period of τ seconds, all stations detect the carrier signal in the channel and defer their own transmissions. Therefore, given that arrival of RTSs to the channel are Poisson with parameter λ ,

$$P_S = P\{\text{No arrivals in } \tau \text{ seconds}\} = e^{-\tau\lambda} \quad (2.7)$$

Because each FAMA-NTR busy period is always either a single successful or failed transmission period, the average busy period can be expressed as the percentage of successful transmission periods times the duration of T , plus the percentage of unsuccessful transmission periods times their average duration T_{FAIL} . Therefore,

$$\begin{aligned} \bar{B} &= T \cdot P_S + T_{FAIL} \cdot (1 - P_S) \\ &= e^{-\tau\lambda}(\gamma + \delta + \tau + \frac{(1 - e^{-\tau\lambda})}{\lambda}) + \gamma + 2\tau - \frac{(1 - e^{-\tau\lambda})}{\lambda} \end{aligned}$$

The average utilization is the average amount of time during which useful data are sent during a successful busy period; therefore,

$$\bar{U} = \delta \cdot P_S = \delta e^{-\tau\lambda} \quad (2.8)$$

According to FAMA-NTR's definition, stations must incur a fixed time waiting period after each transmission period on the channel before making the transition to the PASSIVE or BACKOFF state (Figure 2.3). If the transmission period is a successful data

packet, then the waiting period is τ ; otherwise the waiting period is 2τ . Because the waiting period is directly related to the transmission period preceding it, the average waiting period can be expressed as the percentage of successful transmissions with a waiting period of length τ , plus the percentage of failed transmissions with waiting periods of length 2τ . Therefore, the average idle time \bar{I} can be expressed by

$$\begin{aligned}\bar{I} &= \frac{1}{\lambda} + \tau \cdot P_S + 2\tau \cdot (1 - P_S) \\ &= \frac{1}{\lambda} + \tau e^{-\tau\lambda} + 2\tau(1 - e^{-\tau\lambda})\end{aligned}\quad (2.9)$$

Substituting Eqs. (2.8), (2.8) and (2.9) in (2.1), we obtain Eq. (2.2). \square

2.3.3 Slotted FAMA-NTR

We consider slotted FAMA-NTR with the assumptions that the slot size equals the propagation delay τ , and that the duration of RTS, CTS and data packets are all exact multiples of τ . With slotting, stations are restricted to start transmissions only at slot boundaries. Figure 2.6 shows the transmission periods for slotted FAMA-NTR; arrivals of RTSs scheduled for transmission in the channel at the beginning of the next slot are indicated by vertical arrowheads. As in FAMA-NTR, slotted FAMA-NTR enforces a waiting period after each transmission period. A waiting period of τ seconds is required after a data packet is received, and a 2τ waiting period is required after any other transmission is heard on the channel. Again, as in FAMA-NTR, this limits the busy period to exactly one successful or failed transmission period.

Theorem 4 *The throughput of slotted FAMA-NTR is given by*

$$S = \frac{\delta\lambda\tau e^{-\lambda\tau}}{\lambda\tau e^{-\lambda\tau}(\gamma + \delta + \tau) + (1 - e^{-\lambda\tau})(\gamma + 3\tau) + \tau}\quad (2.10)$$

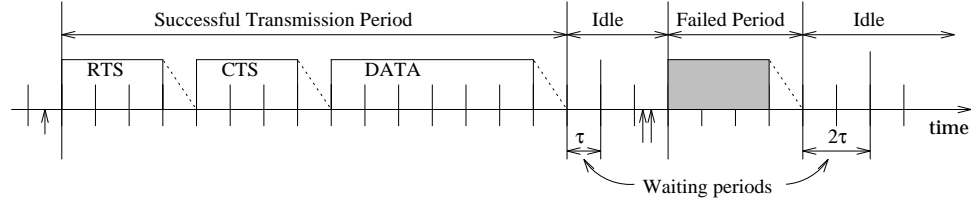


Figure 2.6: Slotted FAMA-NTR transmission periods

Proof: A successful transmission period (T) is made up of a clear RTS followed by a CTS and data packet. Therefore, T is given by Eq. (2.3).

A failed transmission period consists of one or more stations detecting no carrier and sending an RTS during a given slot. The failed period is the length of one RTS, and a slot used at the end for propagation delay. The total time of a failed transmission period is

$$T_{FAIL} = \gamma + \tau \quad (2.11)$$

For an RTS to be successful, it must be the only packet in the channel during its transmission. The probability of an RTS being sent in the clear is

$$\begin{aligned} P_S &= \text{P}\{\text{One arrival in a slot} \mid \text{Some arrivals in the slot}\} \\ &= \frac{\lambda\tau e^{-\lambda\tau}}{1 - e^{-\lambda\tau}} \end{aligned} \quad (2.12)$$

A busy period is made up of both successful and failed transmission periods. Because FAMA-NTR forces an idle period between each transmission (successful or failed) period, the duration of an average busy period equals the sum of the average transmission period size (which equals the percentage of successful transmission periods times their duration T), plus the percentage of unsuccessful transmission periods times their duration, T_{FAIL} . Accordingly,

$$\overline{B} = T \cdot P_S + T_{FAIL} \cdot (1 - P_S) \quad (2.13)$$

Substituting Eqs. (2.3), (2.11) and (2.12) into Eq. (2.13) gives

$$\bar{B} = (\gamma + \delta + 2\tau) \left[\frac{\lambda\tau e^{-\lambda\tau}}{(1 - e^{-\lambda\tau})} \right] + \gamma + \tau \quad (2.14)$$

The utilization of the channel is the data portion of the successful transmission period. Therefore, because a transmission period is successful with probability P_S and the data portion of such period is δ , we obtain

$$\bar{U} = \delta \cdot P_S = \delta \frac{\lambda\tau e^{-\lambda\tau}}{(1 - e^{-\lambda\tau})} \quad (2.15)$$

The idle period consists of consecutive idle slots preceded by the enforced waiting period after each transmission period, as defined in the FAMA-NTR specification. The number of consecutive idle slots has a geometric distribution whose mean is the same as that derived for non-persistent CSMA [KT75, RS90] and is equal to $1/(1 - e^{-\lambda\tau})$. The average idle period is equal to the average number of consecutive idle slots plus the average enforced waiting period. Therefore,

$$\begin{aligned} \bar{I} &= \tau \cdot \left(\frac{1}{1 - e^{-\lambda\tau}} \right) + (\tau \cdot P_S + 2\tau \cdot (1 - P_S)) \\ &= \frac{\tau - \lambda\tau^2 e^{-\lambda\tau}}{(1 - e^{-\lambda\tau})} + 2\tau \end{aligned} \quad (2.16)$$

Substituting Eqs. (2.14), (2.15) and (2.16) into Eq. (2.1) we obtain Eq. (2.10). \square

2.3.4 MACA

Figure 2.7 shows the transmission periods in MACA under the assumption that $\gamma > 2\tau$. Note that, because a station using MACA does not enforce any waiting times after transmission periods (see [BDSZ94] and Figures 2.2, 2.4 and 2.7), the RTS and CTS specify how long stations should defer [Kar90]. MACA does not use carrier sensing before

transmitting an RTS, and a station can start transmitting an RTS (or CTS) even while another RTS has reached the station but has not been received in its entirety (this is similar to the operation of ALOHA [Abr70]). However, a station that understands a clear RTS from another station defers its own transmission for the duration of the balance of a successful transmission period. Following this deferment, there is a random waiting period before transmission begins again. The random waiting time enforces an idle period after a successful transmission, the same as in FAMA-NTR. An unsuccessful period is also followed by an idle period, because any transmission attempt during (or adjacent to) the failed period would be included as part of the unsuccessful period. Therefore, it follows that a MACA busy period is limited to either a single successful transmission period, or a failed transmission period.

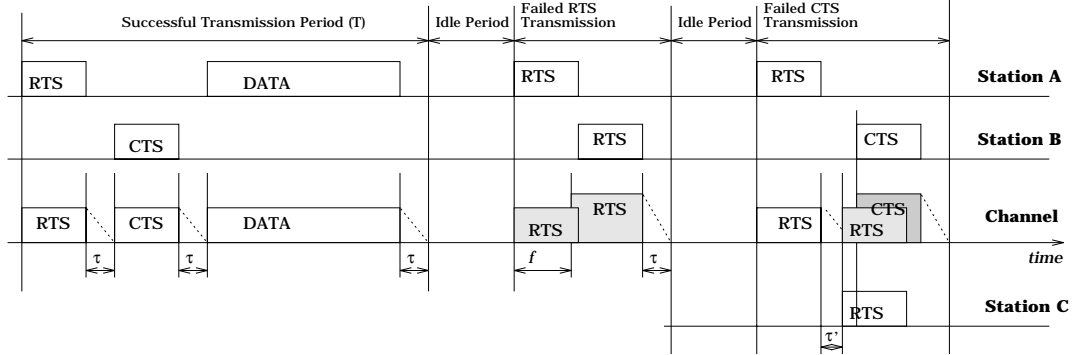


Figure 2.7: MACA transmission periods

Theorem 5 *The throughput of MACA is given by*

$$S = \frac{\delta}{e^{\lambda(2\gamma+\tau)} \left[\gamma + \tau + \frac{1}{\lambda} + F \right] + e^{\lambda\tau} \left[\gamma + \frac{\tau}{2} + P(\tau - F) \right] + \delta + \frac{3\tau}{2} + F + P(\tau - F)}$$

$$\text{where } F = \left[\frac{e^{\lambda\gamma} - 1 - \lambda\gamma}{\lambda\gamma(1 - e^{-\lambda\gamma})} \right]; P = \left[\frac{e^{-\lambda\gamma} - e^{-\lambda(\gamma+\tau)}}{(1 - e^{-\lambda(\gamma+\tau)})} \right] \quad (2.17)$$

Proof: A successful transmission includes the RTS, CTS and data packet with a delay of τ seconds across the channel. Therefore, the size of a successful transmission is given by Eq. (2.3).

As stated above, a busy period is formed by a single transmission period. Under the assumptions that every packet takes τ seconds to reach all stations and that $\gamma > 2\tau$, RTSs and CTSs do not collide with data packets (Theorem 2), and an unsuccessful transmission period is made up of colliding RTSs and CTSs only. A failed period can take one of two possible scenarios in MACA. In the first case, the RTS that starts the busy period collides with one or more RTSs from other stations; in the second case, an RTS is received in the clear by the intended destination, but during the τ seconds of propagation delay incurred by the RTS, and prior to understanding the RTS, at least one other station has an arrival and transmits an RTS of its own that collides with the CTS sent in response to the first RTS of the busy period. In both cases, the length of the average failed transmission period is unbounded. In the first case, the length of a failed transmission period T_{FRTS} consists of only RTSs. In the second case, the average length of the failed period (T_{FCTS}) consists of an RTS; the average time of an RTS arrival within an interval of τ seconds after the end of the first RTS (τ'); a period of either failed RTSs (in which case its average is identical to T_{FRTS}), or if no RTS arrives once the CTS of the period begins, the time needed for a CTS to clear the channel.

Figure 2.8 illustrates in more detail the MACA failed RTS transmission period. The transmission period shown consists of four failed RTS packets; the time periods $f1$, $f2$, $f3$ are the interarrival times of the failed RTS packets. An average failed transmission period consists of a geometrically-distributed indefinite number (L) of interarrival times

whose average duration is \bar{f} seconds (the average time between failed arrivals), plus the duration of an RTS (γ) and τ seconds of propagation time. This is exactly the same as in pure ALOHA! The values for \bar{L} and \bar{f} have been previously derived [TK85] for pure ALOHA as functions of λ and, according to our notation, δ . Substituting γ for δ in such results we obtain $e^{\lambda\gamma}$ and $(\lambda\gamma)^{-1} - e^{-\lambda\gamma}/(1 - e^{-\lambda\gamma})$, respectively. Therefore, when the first RTS of the period collides with other RTSs, the average time of a failed transmission period, T_{FRTS} , equals

$$T_{FRTS} = \left[\frac{e^{\lambda\gamma} - 1 - \lambda\gamma}{\lambda\gamma(1 - e^{-\lambda\gamma})} \right] + \gamma + \tau \quad (2.18)$$

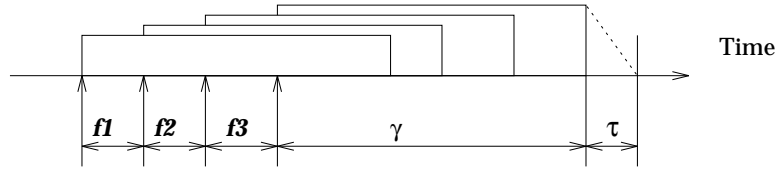


Figure 2.8: A failed RTS transmission period in MACA

The probability that a failed CTS transmission period ends when the failed CTS has cleared the channel is the probability that no other RTSs arrive to the channel once the CTS begins. This is the probability that there are no arrivals in γ seconds (the CTS duration) given that there has been at least one RTS arrival in $\gamma + \tau$ seconds (the time between the end of the RTS that started the period and the end of the corresponding CTS). Therefore,

$$\begin{aligned} P_{FCR} &= \frac{P\{\text{No arrivals in } \gamma\} \cdot P\{\text{at least one arrival in } \tau\}}{P\{\text{at least one arrival in } (\gamma + \tau)\}} \\ &= \frac{e^{-\lambda\gamma} \cdot (1 - e^{-\lambda\tau})}{(1 - e^{-\lambda(\gamma+\tau)})} \end{aligned} \quad (2.19)$$

Because the arrival process is Poisson, arrival times during any given time interval are independent and uniformly distributed [Tri88], which implies that, on the average, τ' equals

$\tau/2$. Therefore the average length of a failed CTS transmission period is,

$$T_{FCTS} = \gamma + P_{FCR}(\gamma + 2\tau) + (1 - P_{FCR}) \cdot (T_{FRTS} + \tau/2) \quad (2.20)$$

The probability of a successful transmission period (P_S) is the probability that a data packet is sent over the channel. This can happen only if an RTS and its corresponding CTS are transmitted without collisions. An RTS is sent in the clear if no other RTS is sent within γ seconds before or after it starts. Because that RTS takes τ seconds to reach all stations, its corresponding CTS is sent in the clear if no RTS is sent within τ seconds after the RTS. Therefore,

$$P_S = P\{\text{No RTS arrivals in } 2\gamma + \tau\} = e^{-\lambda(2\gamma + \tau)} \quad (2.21)$$

The probability that an RTS fails is simply the probability that RTS arrivals occur within the transmission time of another RTS, i.e., $P_{FRTS} = 1 - e^{-2\lambda\gamma}$.

The probability that a CTS fails is the probability that an RTS succeeds and at least one RTS is sent within τ seconds after the end of that RTS; therefore, colliding with the corresponding CTS, i.e., $P_{FCTS} = e^{-2\gamma\lambda}(1 - e^{-\lambda\tau})$.

Because a MACA busy period can be only a single successful transmission, or any of two types of unsuccessful transmission periods. Accordingly,

$$\bar{B} = T \cdot P_S + T_{FRTS} \cdot P_{FRTS} + T_{FCTS} \cdot P_{FCTS} \quad (2.22)$$

Substituting P_S , P_{FRTS} , P_{FCTS} , T , T_{FRTS} and T_{FCTS} into Eq. (2.22) we obtain

$$\begin{aligned} \bar{B} = e^{-\lambda(2\gamma + \tau)} & \left[\delta + \frac{3\tau}{2} - \left[\frac{e^{\lambda\gamma} - 1 - \lambda\gamma}{\lambda\gamma(1 - e^{-\lambda\gamma})} \right] \right. \\ & \left. - \frac{e^{-\lambda\gamma} \cdot (1 - e^{-\lambda\tau})}{(1 - e^{-\lambda(\gamma + \tau)})} \left(\tau - \left[\frac{e^{\lambda\gamma} - 1 - \lambda\gamma}{\lambda\gamma(1 - e^{-\lambda\gamma})} \right] \right) \right] \end{aligned}$$

$$\begin{aligned}
& +e^{-2\gamma\lambda} \left[\gamma + \frac{\tau}{2} + \frac{e^{-\lambda\gamma} \cdot (1 - e^{-\lambda\tau})}{(1 - e^{-\lambda(\gamma+\tau)})} \left(\tau - \left[\frac{e^{\lambda\gamma} - 1 - \lambda\gamma}{\lambda\gamma(1 - e^{-\lambda\gamma})} \right] \right) \right] \\
& + \left[\frac{e^{\lambda\gamma} - 1 - \lambda\gamma}{\lambda\gamma(1 - e^{-\lambda\gamma})} \right] + \gamma + \tau
\end{aligned} \tag{2.23}$$

Because all arrivals to the channel, either new or retransmitted, are preceded by an RTS, the average idle period (\bar{I}) for MACA is equal to the average interarrival time of RTSs, i.e., $\frac{1}{\lambda}$. As in the case of FAMA-NTR, $\bar{U} = \delta \cdot P_S$. Substituting Eq. (2.21) in \bar{U} we obtain

$$\bar{U} = \delta e^{-\lambda(2\gamma+\tau)} \tag{2.24}$$

Substituting \bar{U} , \bar{I} , and \bar{B} into Eq. (2.1) we obtain Eq. (2.17). \square

2.3.5 Slotted MACA

The operation of slotted MACA is similar to MACA, except that a station that receives a packet to be sent cannot start its transmission until the next time slot. We assume that the duration of a slot in slotted MACA equals the size of an RTS or CTS packet (γ) plus a propagation delay τ . Figure 2.9 shows the transmission periods in slotted MACA versus time.

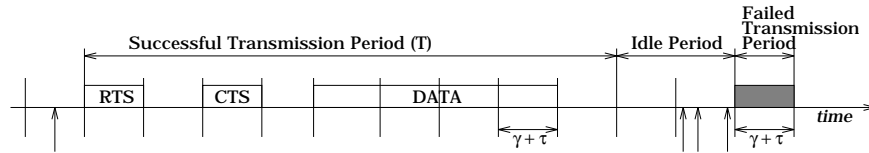


Figure 2.9: Slotted MACA transmission periods

Theorem 6 *The throughput of slotted MACA is given by*

$$S = \frac{\delta}{\delta + 4(\gamma + \tau) + \frac{e^{\lambda(\gamma+\tau)}}{\lambda}} \tag{2.25}$$

Proof: The probability of success (P_S) equals the probability of having only one RTS in a given slot, given that there is a busy period, i.e.,

$$\begin{aligned} P_S &= P\{1 \text{ arrival in a slot} \mid \text{at least one arrival in a slot}\} \\ &= \frac{(\gamma + \tau)\lambda e^{-(\gamma + \tau)\lambda}}{(1 - e^{-\lambda(\gamma + \tau)})} \end{aligned}$$

The duration of an average successful transmission period (T) equals the number of slots used to resolve contention successfully, plus the slots containing the data packet (see Figure 2.9), i.e.,

$$T = \delta + 5(\gamma + \tau) \quad (2.26)$$

where $5(\gamma + \tau)$ accounts for: an RTS slot followed by the empty slot needed for the destination to schedule the CTS; the CTS slot, also followed by a slot for the originator to schedule the data packet; and an empty slot after the data packet where requests for the next period may occur. A successful transmission period can only begin if the RTS packet is sent collision free (i.e., the RTS is the only packet transmitted during a given slot).

Because we assume a slot size to be $\gamma + \tau$, all stations will hear a clear RTS before the next slot begins, and defer. In addition, RTS packets will only collide with other RTS packets in the same slot, and not CTS packets, or data packets. Therefore, a failed transmission period (T_{FAIL}) lasts $(\gamma + \tau)$ seconds (i.e., one RTS packet time plus the propagation delay, or one slot).

The probability that a busy period consists of l slots equals the probability that at least one arrival must be scheduled for transmission in the first $l - 1$ slots and no arrivals can be scheduled for transmission in the last slot. This is geometrically distributed and

equals

$$P\{\text{busy period has } l \text{ slots}\} = (1 - e^{-\lambda(\gamma+\tau)})^{l-1} e^{-\lambda(\gamma+\tau)}$$

Therefore, the average number of slots a busy period contains is $\bar{l} = e^{\lambda(\gamma+\tau)}$. The average busy period is made up of both successful and unsuccessful transmission periods, and can be expressed in terms of the percentage of successful and unsuccessful slots from a set of \bar{l} slots (see Figure 2.10). Therefore, the average busy period is

$$\begin{aligned} \bar{B} &= \bar{l}[T \cdot P_S + T_{FAIL} \cdot (1 - P_S)] \\ &= e^{\lambda(\gamma+\tau)} \left[\left(\frac{(\gamma + \tau)\lambda e^{-(\gamma+\tau)\lambda}}{(1 - e^{-\lambda(\gamma+\tau)})} \right) (\delta + 5(\gamma + \tau)) + \left(1 - \frac{(\gamma + \tau)\lambda e^{-(\gamma+\tau)\lambda}}{(1 - e^{-\lambda(\gamma+\tau)})} \right) (\gamma + \tau) \right] \end{aligned}$$

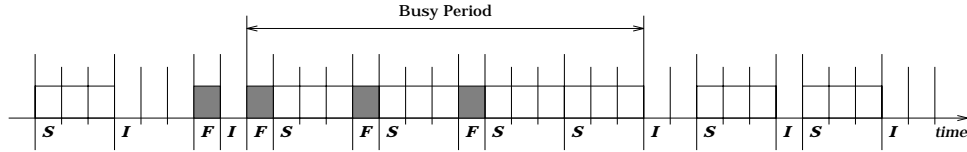


Figure 2.10: Slotted MACA transmission periods.

S = The Beginning of a successful transmission.

F = The beginning of a failed transmission period.

I = The beginning of an idle period.

Because \bar{l} is the average number of slots in a busy period, the average number of successful slots in a busy period is $\bar{l} \cdot P_S$. Therefore, given that each successful slot corresponds to the use of the channel for data traffic for δ seconds,

$$\begin{aligned} \bar{U} &= \bar{l} \cdot \delta \cdot P_S \\ &= \frac{\delta(\gamma + \tau)\lambda}{(1 - e^{-\lambda(\gamma+\tau)})} \end{aligned} \quad (2.27)$$

The idle period is determined similarly to the case of slotted ALOHA [RS90], and is based on the RTS instead of the data packet

$$\bar{I} = \frac{(\gamma + \tau)}{1 - e^{-(\gamma+\tau)\lambda}} \quad (2.28)$$

Substituting Eqs. (2.27), (2.27) and (2.28) into Eq. (2.1) we obtain Eq. (2.25). \square

2.3.6 Performance Comparison

Traditionally, throughput S is expressed in terms of the propagation delay and offered load normalized to data packet transmission time. To facilitate the comparison of the various protocols, we normalize the results obtained for S by making $\delta = 1$ and introducing the following variables

$$a = \frac{\tau}{\delta} \text{ (normalized propagation delay)}$$

$$b = \frac{\gamma}{\delta} \text{ (normalized control packets)}$$

$$G = \lambda \times \delta \text{ (Offered Load, normalized to data packets)}$$

Table 2.1 lists the normalized throughput equations for non-persistent CSMA and the two FAMA variants addressed in this chapter.

<i>Unslotted version</i>	<i>Slotted Version</i>
<i>CSMA</i>	
$\frac{Ge^{-aG}}{G(1+2a)+e^{-aG}}$	$\frac{aGe^{-aG}}{1+a-e^{-aG}}$
<i>MACA</i>	
$\frac{1}{e^{G(2b+a)}\left(b+a+\frac{1}{G}+F'\right)+e^{Gb}\left(b+\frac{a}{2}+P'\cdot(a-F')\right)+1+\frac{3a}{2}+F'+P'\cdot(a-F')}$ where $F' = \left[\frac{e^{Gb}-1-Gb}{Gb(1-e^{-Gb})}\right]; P' = \left[\frac{e^{-Gb}-e^{-G(b+a)}}{(1-e^{-G(b+a)})}\right]$	$\frac{1}{1+4(b+a)+\frac{e^{G(b+a)}}{G}}$
<i>FAMA-NTR</i>	
$\frac{1}{b+1+\frac{(2-e^{-aG})}{G}+e^{aG}(b+4a)}$	$\frac{Gae^{-Ga}}{Gae^{-Ga}(b+1+a)+(1-e^{-Ga})(b+3a)+a}$

Table 2.1: Throughput Equations for CSMA and FAMA protocols

We first compare the throughput of FAMA protocols with that of non-persistent CSMA, in both a low speed network (9600 b/s) and a high speed network (1 Mb/s), using both small data packets of 53 bytes (as in ATM cells) and longer packets of 296 bytes (as in a SLIP link). We assume a network with a maximum diameter of 10 miles, which gives us a propagation delay of approximately $54\mu\text{s}$. The minimum size of RTSs and CTSs is 20 bytes to accommodate the use of IP addresses for destination and source, a CRC, and framing bytes. Table 2.2 shows the values of a and b used to approximate the results for the comparison. Figures 2.11 and 2.12 show the throughput (S) versus the offered load (G) for non-persistent CSMA and the FAMA protocols under these conditions. Figure 2.13 assumes a high-speed network of 1 Mb/s and packet trains of 10 SLIP packets for different propagation delays. Figure 2.14 shows the impact of b on the throughput of FAMA-NTR compared to non-persistent CSMA. The exact values assumed in the network parameters are not as important as the relative differences in throughput among the various protocols.

Network	a	b
Low Speed, small packets	0.0012	0.375
Low Speed, SLIP packets	0.0002	0.067
High Speed, small packets	0.127	0.375
High Speed, SLIP packets	0.022	0.067

Table 2.2: FAMA protocol variables

Our results indicate the importance of using carrier sensing as an integral part of the floor acquisition strategy (see Figures 2.11 and 2.12). In the absence of hidden terminals FAMA-NTR provides a much higher throughput than MACA or slotted MACA, and its performance under high offered load in high-speed networks is even better than non-persistent CSMA. Of course, FAMA-NTR is more attractive for small values of $b = \gamma/\delta$, as

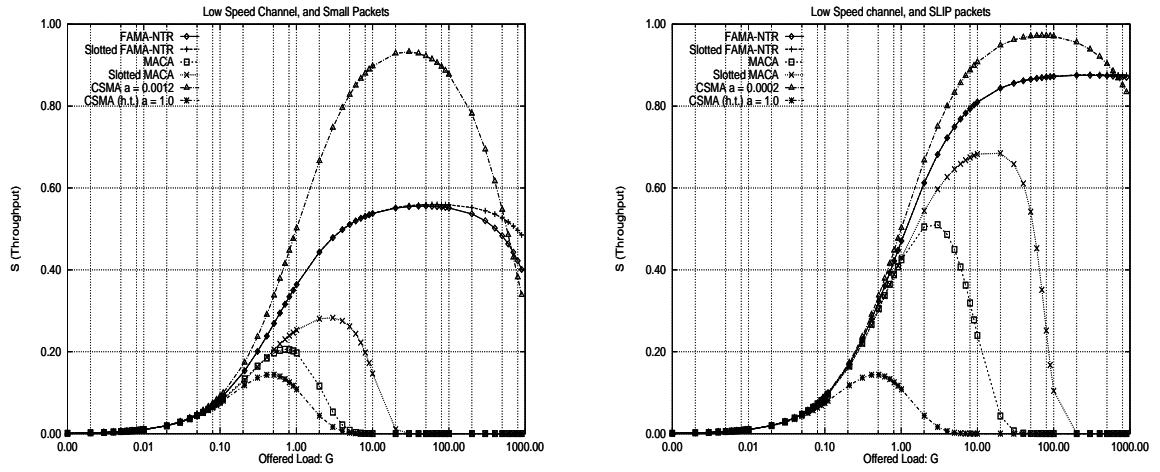


Figure 2.11: Throughput of FAMA protocols in a low-speed network.

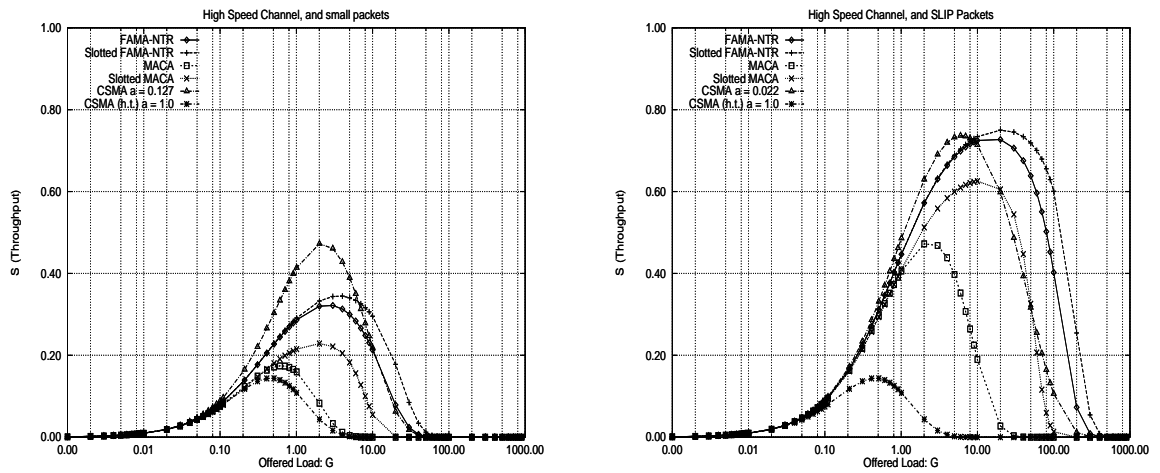


Figure 2.12: Throughput of FAMA protocols in a high-speed network.

shown in Figure 2.14. In practice, the effect of a small b can be obtained by allowing a station to transmit multiple packets per floor acquisition. Our results on MACA throughput are in agreement with the empirical simulation results presented by Bharghavan, et al. [BDSZ94] for single-hop networks. In FAMA-NTR, slotting adds little performance improvement over the basic protocol. This should be expected, as slotting adds little benefit in non-persistent CSMA [KT75]. An interesting result, however, is that even with slotting, MACA does not match FAMA-NTR's performance.

The effect of the hidden-terminal problem can be appreciated indirectly in our analysis for the two extreme cases in which either all or none of the senders and receivers are affected by it, which provides a lower and upper bound on the network's throughput, respectively. Figures 2.11 and 2.12 show two curves for non-persistent CSMA for each throughput comparison. One corresponds to the values of a listed in Table 2.2, and the other corresponds to $a = 1$. This throughput of non-persistent CSMA for $a = 1$ marked as CSMA (h.t.) is plotted to show the impact of hidden terminals. With hidden-terminals, the throughput of non-persistent CSMA becomes as low as the throughput of the ALOHA protocol (which has a maximum throughput of $\frac{1}{2}e \approx 0.18$), because stations are unable to sense the channel reliably and the vulnerability period of each packet is the whole packet. Under the same conditions, the throughput of a protocol similar to FAMA-NTR that can successfully prevent collisions of data packets with other packets can become no lower than the throughput of the MACA protocol, because even if the carrier sensing part of the floor acquisition strategy is unable to detect collisions due to the fact that stations cannot hear one another, the modified RTS-CTS exchange prevents data packets from colliding with any other packets. Hence, without hidden terminals FAMA-NTR achieves a throughput

comparable to CSMA's; with hidden terminals, a modified RTS-CTS exchange in FAMA-NTR can achieve throughput comparable to MACA under no hidden terminals. Of course, without some modifications, the throughput of both MACA and FAMA-NTR degrades with hidden terminals, because data packets can collide with RTSs from hidden terminals.

The importance of acquiring the floor (i.e., ensuring that data packets are sent without collisions) is also made clear by our results. In both low-speed and high-speed channels, it is clear that a larger throughput can be obtained with a larger ratio of δ/γ . Because transmitting very long data packets may not be appropriate in some applications using the network, allowing a station to send packet bursts in the clear after a successful RTS-CTS exchange becomes very attractive. Furthermore, once a station acquires the floor, it can send different packets to different receivers in support of multiple applications. Figure 2.13 further illustrates the importance of floor acquisition in the performance of the network for applications requiring either the transfer of large amounts of data (e.g., video transmissions) or the distribution of different information to different destinations. Again, a large ratio of δ to γ gives a high throughput in FAMA-NTR. For applications able to use larger packet trains, FAMA-NTR in a high-speed channel is even more effective than non-persistent CSMA.

It is also apparent that using MACA (or its derivatives) in low or high-speed channels to transfer single small packet (i.e., comparable to the size of an RTS) is not attractive at all. In such a case, the throughput of MACA is almost as low as what is expected in the ALOHA channel. That is, even though collisions are being detected at the receivers, the overhead incurred to do so is so large that the performance of the network is not much better than allowing a station to transmit its (small) packets whenever it is

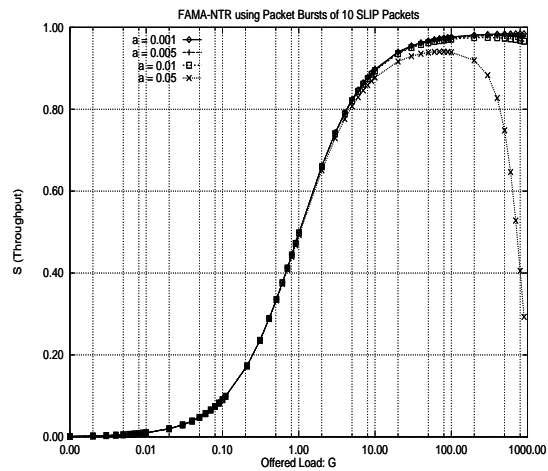


Figure 2.13: Throughput of FAMA-NTR protocols for different values of a using a 10 packet train in a high-speed network.

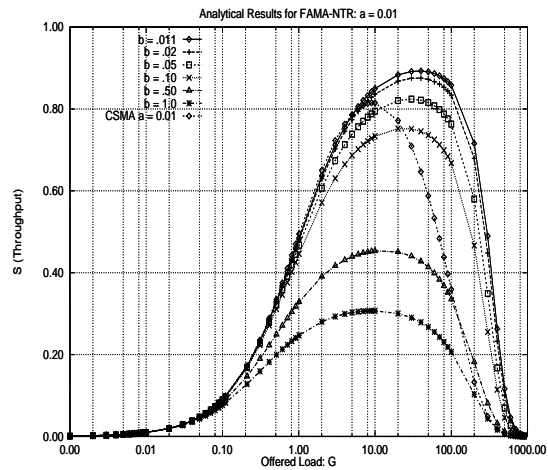


Figure 2.14: Throughput of FAMA-NTR versus traffic for different values of b in a high speed network.

ready.

In conclusion, combining carrier sensing with RTS-CTS exchanges to assign a random access channel dynamically, and allowing stations to transmit finite packet bursts once they acquire the floor, is the best approach.

2.4 Related Work

There are several prior proposals for single-channel MAC protocols similar to the FAMA protocols we have discussed. As we have stated, the protocol used in Apple's local talk link access protocol [SAO90] and the protocol in IEEE 802.11 [IEE97] use an RTS-CTS exchange with non-persistent carrier sensing. These protocols become FAMA-NTR when the duration of RTS packets is longer than the longest propagation time and a single data packet is sent after each successful RTS-CTS exchange.

Lo [LM84] and Rom [Rom86] have proposed protocols similar to non-persistent CSMA that detect collisions by means of pauses. A station that senses the channel busy defers transmission, a transmitter that senses the channel idle starts transmitting but pauses during transmission and senses the channel. If the channel is sensed idle, the sender completes its transmission; otherwise, the sender continues to transmit for a minimum transmission duration (called the collision detection interval or CDI). Unfortunately, this protocol does not guarantee that a station can sense all collisions [Rom86]. Furthermore, these protocols cannot tolerate hidden terminals.

Another CSMA-like protocol based on the idea of sending a request signal and pausing to sense collisions was proposed by Colvin [Col83] and analyzed by Brewster [BG87]. This protocol, however, was designed for LANs in which stations can sense the channel while transmitting.

A number of techniques have been proposed by Bharghavan, et al. [BDSZ94] to improve the performance of MACA [Kar90], which constitutes a variant of FAMA using RTS-CTS exchange and no carrier sensing. These techniques consist of different retrans-

mission strategies and additional handshaking between sender and receiver. The resulting protocol is called MACAW. Like MACA, MACAW is based on the basic premise that collisions are detected not by sensing the channel, but by the receivers being able to understand the transmissions they receive. Given that we have assumed the minimum RTS-CTS handshake of MACA and full connectivity in our analysis, our results on MACA provide an upper bound on MACAW's throughput. Chen [Che94] presents an approximate analysis of the modification of non-persistent CSMA that is part of IEEE 802.11 and based on a four-way RTS-CTS handshake. This analysis assumes that the RTS-CTS transmission cycle occupies a normalized time with respect to the duration of a data packet, and that CTSs are perfectly reliable and consume no overhead. This implies that the only source of losses for the protocol is the collision of RTSs. However, as our throughput analysis shows, collisions of CTSs with RTSs play an important role in the calculation of throughput.

2.5 Conclusions

We have introduced and analyzed a new type of channel access discipline for single-channel packet-radio networks, which we call floor acquisition multiple access (FAMA). FAMA protocols permit a station to acquire control of the channel dynamically before transmitting data packets. The floor acquisition strategy is based primarily on a request-response (RTS-CTS) control dialogue between a sender and an intended proxy receiver. In addition, carrier sensing is used to increase substantially channel throughput.

Although many MAC protocols have been introduced in the past based on RTS-CTS exchanges, our analysis shows, for the first time, sufficient conditions under which an

RTS-CTS dialogue becomes a floor acquisition strategy (i.e., one with which data packets are sent without ever colliding with other transmissions) with and without carrier sensing.

FAMA-NTR's throughput is that of non-persistent CSMA when all the senders wanting to transmit to a receiver can hear one another, and a modified version of FAMA-NTR that eliminates hidden terminals can have a throughput no lower than MACA's throughput with no hidden terminals. Slotting in the FAMA protocols analyzed provided substantial performance improvements only for the case of MACA (i.e., FAMA-NTR with hidden terminals). Our results clearly show that carrier sensing should be used together with the RTS-CTS handshake in MAC protocols for packet-radio networks.

Chapter 3

FAMA Emulation of Collision Detection

Wireless local area networks (WLANs) are playing an increasingly important role in the future of personal communications. Devices such as the personal data assistant (PDA), the personal information communicator (PIC), and personal computers with radio modems are widely available. These devices are generally mobile and operate intermittently due to the power considerations of mobile operation.

In many WLANs, the communication medium the devices use is a single radio channel. Our research investigates the development of a medium access control (MAC) protocol for WLANs in which all stations are within hearing distance of one another.

Our protocol practices collision avoidance using carrier detection in a manner similar to Carrier Sense Multiple Access (CSMA) [KT75]. CSMA protocols use the sensing of the channel prior to a transmission to avoid colliding with other transmissions already

in progress in the channel [KT75, Tob74]. The CSMA protocols can be further extended into two major subgroups: collision detection (CSMA/CD) [MB76] and collision avoidance (CSMA/CA) [Col83]. CSMA/CD is an efficient protocol and is used in Ethernet networks [MB76]. The transceiver listens during its own transmission and terminates the transmission upon detecting a collision with other signals in the channel. However, CSMA/CD cannot be applied directly to WLANs because the radios in these networks cannot listen to the channel during their own transmissions, making collision detection difficult. CSMA/CA is a variant of CSMA/CD. A sender to the channel transmits a short burst if the channel is clear, listens again, transmits another short burst with a short pause followed by the data packet. The destination station returns an ACK/NACK immediately after the end of the data packet for verification to the sending station.

In this chapter we present a new channel access protocol that we call floor acquisition multiple access with pauses and jamming (FAMA-PJ). The objective of FAMA-PJ is to allow a station that has data to send to acquire control of the channel (which we call the floor) before sending any data packets, and to ensure that no data packet can collide with any other packet. In contrast to prior collision avoidance MAC protocols that assign the channel dynamically (e.g., SRAM [TK75], BRAM [CFL79], MSAP [KS80]), FAMA-PJ does not use control subchannels or preambles. Section 3.1 describes the operation of the FAMA-PJ protocol. Section 3.2 shows that the FAMA-PJ protocol assigns the floor correctly, i.e., at most one station can send data packets at any given time. Sections 3.3 and 3.4 analyze the throughput and delay of the protocol and compare it against non-persistent CSMA. Section 3.5 presents our conclusions.

3.1 FAMA-PJ

FAMA-PJ requires a station who wishes to send one or more packets to acquire the floor before transmitting the packet train. The floor is acquired using control packets that are multiplexed together with the data packets in the same channel in such a way that, although control packets may collide with other control packets, data packets are always sent free of collisions. There are different schemes with which the channel floor can be acquired. A number of protocols, including MACA and IEEE 802.11 [Col83, Kar90, BDSZ94, SAO90] use a three or four-way handshake between sender and receiver to avoid collisions. The sender first sends a request-to-send (RTS) packet to the intended receiver; if the RTS is successful the receiver replies with a clear-to-send (CTS) packet, and the sender transmits its data packet only after receiving the CTS. In some protocols, the receiver sends an acknowledgment as part of the MAC protocol after receiving the data packet. This three or four-way handshake amounts to collision detection by the receiver and addresses, at least to some degree, two main problems: (a) the half-duplex nature of single-channel wireless networks, and (b) the hidden-terminal effect [TK75, BDSZ94], which inhibits the transmitter's ability to detect that its intended receiver is receiving multiple transmissions.

In a LAN with a small propagation delay and in which hidden-terminal problems do not occur, collision detection by the transmitter can substantially increase the utilization of the channel; this is the case of CSMA/CD used in Ethernet [MB76]. However, CSMA/CD requires the sender to sense the channel while it transmits; if the carrier signal has an energy level different than what is expected from the senders transmission, collision is detected, which can happen within one round-trip propagation delay. Unfortunately, a WLAN with a

single channel has a half-duplex operation. To cope with this limitation, Lo [LM84] and Rom [Rom86] have proposed protocols similar to non-persistent CSMA that detect collisions by means of pauses. A station that senses the channel busy defers transmission, a transmitter that senses the channel idle starts transmitting but pauses during transmission and senses the channel. If the channel is sensed idle, the sender completes its transmission; otherwise, the sender continues to transmit for a minimum transmission duration (called the collision detection interval or CDI). Unfortunately, these protocols do not guarantee that a station can sense all collisions [Rom86].

Another CSMA/CA protocol based on the idea of sending a request signal and pausing to sense collisions has been proposed by Colvin [Col83] and analyzed by [BG87]. This protocol, however, was designed for LANs in which stations can sense the channel while transmitting.

FAMA-PJ differs from the Lo and Rom protocols in that it enforces floor acquisition and uses jamming from both active and passive stations (i.e., stations involved in sending packets or simply listening).

A station that is just initialized waits two times the maximum channel propagation delay plus the hardware transmit to receive transition time before sending anything over the channel. This permits the station to learn about ongoing packet trains if they exist. A station that is properly initialized (has no packet to send and senses an idle channel) must be in the PASSIVE state. In all states but the PASSIVE state, before transmitting anything a station must listen to the channel for a period of time that is sufficient for the station to receive packets in transit from the station that has the floor. If a station is in the PASSIVE state and detects carrier, it transitions to the REMOTE state; alternatively,

```

Variable Definitions
CD = Carrier Detected
 $T_{PROP}$  = Propagation Delay across the channel
 $T_{RTS}$  = Time required to transmit an RTS packet
 $T_{PROC}$  = Processing time for carrier detection
 $T_{XRX}$  = Hardware transmit to receive transition time
 $T_{FAMA} = 2T_{PROP} + T_{PROC} + T_{XRX}$ 
Burst = Number of packets to send in a burst

Procedure START()
Begin
  Timer  $\leftarrow$  to  $(2 \times T_{PROP}) + T_{XRX}$ ;
  While( $\overline{CD} \wedge$  Timer not expired) wait;
  If (CD) Then call REMOTE( $T_{FAMA},0$ );
  Else call PASSIVE()
End

Procedure PASSIVE()
Begin
  While( $\overline{CD} \wedge$  No Local Packet) wait;
  If (CD) Then call REMOTE( $T_{FAMA},1$ );
  Else call RTS( $T_{PROP}$ )
End

Procedure RTS( $T_{\sigma}$ )
Begin
  Transmit RTS Packet;
  Timer  $\leftarrow T_{\sigma} + T_{XRX}$ ;
  While( $\overline{CD} \wedge$  Timer not expired) wait;
  If (CD) Then call BACKOFF();
  Else call XMIT();
End

Procedure XMIT()
Begin
  Burst  $\leftarrow$  maximum burst;
  While (Burst > 0)
  Do Begin
    Transmit Data Packet;
    Burst  $\leftarrow$  Burst - 1;
  End;
  Timer  $\leftarrow T_{PROP} + T_{XRX}$ ;
  While(Timer not expired) wait;
  If (Local Packet) Then call BACKOFF();
  Else call PASSIVE();
End

Procedure BACKOFF()
Begin
  If (CD) call REMOTE( $T_{FAMA},0$ );
  Else Begin
    Timer  $\leftarrow$  RANDOM(0,10  $\times T_{RTS}$ );
    While( $\overline{CD} \wedge$  Timer not expired) wait;
    If (CD) Then call REMOTE( $T_{FAMA},1$ );
    Else call RTS( $T_{PROP}$ )
  End
End

Procedure REMOTE( $T_{\sigma},flag$ )
Begin
  Timer  $\leftarrow T_{\sigma}$ ;
  While( $\overline{CD} \wedge$  Timer not expired) wait;
  If (Timer Expired)
  Then Begin
    If (Local Packet) Then call BACKOFF();
    Else call PASSIVE();
  End
  Else Begin
    If ( $flag = 1$ )
    Then Begin;
    Timer  $\leftarrow T_{RTS}$ ;
    While(Timer not expired) wait;
    Receive Packet;
    DO CASE of (received packet type)
    Begin
      RTS:
        call REMOTE( $T_{FAMA},0$ );
      ERROR:
        Transmit jamming signal for ( $T_{XRX} + 2T_{PROP}$ );
        call REMOTE( $T_{FAMA},0$ );
    End
  End
  Else Begin
    While (CD) wait;
    Receive Packet;
    DO CASE of (received packet type)
    Begin
      RTS:
        call REMOTE( $T_{FAMA},0$ );
      DATA:
        If(Destination ID = Local ID)
          Then pass packet to upper layer;
        call REMOTE( $T_{PROP} + T_{PROC},0$ );
      ERROR:
        call REMOTE( $T_{FAMA},0$ );
    End
  End
End

```

Figure 3.1: FAMA-PJ Specification

if the station receives a packet to send, it sends an RTS and transitions to the RTS state. Note that, although the station sends its RTS after receiving a local data packet, this can occur only after the station has waited the necessary amount of time in another state to learn that the last station having the floor has relinquished the floor. A station that has a packet to send and senses no carrier in the channel for an amount of time longer than the propagation time in the channel plus the transmit to receive turn-around time or the maximum gap allowed between data packets in a packet train transmits an RTS whose

duration is longer than twice the maximum propagation time in the channel; the station then pauses to sense the channel. If the station senses the channel to be idle for τ seconds (after the station begins sensing it), the station concludes that its RTS was successful and transmits one or more data packets; otherwise, that station jams the channel for at least one maximum propagation delay.

Jamming of the channel by stations that fail in sending a successful RTS is called *active jamming*; this type of jamming has been proposed in all previous approaches based on pausing and jamming. Active jamming is sufficient to inform all stations that a collision of control packets has occurred if the channel propagation time is longer than the transceiver turn around time ε (the time a packet radio takes to transition from the transmit to listen state). However, as Fig. 3.2 illustrates, if $\varepsilon \geq \tau$, active jamming is not sufficient to detect collisions. In the example shown in Fig. 3.2, station *A* transmits an RTS first, and station *B* transmits an RTS within τ seconds of *A*'s RTS start. Station *B* cannot detect collision, because it transitions to the listen state when there is no more carrier produced by *A*. Station *A* cannot detect collision, because the carrier produced by *B* is too weak to be detected by the time *A* is able to listen to the channel.

To solve the limitations of active jamming, FAMA-PJ uses “passive jamming.” Passive jamming can be performed only by stations in the REMOTE state, which prevents stations from jamming one another for indefinite periods of time. When a passive or backed-off station detects carrier and transitions to the REMOTE state it waits for γ seconds. After this time, if it has understood an RTS, it waits another $\varepsilon + \tau$ seconds to begin receiving one or more data packet, or to let another station receive such packets. Otherwise, if no RTS is understood after γ seconds, the station ascertains that there has been a collision in

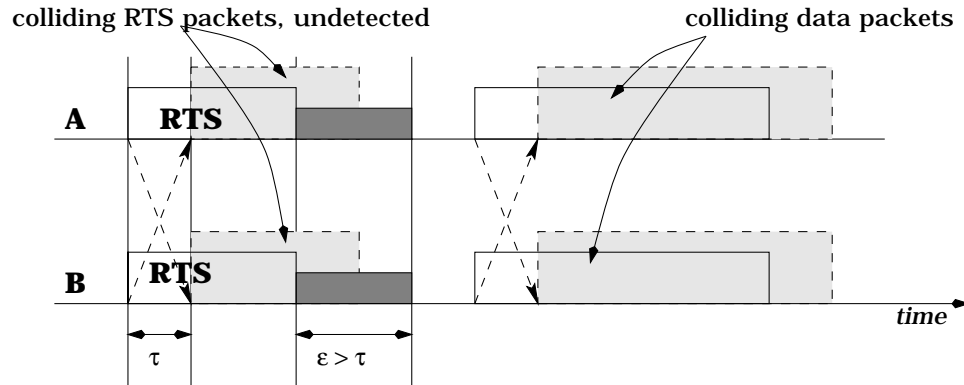


Figure 3.2: FAMA-PJ active jamming failure under conditions where $\epsilon > \tau$

the channel and the station begins “passive jamming” by transmitting a jamming signal for $\epsilon + 2\tau$ seconds. FAMA-PJ includes active jamming to handle 2-node cases. In a WLAN, the maximum propagation times are on the order of a few microseconds, while ϵ can be as large as $35\mu\text{s}$ [IEE97]; therefore, we make the assumption that $\epsilon \geq \tau$ throughout the rest of this chapter. Figure 3.1 specifies FAMA-PJ.

The backoff times used in FAMA-PJ are obtained using a uniformly distributed random variable distributed over the values from zero to ten times the duration of an RTS transmission. Other backoff strategies can also be used (e.g., see [BDSZ94]).

3.2 Floor Assignment in FAMA-PJ

The size of the RTS packets in relation to the data packets is critical to the efficient operation of the protocol. If the size of an RTS packet approaches the size of the data packets, the overhead of the contention period will degrade the performance considerably. Therefore, RTS packets must be kept small as compared to the data packets. RTS packets must also be larger than two times the maximum propagation time (T_{PROP}) across the

network. If the RTS packet size is less than two times the propagation time, it is possible for a passive station to hear a clear RTS before others in the network, not jam as required, and allow a data packet to be transmitted that could collide with other traffic on the channel, violating our requirement of collision free data transmissions.

Theorem 7 below shows that, under a number of assumptions, FAMA-PJ ensures that all data packets accepted by the link layer are delivered to the channel within some finite period of time, and that all data packets delivered to the channel reach their proper destination without collisions. The assumptions used are the following:

- A1) A station transmits an RTS that does not collide with other transmission with probability larger than 0.
- A2) All stations can hear one another and are within one maximum propagation delay (τ) of all other stations.
- A3) All stations execute the FAMA-PJ protocol correctly.
- A4) The transmission time of an RTS packet is γ , the transmission time of a data packet is δ , and the processing time is $t_p < \infty$. The hardware transmit to receive transition time is ε , where $\varepsilon \geq \tau$ and $2\tau < \gamma \leq \delta < \infty$.
- A5) There are three or more stations in the network
- A6) The probability that all stations in the network transmit an RTS simultaneously is 0
- A7) The maximum end-to-end propagation time in the channel is $\tau < \infty$

Theorem 7 *FAMA-PJ ensures that each new packet, or any of its retransmissions, is sent*

to the channel within a finite time after it becomes ready for transmission, and that a data packet does not collide with any other transmission provided that $2\tau < \gamma < \infty$.

Proof:

By this theorem's assumptions, an RTS lasts longer than two times the channel propagation time. Therefore, if an arbitrary station A is able to send an RTS that does not collide with other transmissions, all other stations must detect carrier before A ends transmitting its RTS and must enter the REMOTE state, which forces them to enter a waiting period that lasts longer than $\varepsilon + \tau$ seconds after detecting the end of A 's RTS transmission.

Let t_0 be the time when A sends its RTS; station A will be able to send data packets if it senses the channel idle for τ seconds once it transitions to listening mode after sending its RTS; which occurs at time $t_1 = t_0 + \gamma + \varepsilon + \tau$.

If a station $B \neq A$ starts to receive A 's RTS at time t_2 , it must wait for $\varepsilon + 2\tau$ seconds after receiving A 's entire RTS before it can be allowed to send any traffic to the channel. B must receive A 's data packet at most τ seconds after A sends it, i.e., at time $t_3 = t_1 + \tau = t_0 + \gamma + \varepsilon + 2\tau$.

Let t_4 denote the time when B is allowed to transmit after not receiving any carrier due to a data packet from A , following A 's RTS; we have that $t_4 = t_2 + \gamma + \varepsilon + 2\tau$. Because B cannot receive A 's RTS in 0 time $t_3 > t_0$; therefore, $t_4 > t_0 + \gamma + \varepsilon + 2\tau = t_3$ and B cannot transmit anything to the channel.

This means that once station A transmits an RTS in the clear, it can send a data packet to the channel in the clear. Furthermore, every station in REMOTE state must wait

$\varepsilon + \tau$ seconds after the end of any packet received and the station controlling the floor ensures that the gap between any two data packets in a packet train is less than τ . Accordingly, it follows from the above and assumption A1 that, if a station has a data packet to be sent, it delivers the packet to the channel without collisions within a finite time.

Let t_0 be the time when a given passive station PJ receives the first RTS of a series of RTSs that collide in the channel and denote that packet by RTS_0 . Any other colliding RTS must be transmitted by a station no later than $t_0 + \tau$; otherwise the station would detect the carrier of RTS_0 ; furthermore, that RTS must arrive at PJ within τ seconds after it is transmitted. Accordingly, because all propagation delays are positive, it follows that PJ must receive any RTS that collides with RTS_0 no later than $t_0 + 2\tau$.

Because $\gamma > 2\tau$, PJ must detect a collision and jam the channel at time $t_0 + \gamma$, and that jamming persists for $\varepsilon + 2\tau$ seconds. Any station that sends an RTS that collides with RTS_0 must return to the listening mode by time $t_1 = t_0 + \tau + \gamma + \varepsilon$; therefore, given that $t_2 = t_0 + \gamma + \varepsilon + 2\tau > t_1$ is the time when PJ stops jamming the channel, any station that sends an RTS that collides with RTS_0 must detect carrier with PJ 's jamming and go the BACKOFF state. It then follows that no station whose RTS collides with other RTSs can send a data packet; therefore, the theorem is true. \square .

3.3 Approximate Throughput Analysis

To simplify our analysis, we assume the same traffic model first introduced by Kleinrock and Tobagi [KT75]. The protocols we analyze are non-persistent CSMA, FAMA-PJ, and their slotted counterparts.

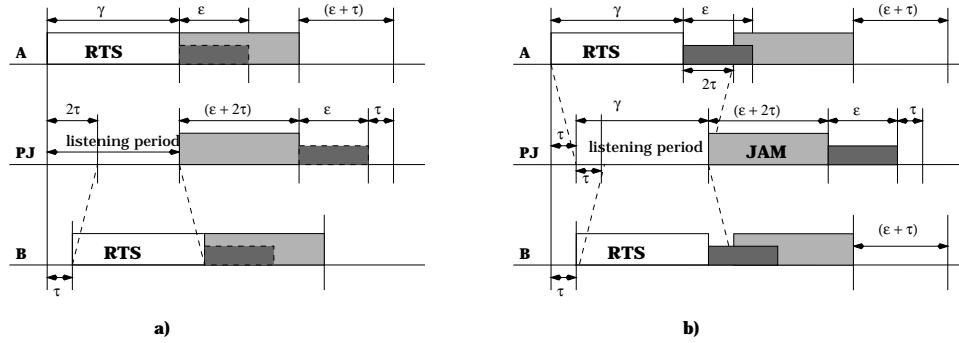


Figure 3.3: FAMA-PJ passive jamming periods:
a) Stations *A* and *PJ* are next to each other.
b) Stations *A* and *PJ* are τ seconds apart.

3.3.1 Assumptions and Notations

Using the traffic model introduced by Tobagi and Kleinrock [KT75], there is an infinite number of stations who constitute a Poisson source sending RTS packets (for the case of FAMA-PJ) or data packets (for the case of CSMA) to the the channel with an aggregate mean generation rate of λ packets.

Each station is assumed to have at most one data block to be sent at any time. In both protocols a station transmits the entire data block as a single packet (which is the case of CSMA) or as multiple packets (which is the case of FAMA-PJ). The hardware is assumed to require a fixed turn-around time of ε seconds to transition from transmit to receive, for any given transmission to the channel. The average size of a data block is δ seconds. RTS packets are of size γ seconds, and the maximum end-to-end propagation delay of the channel is τ seconds. Collisions (e.g., RTS packets in FAMA-PJ, data packets in CSMA) can occur in the channel, and we assume that, when a station has to retransmit a packet it does so after a random retransmission delay that, on the average, is much larger

than δ . The average channel utilization is given by [Tob74]

$$S = \frac{\bar{U}}{\bar{B} + \bar{I}}. \quad (3.1)$$

where \bar{B} is the expected duration of a busy period, defined to be a period of time during which the channel is being utilized; \bar{I} is the expected duration of an idle period, defined as the time interval between two consecutive busy periods; and \bar{U} is the time during a busy period that the channel is used for transmitting user data successfully.

The channel is assumed to introduce no errors, so packet collisions are the only source of errors, and stations detect such collisions perfectly. To further simplify the problem, we assume that every station can listen to the transmissions of any other station, that two or more transmissions that overlap in time in the channel must all be retransmitted, and that a packet propagates to all stations in exactly τ seconds [Tob74]. The later assumption provides a lower bound on the performance of the protocols we analyze.

The turn around time ε for a station to transition from transmit to receive or receive to transmit is assumed to be greater than or equal to the propagation delay τ .

Of course, this model is only a rough approximation of the real case, in which a finite number of stations access the same channel, some stations may not be able to hear some other stations' transmissions, stations can queue multiple packets for transmission, and the stations' transmissions and retransmissions (of RTS or data packets) are highly correlated because of the relationships between them (i.e., a failed RTS is followed by another RTS within a bounded time, and a singular data packet or packet train is always preceded by a successful RTS). However, our analysis provides additional insight on the performance of MAC protocols for WLANs based on collision detection at the sender as

well as passive receivers, which has not been addressed in recent protocol proposals, and the impact of channel speed and propagation delay on the floor acquisition technique. Our analysis favors CSMA in that we assume that the applications sending data to the channel can efficiently use data packets much larger than the size of an RTS.

3.3.2 FAMA-PJ

Figure 3.4 shows the transmission periods for the FAMA-PJ protocol. In FAMA-PJ, a station transmits an RTS packet and then listens for τ seconds. If the channel remains clear during this period, the station transmits the data packet. Otherwise, it transmits a jamming signal of τ seconds in length. Additionally, all passive stations (all stations either in the PASSIVE or BACKOFF state) listen to the signal, and after γ seconds from carrier detection they make a determination the the RTS is clear, or there has been a collision. If the RTS is clear, the RTS is processed normally and the station waits for the data impending data packet to arrive. If a collision has been detected the station begins transmission of a jamming signal $\varepsilon + 2\tau$ seconds in length. After any transmission period, the FAMA-PJ specification enforces a τ second long waiting time before stations transition to the PASSIVE or BACKOFF state.

Theorem 8 *The throughput of FAMA-PJ is given by*

$$S = \frac{\delta}{\delta - 2\tau + e^{\lambda\tau}(\gamma + 5\tau + 2\varepsilon + \frac{1}{\lambda})} \quad (3.2)$$

Proof:

A successful transmission period (T) consists of an RTS packet followed by the hardware transmit-to-receive transition (ε seconds), a listening period of τ seconds, and a

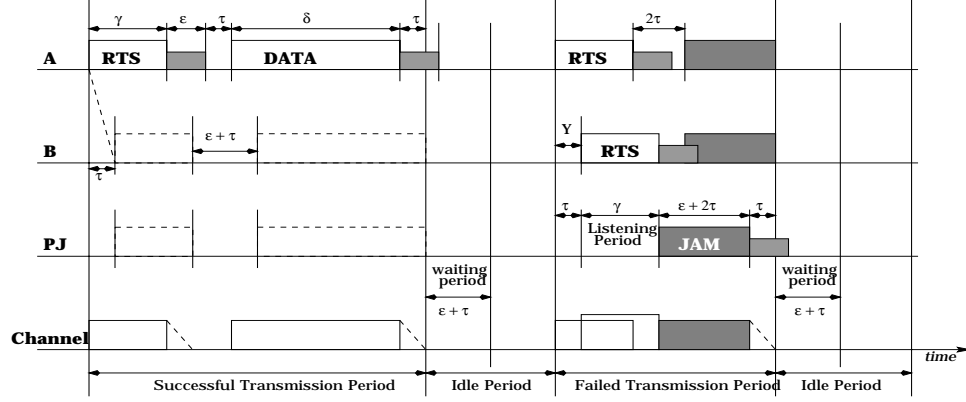


Figure 3.4: FAMA-PJ transmission periods

data packet followed by the τ second propagation delay across the channel; therefore,

$$T = \delta + \gamma + 2\tau + \varepsilon \quad (3.3)$$

An unsuccessful transmission period (T_{FAIL}) consists of an RTS payload (γ seconds in length), a τ second propagation delay to the passive jammer, a jamming signal of $\varepsilon + 2\tau$ seconds, and a final propagation delay of τ seconds across the channel. Therefore, the duration of the average failed transmission period is

$$T_{FAIL} = \gamma + 4\tau + \varepsilon \quad (3.4)$$

The probability P_S of a successful transmission is the probability that no other packet arrives during a propagation delay, i.e., $e^{-\lambda\tau}$.

The FAMA-PJ specification enforces a waiting period of $\tau + \varepsilon$ seconds after any given busy period, which is in turn followed by an idle period. Because of this, a busy period is made up of either one successful or unsuccessful transmission period; therefore,

$$\begin{aligned} \bar{B} &= P_S T + (1 - P_S) T_{FAIL} \\ &= e^{-\lambda\tau} (\delta - 2\tau) + \gamma + 4\tau + \varepsilon \end{aligned}$$

\bar{U} equals $\delta P_S = \delta e^{-\lambda\tau}$, and \bar{I} is simply the average interarrival time of RTSs into the channel, plus the enforced waiting time of $\tau + \varepsilon$ seconds, i.e., $(1/\lambda) + \tau + \varepsilon$.

Substituting $\bar{U}, \bar{B}, \bar{I}$ into Eq. (3.1) we obtain Eq. (3.2). \square

3.3.3 Slotted FAMA-PJ

We consider Slotted FAMA-PJ with the assumptions that the slot size equals the propagation delay τ , the length of the hardware transmit to receive transition time ε and the duration of an RTS and DATA packets are exact multiples of τ . Stations may only start transmissions at slot boundaries. Figure 3.5 shows the transmission periods possible for Slotted FAMA-PJ.

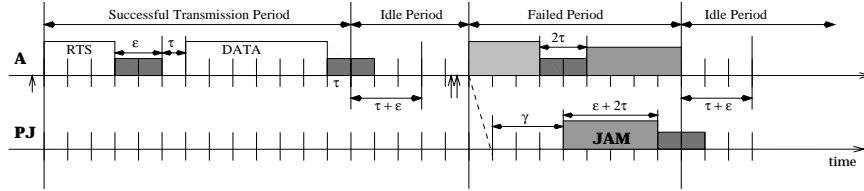


Figure 3.5: Slotted FAMA-PJ transmission periods

Theorem 9 *The throughput of Slotted FAMA-PJ is given by*

$$S = \frac{\delta}{\delta - 2\tau + \left[\frac{\gamma + 6\tau + 2\varepsilon - e^{-\lambda\tau}(\gamma + 5\tau + 2\varepsilon)}{\lambda\tau e^{-\lambda\tau}} \right]} \quad (3.5)$$

A successful transmission period for slotted FAMA-PJ consists of one RTS packet followed by the hardware transmit to receive transition time (ε seconds in length) along with one slot (τ seconds in length) to listen for other transmissions, and one DATA packet followed by one slot for final propagation delay. This is the same as in unslotted FAMA-PJ, and T is given in Eq. (3.3).

An unsuccessful period consists of one propagation slot for the colliding packets to reach the passive jammer (PJ), the length of an RTS packet (γ) for PJ to determine the collision has taken place, a jamming signal of $\varepsilon + 2\tau$ seconds duration, with one slot for final propagation. The failed period is therefore,

$$T_{FAIL} = \gamma + 4\tau + \varepsilon \quad (3.6)$$

A busy period consists of either one successful, or one unsuccessful transmission period. This is because a waiting period is enforced by the FAMA-PJ specification after any transmission period, followed by an idle period. The probability P_S of a successful transmission is the probability of having only one arrival in a busy slot, i.e.,

$$P_S = \frac{\lambda\tau e^{-\lambda\tau}}{(1 - e^{-\lambda\tau})} \quad (3.7)$$

The average length of a busy period (\overline{B}) is then

$$\begin{aligned} \overline{B} &= P_S T + (1 - P_S) T_{FAIL} \\ &= \left[\frac{\lambda\tau e^{-\lambda\tau}}{(1 - e^{-\lambda\tau})} \right] (\delta - 2\tau) + \gamma + 4\tau + \varepsilon \end{aligned} \quad (3.8)$$

The utilization of the channel, \overline{U} , is the data portion of a successful transmission period. Therefore, with probability P_S a transmission is successful and the data portion of such a period is δ , we obtain

$$\overline{U} = \delta P_S = \delta \left[\frac{\lambda\tau e^{-\lambda\tau}}{(1 - e^{-\lambda\tau})} \right] \quad (3.9)$$

The idle period consists of consecutive empty slots preceded by an enforced waiting period of $\tau + \varepsilon$ seconds in length, as defined by the FAMA-PJ specification. The number of consecutive idle slots has a geometric distribution whose mean is the same as that derived

for slotted non-persistent CSMA [KT75, RS90] and is equal to $1/(1 - e^{-\lambda\tau})$. The average idle period is then equal to the average number of consecutive empty slots plus the $\tau + \varepsilon$ second enforced waiting period. Therefore,

$$\bar{T} = \frac{\tau}{(1 - e^{-\lambda\tau})} + \tau + \varepsilon \quad (3.10)$$

Substituting \bar{U} , \bar{B} and \bar{T} into Eq.(3.1) we obtain Eq.(3.5). \square .

3.3.4 Performance Comparison

To facilitate the comparison of the various protocols, we normalize the results obtained for S by making $\delta = 1$ and introducing the following variables

$$\begin{aligned} a &= \frac{\tau}{\delta} \text{ (normalized propagation delay)} \\ b &= \frac{\gamma}{\delta} \text{ (normalized control packets)} \\ c &= \frac{\varepsilon}{\delta} \text{ (normalized transmit to receive turn around time)} \\ G &= \lambda \times \delta \text{ (Offered Load, normalized to data packets)} \end{aligned}$$

Table 3.1 lists the normalized throughput equations for non-persistent CSMA and FAMA-PJ. For the case of non-persistent CSMA, we assume the existence of an additional perfect channel for the transmission of acknowledgments to data packets; therefore, the throughput shown for CSMA is an upper bound. For the case of FAMA-PJ, because we assume that errors are due only to packet collisions and data packets are always transmitted in the clear, there is no need for acknowledgments to data packets under the assumption of no channel errors. However, because propagation delays and acknowledgments are much smaller than data packets, the effect of acknowledgments would have a negligible effect on FAMA-PJ throughput.

Protocol	<i>Unslotted version</i>	<i>Slotted Version</i>
CSMA [KT75]	$\frac{Ge^{-aG}}{G(1+2a)+e^{-aG}}$	$\frac{aGe^{-aG}}{1+a-e^{-aG}}$
FAMA-PJ	$\frac{1}{1-2a+e^{aG}(b+5a+2c+\frac{1}{G})}$	$\frac{1}{1-2a+\left[\frac{b+6a+2c-e^{-aG}(b+5a+2c)}{aGe^{-aG}}\right]}$

Table 3.1: Throughput Equations for CSMA and FAMA-PJ protocols

We have compared the operation of the FAMA-PJ protocol with non-persistent CSMA under a variety of possible scenarios. We compare operation in a high speed network (1 Mb/s), and using data packets of 500, 1000 and 1500 bytes (as might be seen in normal Ethernet traffic). We assume a network with a maximum diameter of 1000 feet, which gives us a propagation delay of approximately $1\mu\text{s}$. The RTS packets are 20 bytes long to accommodate the use of IP addresses for destination and source information, a CRC, and framing bytes. The transmit to receive turnaround time is assumed to be $20\mu\text{s}$, similar to the recommendations of the IEEE 802.11 standard [IEE97]. Table 3.2 shows the values of a , b and c used to approximate the results for the comparison. Figure 3.6 shows the throughput (S) versus the offered load (G) for the FAMA protocols under these conditions.

1Mbit/s Network	a	b	c
500 Byte data packets	0.00025	0.040	0.0050
1000 Byte data packets	0.000125	0.020	0.0025
1500 Byte data packets	0.000083	0.0133	0.00167

Table 3.2: FAMA-PJ protocol variables

Our results show the viability of using a collision detection mechanism at the sender

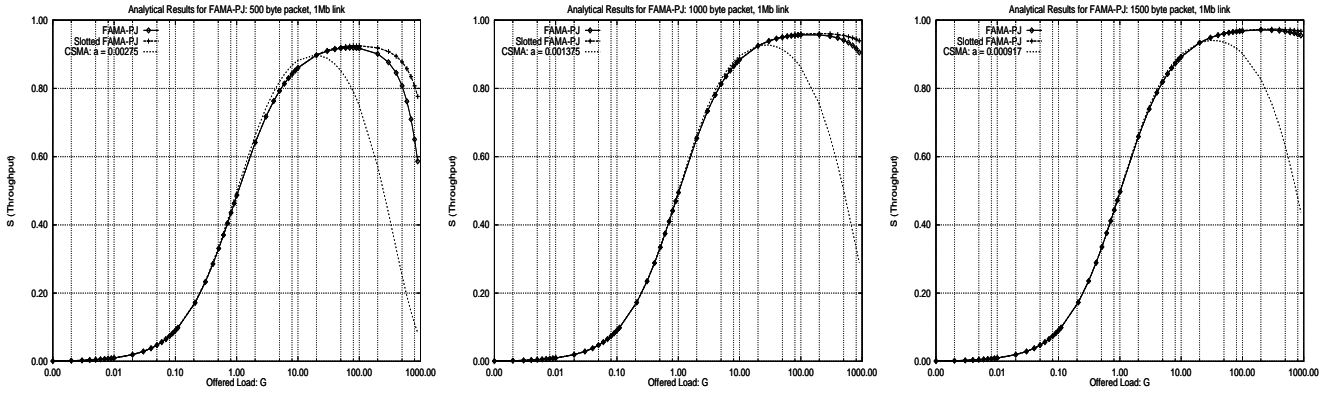


Figure 3.6: Analytical performance comparison of the FAMA-PJ protocol in a high speed network.

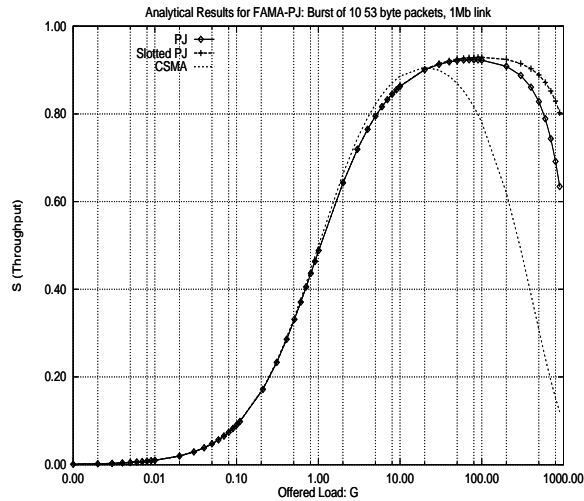


Figure 3.7: Throughput of FAMA-PJ protocols using a 10 packet train in a high-speed network.

and passive receivers along with carrier sensing as a floor acquisition scheme. Slotting adds little performance improvement over the basic FAMA-PJ protocol.

In high-speed channels, it is clear that a larger throughput can be obtained with a larger ratio of δ/γ . Because transmitting very long data packets may not be appropriate in some applications using the network, floor acquisition, i.e., allowing a station to send packet bursts in the clear after a successful RTS, becomes very attractive. Figure 3.7

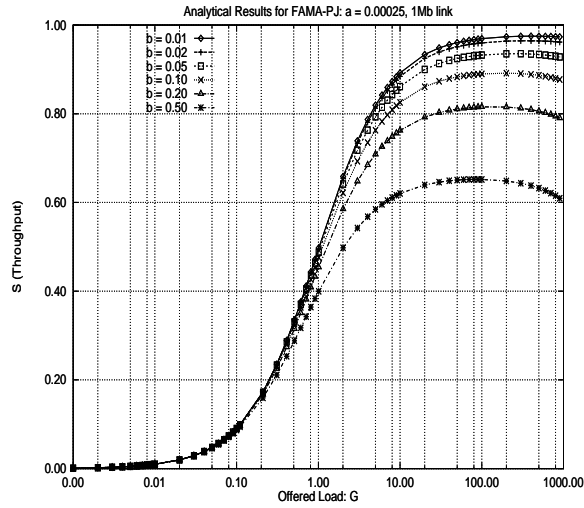


Figure 3.8: Throughput of FAMA-PJ versus b in a high speed network.

further illustrates the importance of floor acquisition in the performance of the network for applications requiring either the transfer of large amounts of data (e.g., video transmissions) or the distribution of different information to different destinations. Again, a large ratio of δ to γ gives a high throughput in FAMA-PJ. For applications able to use larger packet trains, FAMA-PJ in a high-speed channel is more effective than non-persistent CSMA.

3.4 Average Delay

To determine the average delay, we consider the transitions a station makes upon receiving a data packet, the probability of such transitions, and the related average delays incurred until the data packet is successfully delivered. Given the memoryless properties of the interarrival times of packets in the channel, we model the average delay experienced by a data packet as a Markov process; where a state of this process corresponds to the state in which a station with a packet to deliver would be in. Our model consists of

four states, ARRIVE, BACKOFF, ATTEMPT and COMPLETE, corresponding to the FAMA-PJ states of PASSIVE, BACKOFF, RTS and XMIT, respectively (see the FAMA-PJ specification, Figure 3.1). Figure 3.9 shows the individual states and their respective possible transitions along with the probabilities and delays associated with each of them.

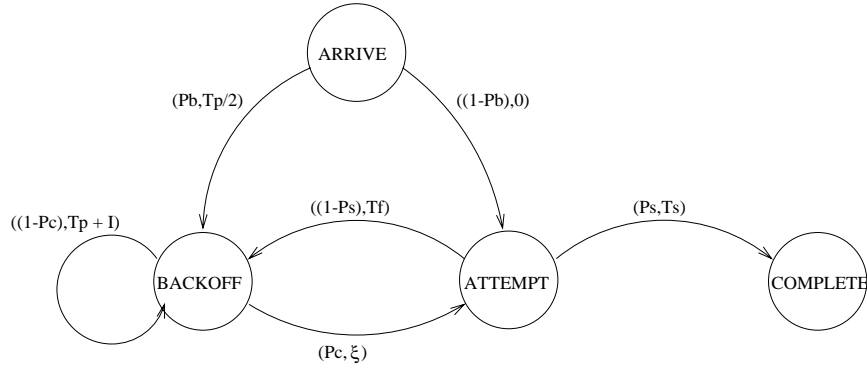


Figure 3.9: Markov chain defining FAMA-PJ delay characteristics.

The ARRIVE state is the entry point to the system for a station receiving a new packet to deliver. On the arrival of the data packet, the station finds the channel busy with probability P_B and incur a partial transmission period. Because the arrival process is Poisson, arrival times during any given time interval are independent and uniformly distributed [Tri88]. Therefore, on the average, the partial transmission period the station experiences is $T_P/2$, where T_P is the duration of an average transmission period. The station will then transition to the BACKOFF state.

Alternatively, with probability $(1 - P_B)$, the channel is clear when the station receives a packet to deliver, and the station incurs no delay and transitions to the ATTEMPT state. In the ATTEMPT state a station attempts to gain the floor. With probability P_S the station is successful and the data packet is delivered successfully, incurring a delay of

one successful transmission period, denoted by T_S . With probability $(1 - P_S)$ the station fails to acquire the floor, incurs a delay of a failed transmission period, denoted by T_F , and transitions to the BACKOFF state.

The BACKOFF state represents the stations random waiting periods before attempting to acquire the channel again. The average waiting period is ξ seconds. With probability P_C a station completes its waiting period and transitions to the ATTEMPT state and incurs a ξ -second delay. Otherwise, with probability $(1 - P_C)$, some other arrival to the channel occurs first and causes the station to be delayed one average transmission period, plus the average idle time, and be returned to the BACKOFF state.

The COMPLETE state represents the completed successful delivery of the data packet by the station, and ends the process.

To calculate the average delay we solve the system of equations implied by the graph in Figure 3.9. Let \bar{D} equal the expected delay incurred by a station with a new packet received at the ARRIVED state until it is successfully delivered at the COMPLETE state. Let $E(A)$ equal the expected delay incurred on each visit by the station to the ATTEMPT state, and let $E(B)$ equal the expected delay incurred on each visit to the BACKOFF state. From the graph in Figure 3.9 we obtain

$$\bar{D} = P_B \left[\frac{T_P}{2} + E(B) \right] + (1 - P_B) [0 + E(A)] \quad (3.11)$$

$$E(B) = P_C [\xi + E(A)] + (1 - P_C) [T_P + \bar{I} + E(B)] \quad (3.12)$$

$$E(A) = P_S \cdot T_S + (1 - P_S) [T_F + E(B)] \quad (3.13)$$

Solving for $E(B)$ we obtain

$$E(B) = \frac{T_P + \bar{I}}{P_C P_S} + \frac{\xi + T_F - T_P - \bar{I}}{P_S} + T_S - T_F \quad (3.14)$$

Substituting Eqs. (3.13) and (3.14) into Eq. (3.12) we obtain the formula for the expected delay

$$\begin{aligned} \bar{D} = & T_P + T_S - T_F - \xi - \bar{I} + P_B \left[\xi - \bar{I} - \frac{T_P}{2} \right] \\ & + \frac{\xi + T_F - T_P - \bar{I}}{P_S} + \frac{(P_B - 1)(T_P + \bar{I})}{P_C} + \frac{T_P + \bar{I}}{P_C P_S} \end{aligned} \quad (3.15)$$

Eq.(3.15) may be used for both FAMA-PJ and Slotted FAMA-PJ by a simple substitution of the appropriate values for each protocol.

3.4.1 FAMA-PJ

Some of the parameters for Eq.(3.15) have been derived previously in Section 3.3. $T_S = T$, is given in Eq.(3.3). The probability of success, P_S , is derived in the proof of Theorem 3.2, along with \bar{I} and \bar{B} (given in Eq.(3.5)).

The probability of finding the channel busy P_B is equal to the average busy period divided by the average cycle time

$$\begin{aligned} P_B &= \frac{\bar{B}}{(\bar{B} + \bar{I})} \\ &= \frac{e^{-\lambda\tau}(\delta - 2\tau) + \gamma + 4\tau + \varepsilon}{e^{-\lambda\tau}(\delta - 2\tau) + \gamma + 5\tau + 2\varepsilon + \frac{1}{\lambda}} \end{aligned} \quad (3.16)$$

The probability of completing a backoff waiting period P_C is equivalent to having no other arrivals during the waiting period. The length of an average waiting period is ξ seconds, therefore

$$P_C = e^{-\lambda\xi} \quad (3.17)$$

The delay incurred by a station failing at the attempt to acquire the floor is the length of an RTS plus the length of the jamming period. If the station is the first station

in the failed period it will incur a delay of $\gamma + \varepsilon + 4\tau$, if it is the last station in the failed period it will incur a delay of $\gamma + \varepsilon + 4\tau - \bar{Y}$, where \bar{Y} is the average time of the last arrival in a failed period. The average delay incurred by all stations in a failed period is therefore

$$T_F = \gamma + \varepsilon + 4\tau - \frac{\bar{Y}}{2} \quad (3.18)$$

As in non-persistent CSMA [KT75], The cumulative distribution function for Y is the probability that no arrivals occur in the interval of length $\tau - y$ and equals $F_Y(y) = e^{-\lambda(\tau-y)}$ [KT75] (where $y \leq \tau$); therefore the expected value of Y is [KT75]

$$\bar{Y} = \tau - \frac{(1 - e^{-\lambda\tau})}{\lambda}. \quad (3.19)$$

Substituting Eq.(3.19) in Eq. (3.18) we obtain

$$T_F = \gamma + \varepsilon + \frac{7}{2}\tau + \frac{(1 - e^{-\lambda\tau})}{2\lambda} \quad (3.20)$$

The average transmission period T_P is equivalent to an average busy period, there $T_P = \bar{B}$ and is given in Eq.(3.5).

Substituting these values into Eq.(3.15) for FAMA-PJ we obtain

$$\begin{aligned} \bar{D}_{PJ} = & e^{-\lambda\tau}(\delta - 2\tau) + \delta + \gamma + \frac{3}{2}\tau + \frac{(1 + e^{-\lambda\tau})}{2\lambda} - \xi + P_B \left(\xi - \frac{1}{\lambda} - \frac{e^{-\lambda\tau}(\gamma + 6\tau + 3\varepsilon)}{2} \right) \\ & + \frac{e^{-\lambda\tau}(2\tau - \delta) - \frac{3}{2}\tau - \varepsilon + \xi - \frac{(1+e^{-\lambda\tau})}{2\lambda}}{e^{-\lambda\tau}} + \frac{e^{-\lambda\tau}(\delta - 2\tau) + \gamma + 5\tau + 2\varepsilon + \frac{1}{\lambda}}{e^{-\lambda(\tau+\xi)}} \\ & + \frac{(P_B - 1) \left[e^{-\lambda\tau}(\delta - 2\tau) + \gamma + 5\tau + 2\varepsilon + \frac{1}{\lambda} \right]}{e^{-\lambda\xi}} \end{aligned} \quad (3.21)$$

3.4.2 Slotted FAMA-PJ

As for FAMA-PJ many of the values needed for Eq.(3.15) have already been determined in Section 3.3. $T_S = T$ and is given in Eq.(3.3). $T_P = \bar{B}$ given in Eq.(3.8) and

P_S is given in Eq.(3.7). Also, the value used for ξ , the average backoff timer, is an exact multiple of τ .

The probability P_B of finding the channel busy on arrival is found from \overline{B} and \overline{I} , given in Eqs.(3.8) and (3.10), and equals

$$\begin{aligned} P_B &= \frac{\overline{B}}{(\overline{B} + \overline{I})} \\ &= \frac{\left[\frac{\lambda\tau e^{-\lambda\tau}}{(1-e^{-\lambda\tau})} \right] (\delta - 2\tau) + \gamma + 4\tau + \varepsilon}{\left[\frac{\lambda\tau e^{-\lambda\tau}}{(1-e^{-\lambda\tau})} \right] (\delta - 2\tau) + \gamma + 5\tau + 2\varepsilon + \frac{\tau}{(1-e^{-\lambda\tau})}} \end{aligned} \quad (3.22)$$

The probability P_C of the backoff timer expiring is equal to the probability of no arrivals during this period. This is the same as for FAMA-PJ and is given in Eq.(3.17).

In a failed attempt to gain the floor a station incurs the time to transmit the RTS, a 2τ propagation delay before the jamming signal arrives, and the length of the jamming signal. Therefore, a failed attempt to transmit, $T_F = \gamma + \varepsilon + 4\tau$, Substituting these into Eq.(3.15) for FAMA-PJ we obtain

$$\begin{aligned} \overline{D}_{SPJ} &= \frac{\lambda\tau e^{-\lambda\tau}}{(1-e^{-\lambda\tau})}(\delta - 2\tau) + \delta + \gamma + 3\tau - \xi - \frac{\tau}{(1-e^{-\lambda\tau})} \\ &+ P_B \left[\xi - \frac{\tau}{(1-e^{-\lambda\tau})} - \frac{\frac{\lambda\tau e^{-\lambda\tau}}{(1-e^{-\lambda\tau})}(\delta - 2\tau) + \gamma + 6\tau + 3\varepsilon}{2} \right] \\ &+ \frac{\xi - \frac{\lambda\tau e^{-\lambda\tau}}{(1-e^{-\lambda\tau})}(\delta - 2\tau) - \frac{\tau}{(1-e^{-\lambda\tau})} - \tau - \varepsilon}{\frac{\lambda\tau e^{-\lambda\tau}}{(1-e^{-\lambda\tau})}} \\ &+ \frac{(P_B - 1) \left[\frac{\lambda\tau e^{-\lambda\tau}}{(1-e^{-\lambda\tau})}(\delta - 2\tau) + \gamma + 5\tau + 2\varepsilon + \frac{\tau}{(1-e^{-\lambda\tau})} \right]}{e^{-\lambda\xi}} \\ &+ \frac{\frac{\lambda\tau e^{-\lambda\tau}}{(1-e^{-\lambda\tau})}(\delta - 2\tau) + \gamma + 5\tau + 2\varepsilon + \frac{\tau}{(1-e^{-\lambda\tau})}}{e^{-\lambda\xi} \frac{\lambda\tau e^{-\lambda\tau}}{(1-e^{-\lambda\tau})}} \end{aligned} \quad (3.23)$$

3.4.3 Throughput-Delay Characteristics

Figure 3.10 compares the delays obtained in FAMA-PJ and non-persistent CSMA. The graph uses $\tau = 0.0001$, $\gamma = 0.02$, $\varepsilon = 0.002$ and $\xi = 5 \times \gamma$ (the size of our RTS packet) for FAMA-PJ, and $\xi = T$ (the size of a data packet) for non-persistent CSMA. FAMA-PJ shows to be more stable than non-persistent CSMA, maintaining a high throughput during periods of high traffic conditions. Our results also indicate that the addition of collision detection by the sender does not impact the delay characteristics appreciably. The delay for both FAMA-PJ and slotted FAMA-PJ are never worse than that of non-persistent CSMA under low-traffic conditions, and perform much better than non-persistent CSMA under high-traffic conditions. This is an important finding, as it shows that the cost of providing a more stable channel access strategy that eliminates collisions of data packets dynamically using collision of control packets by the sender is negligible compared to using non-persistent CSMA.

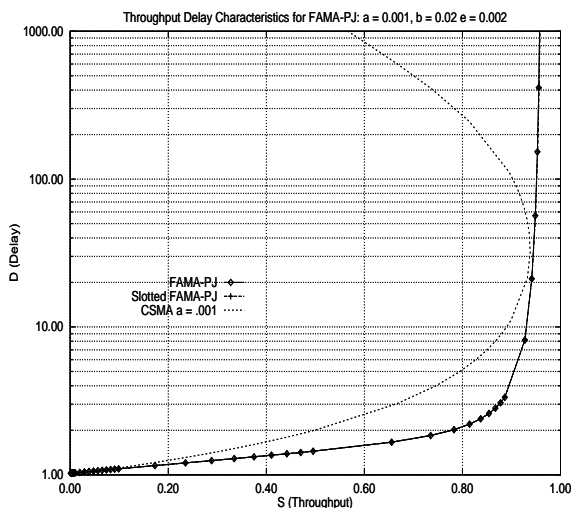


Figure 3.10: Delay performance of FAMA-PJ and slotted FAMA-PJ.

3.5 Conclusions

We have specified and analyzed the floor acquisition multiple access using pauses and jamming (FAMA-PJ) protocol. We have shown that the protocol guarantees that a station will be able to transmit one or more data packets with no collision from other stations transmissions. We have also provided an analysis of the average delay characteristics of FAMA-PJ and compared it with non-persistent CSMA. Our results show that the delay costs of providing floor acquisition using collision detection in a WLAN is the same or better than that for non-persistent CSMA. In addition, our results indicate that a floor acquisition strategy can be more stable than non-persistent CSMA. The results also show slotting provides little improvement in throughput over the unslotted version of the protocol in a high-speed WLAN.

Chapter 4

FAMA for Ad-Hoc Networks

Multihop packet-radio networks (i.e., ad-hoc networks) extend packet switching technology into environments with mobile users, can be installed quickly in emergency situations, and are self configurable [LNT87]. As such, they are likely to play an important role in the future of computer communication. The medium access control (MAC) protocol with which packet-radios (or stations) can share a common broadcast channel is essential in a packet-radio network. CSMA (carrier sense multiple access) protocols [KT75] have been used in a number of packet-radio networks in the past [LNT87]; these protocols attempt to prevent a station from transmitting simultaneously with other stations within its transmitting range by requiring each station to listen to the channel before transmitting.

The hardware characteristics of packet-radios are such that a packet-radio cannot transmit and listen to the same channel simultaneously; therefore, collision detection (CSMA/CD [MB76]) cannot be used in a single-channel packet-radio network. The throughput of CSMA protocols is very good, as long as the multiple transmitters within range of the same receivers can sense one another's transmissions. Unfortunately, "hidden terminal"

problems [TK75] degrade the performance of CSMA substantially, because carrier sensing cannot prevent collisions in that case.

The busy tone multiple access (BTMA) protocol [TK75] was the first proposal to combat the hidden-terminal problems of CSMA. BTMA is designed for station-based networks and divides the channel into a message channel and the busy-tone channel. The base station transmits a busy-tone signal on the busy-tone channel as long as it senses carrier on the data channel. Because the base station is in line of sight of all terminals, each terminal can sense the busy-tone channel to determine the state of the data channel. The limitations of BTMA are the use of a separate channel to convey the state of the data channel, the need for the receiver to transmit the busy tone while detecting carrier in the data channel, and the difficulty of detecting the busy-tone signal in a narrow-band channel.

A receiver initiated busy-tone multiple access protocol for packet-radio networks has also been proposed [WL87]. In this scheme, the sender transmits a request-to-send (RTS) to the receiver, before sending a data packet. When the receiver obtains a correct RTS, it transmits a busy tone in a separate channel to alert other sources nearby that they should backoff. The correct source is always notified that it can proceed with transmission of the data packet. The limitations of this scheme is that it still requires a separate busy-tone channel and full-duplex operation at the receiver.

One of the first protocols for wireless networks based on a handshake between sender and receiver was the SRMA (split-channel reservation multiple access) [TK76]. According to SRMA, the sender of a packet uses ALOHA or CSMA to decide when to send a request-to-send (RTS) to the receiver. In turn, the receiver responds with a clear-to-send (CTS) if it receives the RTS correctly; the CTS tells the sender when to transmit its data

packet. Although SRMA was proposed with one or two control channel for the RTS/CTS exchange, the same scheme applies for a single channel.

Since the time SRMA was first proposed, several other medium access control (MAC) protocols have been proposed for either single-channel wireless networks or wireline local area networks that are based on similar RTS-CTS exchanges, or based on RTSs followed by pauses [BDSZ94, Vad95, Col83, LM84, Rom86, SAO90]. Karn [Kar90] proposed a protocol called MACA (multiple access collision avoidance) to address the problems of hidden terminals in single-channel networks. MACA amounts to a single-channel SRMA using ALOHA for the transmission of RTSs; it attempts to detect collisions at the receiver by means of the RTS-CTS exchange without carrier sensing. The IEEE 802.11 MAC protocol for wireless LANs includes a transmission mode based on an RTS-CTS handshake (DFWMAC [Che94, IEE97]).

In this chapter, we introduce a new variation on MAC protocols based on RTS-CTS exchanges that is particularly attractive for ad-hoc networks. We call the new protocol FAMA-NCS (floor acquisition multiple access with non-persistent carrier sensing). The objective of FAMA-NCS is for a station that has data to send to acquire control of the channel in the vicinity of the receiver (which we call “the floor”) before sending any data packet, and to ensure that no data packet collides with any other packet at the receiver.

Ensuring that floor acquisition is enforced among competing senders hidden from one another and who have requested the floor (i.e., sent an RTS) can only be achieved by the receivers. Accordingly, in FAMA-NCS, the length of a CTS is longer than the duration of an RTS and ensures that the CTS from a receiver lasts long enough for any hidden sender that did not hear the RTS being acknowledged to hear what amounts to a jamming

signal from the receiver. Section 4.1 describes FAMA-NCS and a variant of FAMA based on packet sensing (i.e., the transmission of RTSs without carrier sensing), which amounts to MACA or SRMA using ALOHA.

Although the original motivation for such protocols as MACA, IEEE 802.11 DFW-MAC, MACAW [BDSZ94], and BAPU [Vad95] was to solve the hidden-terminal problems of CSMA by using RTS-CTS handshakes, it is easy to show by example that simply introducing three-way handshakes (RTS-CTS-data) or even more complex handshakes (RTS-CTS-data-ACK or others) does not suffice to eliminate all instances in which two or more senders are led to believe that they can transmit data packets to their intended receivers, only to create collisions. This is the case even if carrier sensing and RTS-CTS based handshakes are used in combination. Section 4.2 verifies a sufficient condition for correct floor acquisition in single-channel networks with hidden terminals. We show that carrier sensing is necessary in protocols based on RTS-CTS handshakes to eliminate hidden-terminal problems efficiently in single-channel networks in which nodes can transmit control packets without using time synchronization.

Section 4.3 analyzes the throughput of FAMA-NCS in fully-connected networks, wireless LANs with hidden terminals, and ad-hoc networks. The objective of our analysis is to address several important questions: How useful is carrier sensing when RTS-CTS handshakes are used? What is the impact of the RTS-CTS overhead on the performance of the network? How important is the role of the CTS as a busy-tone signal? Our analysis shows that, with or without hidden terminals, protocols that use carrier sensing in combination with RTS-CTS handshakes attain higher throughput than protocols that do not use carrier sensing. In wireless LANs with hidden terminals or ad-hoc networks, FAMA-

NCS achieves higher throughput than ALOHA, CSMA, MACA, and DFWMAC, which is due to the CTSs acting as same-channel busy tones. Due to space considerations, we do not address the average delay of FAMA protocols; however, it is easy to show that FAMA protocols provide smaller average delays than CSMA [FG95a].

Section 4.4 compares by simulation the performance of FAMA-NCS with MACAW and DFWMAC. Our results show very clearly that carrier sensing at the sender and the longer duration of CTSs compared to RTSs are critical to the performance and simplicity of MAC protocols based on RTS-CTS handshakes for networks with hidden-terminals in which nodes can transmit packets asynchronously. The simulations also help to validate our analytical results. Section 4.5 discusses related work; FAMA-NCS is the first proposal based on RTS-CTS exchanges that recognizes the importance of making a CTS act as a busy tone for half-duplex operation.

4.1 FAMA Protocols

4.1.1 Overview

FAMA-NCS requires a station who wishes to send one or more packets to acquire the floor before transmitting the packet train. The floor is acquired using an RTS-CTS exchange multiplexed together with the data packets in such a way that, although multiple RTSs and CTSs may collide, data packets are always sent free of collisions. The basic principles of floor acquisition are inspired on earlier work by Kleinrock and Tobagi on BTMA [TK75], the use of RTS-CTS exchanges first described for SRMA [TK76], and the provision of priorities among packets introduced for the transmission of priority acknowledgments in

ALOHA and CSMA [TK78].

To acquire the floor, a station sends an RTS using either packet sensing or carrier sensing. The first variant corresponds to using the ALOHA protocol for the transmission of RTSs; the second consists of using a CSMA protocol to transmit RTSs. A station sends a CTS after receiving an error-free RTS addressed to it. When a station receives an error-free CTS, it knows that the floor has been acquired by the station to whom the CTS is addressed. After floor acquisition, either the floor holder or any of the receivers addressed by the floor holder are able to send data packets and acknowledgments free of collisions over the channel. Any reliable link control scheme can be implemented on top of FAMA-NCS between the floor holder and the stations with whom it wishes to communicate. This is accomplished by forcing stations that do not have the floor to wait a predefined minimum amount of time (at least twice the maximum propagation delay) before being able to bid for the floor. This is similar to the schemes for the provision of priority acknowledgments proposed for CSMA and ALOHA by Kleinrock and Tobagi [TK78].

To ensure that floor acquisition is enforced among competing senders hidden from one another and who have requested the floor (i.e., sent an RTS), the CTS sent by a receiver is guaranteed to last long enough (or to be repeated enough times) to jam any hidden sender that did not hear the RTS being acknowledged. This corresponds to a single-channel BTMA scheme in which sensing of error-free CTSs (for packet sensing) or the carrier of a CTS (for carrier sensing) over the data channel is used instead of the busy-tone signal.

When a station with data to send fails to acquire the floor or detects the floor being held by another station, it must reschedule its bid for the floor. This can be done using different persistence and backoff strategies. In this chapter, we choose to consider non-

persistent protocols over persistent protocols, because the throughput of non-persistent CSMA is much higher under high load and only slightly lower under low load than the throughput of p-persistent CSMA [KT75]. We also specify FAMA-NCS as using a uniform distribution when choosing backoff times; however, other backoff strategies can be adopted (e.g., see those proposed for MACAW [BDSZ94]).

To simplify our analysis and description of FAMA-NCS, we do not address the effect of acknowledgments in the rest of this chapter, and assume the simplest three-way handshake (RTS-CTS-data) with no acknowledgments.

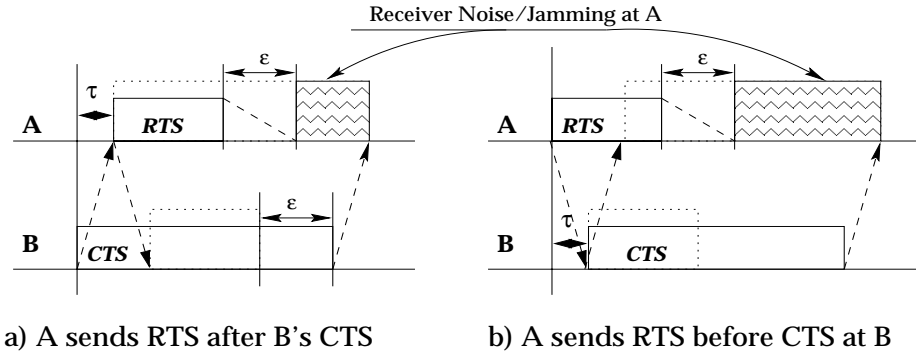


Figure 4.1: Dominance of the CTS in FAMA for hidden-terminal:

- a) A begins its RTS just as CTS arrives at A
- b) A begins RTS τ seconds in advance of the CTS

Figure 4.1 shows how the CTS dominance operates in more detail. Station B is sending a CTS while station A is attempting to send its RTS and acquire the floor. Because stations use carrier sensing, A must send its RTS within τ seconds of the start of B 's CTS; otherwise one of the stations would detect carrier and back off. Figure 4.1a illustrates the case in which B 's CTS arrives at A just as A begins its RTS transmission. Because B 's CTS is longer than the RTS plus the transmit to receive turnaround time, A hears the

overlap as noise and backs off. Figure 4.1b) illustrates the other possible case, in which A can begin its RTS before B starts sending its CTS. A can start its transmission no earlier than τ seconds before B begins its CTS transmission; otherwise, A would have interfered with the RTS being sent to B and no CTS would have been transmitted by B). In this case, the CTS arrives at A 2τ seconds after A 's RTS began. Again, because the CTS is longer than the RTS plus the transmit to receive turnaround time, A hears the end of the CTS as noise and backs off.

4.1.2 FAMA-NCS

FAMA-NCS combines non-persistent carrier sensing with the RTS-CTS exchange. This is similar to SRMA with CSMA, IEEE 802.11 DFWMAC [Bib92], and Apple's Local Talk Link Access protocol [SAO90]. However, none of these and other protocols based on carrier sensing and RTS-CTS handshakes provide floor acquisition in networks with hidden terminals.

The length of a CTS in FAMA-NCS is larger than the aggregate of the length of an RTS plus one maximum round trip time across the channel, the transmit to receive turn around time, and any processing time. The length of an RTS is larger than the maximum channel propagation delay plus the transmit-to-receive turn around time and any processing time. This is required to avoid one station hearing a complete RTS before another has started to receive it. The relationship of the size of the CTS to the RTS gives the CTS *dominance* over the RTS in the channel. Once a station has begun transmission of a CTS, any other station within range of it that transmits an RTS simultaneously (i.e., within one propagation delay of the beginning of the CTS) will hear at least a portion of

the dominating CTS and backoff, thereby letting the data packet that will follow to arrive free from collision. The dominating CTS plays the role of a busy tone.

Figure 4.2 specifies FAMA-NCS in detail. The specification assumes that the turnaround times of the radios are longer than the maximum round-trip time between any two nodes, which is the case with existing commercial-off-the-shelf (COTS) radios operating in ad-hoc networks and wireless LANs.

A station that has just been initialized must wait the time it takes to transmit the maximum-size data packet in the channel plus one maximum round-trip time across the channel. This allows any neighboring station involved in the process of receiving a data packet to complete the reception un-obstructed. The initialization time also gives the station the ability to learn of any local traffic in progress. If no carrier is detected during the initialization period, the station transitions to the PASSIVE state. Otherwise, it transitions to the REMOTE state. A station can only be in the PASSIVE state if it is properly initialized (i.e., has no packet to send, and senses an idle channel). In all other states, the station must have listened to the channel for a time period that is sufficiently long for any neighbor involved in receiving data to have finished.

A station that is in the PASSIVE state and detects carrier on the channel transitions to the REMOTE state. Alternatively, a station that receives a packet to send in the PASSIVE state transmits an RTS and transitions to the RTS state. The sending station waits long enough for the destination to send the CTS. If the CTS packet is not received within the time allowed, the sender transitions to the BACKOFF state. If the sender hears noise on the channel after its RTS, it assumes a collision with a neighbor's dominating CTS and waits long enough for a maximum-length data packet to be received. Otherwise,

upon receiving the CTS, the sender transmits its data packet. Because the CTS could be corrupted at the sender, once the destination station sends its CTS, it only needs to wait one maximum round-trip time to sense the beginning of the data packet from the source. If the data packet does not begin, the destination transitions either to the BACKOFF state (if it has traffic pending) or to the PASSIVE state.

In the BACKOFF state, if no carrier is detected during the entire backoff waiting period computed by the station, the station transmits an RTS and transitions to the RTS state as before. Otherwise, upon sensing carrier the station transitions to the REMOTE state.

For stations in the REMOTE state, FAMA-NCS enforces different waiting periods on passive stations (those stations not directly involved in the current transmission period) based on what was last heard on the channel. Any passive station that detects carrier transitions to the REMOTE state, and after the channel clears the waiting period is determined as follows:

- After hearing an RTS for another station, the station must wait long enough for a CTS to be transmitted by the destination and received by the sender, and the data packet to begin.
- After hearing a CTS from another station, the station must wait long enough to allow the other station to receive its data packet.
- After hearing a data packet, the waiting time is the enforced FAMA waiting period.
- After hearing noise (colliding control packets) on the channel, the waiting period must be long enough to allow another station time to receive a maximum size data packet.

```

Variable Definitions
CD = Carrier Detected
 $T_{PROP}$  = Maximum channel propagation delay
 $T_{PROC}$  = Processing time for carrier detection
 $T_{TR}$  = Transmit to receive turn-around time
 $\gamma$  = Time to transmit an RTS packet
 $\gamma'$  = Time to transmit a CTS packet
 $\delta$  = Time to transmit a maximum sized data packet
Burst = Number of packets to send in a burst
 $T_{FAMA} = 2 \times T_{PROP} + T_{TR} + T_{PROC}$ 

Procedure START()
Begin
  Timer  $\leftarrow \delta + 2 \times T_{PROP}$ 
  While( $\overline{CD}$   $\wedge$  Timer not expired) wait
  If (CD) Then call REMOTE( $(\delta + T_{FAMA})$ ,TRUE)
  Else call PASSIVE()
End

Procedure PASSIVE()
Begin
  While( $\overline{CD}$   $\wedge$  No Local Packet) wait
  If (CD) Then call REMOTE( $(\delta + T_{FAMA})$ ,FALSE)
  Else Begin
    Burst  $\leftarrow$  maximum burst
    Transmit RTS Packet
    call RTS( $T_{FAMA}$ )
  End End

Procedure RTS( $T_{\sigma}$ )
Begin
  Timer  $\leftarrow T_{\sigma}$ 
  While( $\overline{CD}$   $\wedge$  Timer not expired) wait
  If (Timer Expired) Then call BACKOFF()
  Else Begin
    Receive Packet
    DO CASE of (received packet type)
      Begin
        CTScall XMIT()
        Default:
          call REMOTE( $(\delta + T_{FAMA})$ ,TRUE)
      End
    End
  End
End

Procedure BACKOFF()
Begin
  Timer  $\leftarrow$  RANDOM(1,10  $\times$   $\gamma'$ )
  While( $\overline{CD}$   $\wedge$  Timer not expired) wait
  If (CD) Then call REMOTE( $(\delta + T_{FAMA})$ ,FALSE)
  Else Begin
    Burst  $\leftarrow$  maximum burst
    Transmit RTS Packet
    call RTS( $T_{FAMA}$ )
  End End

Procedure XMIT()
Begin
  Wait  $T_{TR}$ 
  If ( $(Burst > 1) \wedge$  Local Packet)
  Then Begin
    Mark MORE flag in header
    Transmit Data Packet
    Burst  $\leftarrow$  Burst - 1
    call RTS( $T_{FAMA}$ )
  End
  Else Begin
    Transmit Data Packet
    Timer  $\leftarrow T_{FAMA}$ 
    While(Timer not expired) wait
    If (Local Packet) Then call BACKOFF()
    Else call PASSIVE()
  End End

Procedure REMOTE( $T_{\sigma}$ ,dflag)
Begin
  Timer  $\leftarrow T_{\sigma}$ 
  While( $\overline{CD}$   $\wedge$  Timer not expired) wait
  If (Timer Expired)
  Then Begin
    If (Local Packet) Then call BACKOFF()
    Else call PASSIVE()
  End
  Else Begin
    Receive Packet
    DO CASE of (received packet type)
      Begin
        RTS:
          If(dflag= TRUE) call REMOTE( $T_{\sigma}$ ,TRUE)
          If(Destination ID = Local ID)
          Then Begin
            Wait  $T_{TR}$ 
            Transmit CTS Packet
            call REMOTE( $(T_{FAMA})$ ,TRUE)
          End
          call REMOTE( $(\gamma' + T_{FAMA})$ ,TRUE)
        CTS:
          call REMOTE( $(\delta + T_{FAMA})$ ,TRUE)
        DATA:
          If(Destination ID = Local ID)
          Then Begin
            Pass packet to upper layer
            If (MORE flag set in header)
            Then Begin
              Transmit CTS
            End
            call REMOTE( $(T_{FAMA})$ ,TRUE)
          End
          Else Begin
            If (MORE flag set in header)
            Then Begin
              call REMOTE( $(\gamma' + T_{FAMA})$ ,TRUE)
            End
            Else Begin
              call REMOTE( $(T_{FAMA})$ ,TRUE)
            End
          End
        ERROR:
          call REMOTE( $(\delta + T_{FAMA})$ ,TRUE)
      End
    End
  End
End

```

Figure 4.2: FAMA-NCS Specification

The channel becomes idle when all stations are in either the PASSIVE or BACK-OFF state. The next access to the channel is driven by the arrival of new packets to the network and retransmission of packets that have been backed off.

To increase the efficiency of the channel, a station that has successfully acquired

the floor can dynamically send multiple packets together in a train, bounded by an upper limit. To allow this to be successful in a hidden-terminal environment, the destination station must alert its neighbors that it has more data packets coming, and to continue to defer their transmissions. FAMA-NCS uses a simple handshake mechanism to support packet trains.

If the sending station has multiple packets to send, it sets a MORE flag in the header of the data packet. When the destination receives the data packet and sees the MORE flag set, it immediately responds with a CTS, just as when hearing an RTS. This CTS alerts all neighbors that might interfere with the next data packet that they must continue to defer.

Additionally, stations in the REMOTE state must extend their waiting period after hearing a data packet with the MORE flag set to allow additional time for the sender to receive the CTS from the destination signaling that it can receive the next data packet.

4.1.3 FAMA-NBR

The next solution we present is for WLANs that have a centralized, fixed location base station (BS) with mobile host (MH) stations that are always within range of the BS. The mobile hosts are assumed to move very slowly with respect to the time it takes to accomplish an RTS-CTS handshake and as such are considered fixed for the purposes of our discussion. MHs cannot hear all of the other MHs within a base station's area, and appear as hidden terminals to each other. In addition, neighboring base station areas do not overlap, but MHs of neighboring BSs can hear and interfere with each other. This protocol is a variant of FAMA-NPR (§4.1.2) with extensions based on the assumption that a central

BS is the receiver for MHs in its domain. The same RTS-CTS exchange and relative size of the RTSs and CTSs are used.

BSs and MHs operate somewhat differently. The protocol for the MHs is identical to that of FAMA-NPR with the minor change that MHs only respond and send to the BS, and therefore won't be repeated.

The major difference between the BS and MH operation is in the waiting periods during the REMOTE state. Because the BS cannot interfere with any data packets that it is not aware of, the timing constraints are much less conservative. (i.e., A MH that hears noise must wait the maximum time for a data packet to be transmitted, while a BS only needs to wait a minimum amount of time after the channel clears). Once the channel again becomes idle the BS determines how long of a wait period to set as follows:

- After hearing an RTS from any MH it responds immediately with a CTS and waits long enough to hear the beginning of the data packet.
- After receiving a data packet from a MH, or hearing noise on the channel, the BS waits the minimum required FAMA waiting period.

After the expiration of a wait period in the REMOTE state, the station (BS or MH) transitions to the PASSIVE state if no packets are pending, or the BACKOFF state otherwise.

4.1.4 FAMA-NPS

We present here a variant of FAMA that does not use carrier sensing and which we call FAMA-NPS (for non-persistent packet sensing). It basically amounts to MACA or

```

Variable Definitions
  TPROP = Maximum channel propagation delay
  TRTS = Transmission time of an RTS packet
  TCTS = Transmission time of a CTS packet
  TDATA = Transmission time of a DATA packet
  TTR = Time to transition from transmit to receive

Procedure START()
Begin
  Timer ← TDATA + TTR + 2TPROP
  While(Timer not expired) wait
  call PASSIVE()
End

Procedure PASSIVE()
Begin
  While(No Packet Received ∧ No Local Packet) wait
  If(Packet Received) Then call REMOTE(received packet)
  Else call RTS()
End

Procedure RTS()
Begin
  Transmit RTS
  Timer ← TCTS + TTR + 2TPROP
  While(Timer not expired ∧ No Packet Received) wait
  If(Timer expired) Then call BACKOFF()
  Else DO CASE of (received packet type)
  Begin
    Local CTS: call XMIT()
    Default: call REMOTE(received packet)
  End
End

Procedure BACKOFF()
Begin
  Timer ← RANDOM(1,10 × TRTS)
  While(Timer not expired ∧ No Packet Received) wait
  If(Timer expired) Then call PASSIVE()
  Else call REMOTE(received packet)
End

Procedure XMIT()
Begin
  Wait TTR
  Transmit Data Packet
  call PASSIVE()
End

Procedure REMOTE(packet)
Begin
  DO CASE of (packet type)
  Begin
    Local RTS:
      Wait TTR
      Transmit CTS
      timer ← TDATA + TTR + 2TPROP
    Other RTS:
      timer ← TCTS + TTR + 2TPROP
    CTS:
      timer ← TDATA + TTR + 2TPROP
    DATA:
      If(Local DATA) Then pass packet to upper layer
      call PASSIVE()
  End
  While(Timer not expired ∧ No Packet Received) wait
  If(Timer expired) Then call PASSIVE()
  Else call REMOTE(received packet)
End

```

Figure 4.3: FAMA-NPS Specification

a single-channel SRMA using ALOHA. Figure 4.3 specifies FAMA-NPS in detail.

Section 4.2 shows that, for a FAMA protocol with packet sensing to work with hidden terminals, the CTSs must be transmitted multiple times, which means that floor acquisition can be supported efficiently only in fully connected networks. Accordingly, our specification of FAMA-NPS assumes that it is used in a fully connected network and that a CTS is transmitted only once. RTSs and CTSs have the same duration, which is longer than one maximum round-trip delay.

A station that has a data packet to send and that is not expecting to hear a CTS or a data packet first transmits an RTS to the receiver. When a station processes a correct RTS, it defers transmission of any RTS for an amount of time specified in the RTS. If the RTS is addressed to the station, it sends a CTS and waits long enough for an entire data packet to arrive from the sender. Following the deferment specified by the RTS, a station

with a packet to send waits a random waiting period before it transmits an RTS.

MACA and improvements over it are also discussed in detail by Bharghavan, et al. [BDSZ94].

4.2 Correct Floor Acquisition

4.2.1 Using Carrier Sensing

For FAMA-NCS to provide correct floor acquisition, it must ensure that that each new packet, or any of its retransmissions, is sent to the channel within a finite time after it becomes ready for transmission, and that a data packet does not collide with any other transmission.

Theorem 10 below shows that FAMA-NCS provides correct floor acquisition if an RTS lasts longer than the maximum propagation delay and a CTS lasts longer than the time it takes to transmit an RTS, plus a maximum round-trip time and a maximum hardware transmit-to-receive transition time. We make the following assumptions to prove the theorem¹:

- A1) A packet sent over the channel that does not collide with other transmissions is delivered error free with probability $p > 0$.
- A2) A station sends an RTS to the intended destination and receives a CTS in return that does not collide with any other transmission with probability larger than 0.
- A3) All stations execute FAMA-NCS correctly.

¹Similar results can be obtained under different assumptions using a similar approach to the one presented here.

A4) The transmission time of an RTS is $\gamma < \infty$, the transmission time of a CTS is $\gamma' < \infty$, the maximum transmission time of a data packet is $\delta < \infty$, and the hardware transmit-to-receive transition time is $2\tau < \varepsilon < \infty$.

A5) There is no capture or fading on the channel.

A6) Any overlap by transmissions at a particular receiver causes that receiver to not understand either packet.

A7) The maximum end-to-end propagation time in the channel is $\tau < \infty$.

Theorem 10 *FAMA-NCS provides correct floor acquisition in the presence of hidden terminals, provided that $\gamma > \tau$ and $\gamma + 2\tau + \varepsilon < \gamma' < \infty$.*

Proof: Figure 4.4 illustrates any possible case of hidden terminals with respect to a given pair of source S and receiver R . Station L characterizes any neighbor of S that is hidden from R but can cause interference at S . Station K characterizes any neighbor of L hidden from S that can cause interference at L and can prevent L from following S 's dialogue with R . Similarly, Station X is a neighbor of R that is hidden from S but can cause interference at R ; and station Y is a neighbor of X that is hidden from R and can prevent X from following R 's dialogue with S . The proof must show that, if S sends a data packet to R , no other transmission can collide with it, regardless of the possible transmissions of other interfering nodes.

For S to be able to send data packets to R , it must first receive a CTS from R . Without loss of generality, assume that, at time t_0 , S sends an RTS to R .

Because the channel has a minimum propagation delay larger than 0, any neighbor of S (e.g., Station L) must start receiving S 's RTS at time $t_0^L > t_0$. If L receives S 's RTS

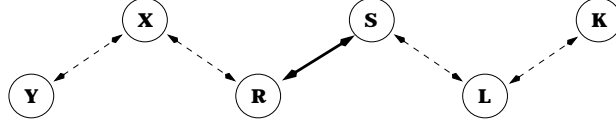


Figure 4.4: Stations involved in interference of the exchange between S and R

correctly, then it must back off for a period of time larger than $2\tau + \gamma'$ after the end of S 's RTS reaches L , which means that L backs off for $\gamma + 2\tau + \gamma'$ seconds after t_0^L . Alternatively, if the *RTS* reaches L in error or Station K 's transmission interferes with S 's RTS at Station L , then, starting with the end of carrier, Station L must back off for a period of time larger than $2\tau + \delta$. The minimum amount of time that L must back off then corresponds to the case in which the end of carrier coincides with the end of S 's RTS. Accordingly, L must back off for $\gamma + 2\tau + \delta$ seconds after t_0^L . It follows that the RTS sent by S at time t_0 forces any neighbor of S other than R to back off until time $t_1 > t_0 + \gamma + \gamma' + 2\tau$.

If the RTS is received at Station R with errors or collides with transmissions from other neighbors of R who are hidden from S (e.g., X), then R cannot send a CTS and S cannot send its data packet in return.

Assume that S 's RTS is received correctly by R at time t_2 . If S receives R 's CTS with errors or the CTS collides with transmissions from neighbors of S hidden from R (e.g., L), then S cannot send its data packet.

For the rest of the proof, assume that the RTS that S sends at time t_0 is received error free at station R within one maximum propagation delay, which means that R must start sending its CTS to S at time $t_2 \leq t_0 + \gamma + \tau$ (given that zero processing delays are assumed). This CTS must reach S within one maximum propagation delay after R sends it. Therefore, S must receive R 's entire CTS at time $t_3 \leq t_2 + \gamma' + \tau = t_0 + \gamma + \gamma' + 2\tau$.

Because $t_1 > t_3$, it follows that any potential interfering neighbor of S (e.g., L), must back off long enough for S to be able to receive R 's CTS without collisions.

Station S must start to receive R 's CTS no later than τ seconds after R starts its transmission, and must receive R 's entire CTS and send its data packet at time $t_4 \leq t_2 + \tau + \gamma'$. In turn, Station R must receive the end of S 's data packet by time $t_5 \leq t_4 + \delta + \tau \leq t_2 + 2\tau + \gamma' + \delta$.

On the other hand, any station X other than S within range of R must start receiving R 's CTS at time $t_2^X > t_2$. If X receives R 's CTS with no errors, then it must back off for a period of time larger than $2\tau + \delta$ after the end of R 's CTS reaches X , which means that X backs off for $2\tau + \delta + \gamma'$ seconds after t_2^X . Conversely, if R 's CTS reaches X in error or a transmission from one of its neighbors hidden from R , call it Y , interferes with the CTS, then, starting with the end of carrier, X must back off for more than $\delta + 2\tau$ seconds. The minimum amount of time that X backs off corresponds to the case in which the time when X detects the end of carrier equals the time when X receives R 's entire CTS; therefore, X must back off for $2\tau + \delta + \gamma'$ seconds after t_2^X . It follows that the CTS sent by R at time t_2 forces X and any neighbor of R other than S to back off until time $t_6 > t_2 + 2\tau + \gamma' + \delta$.

Because $t_6 > t_5$, it follows that Station X and any other potential interfering neighbor of R must back off long enough for R to be able to receive S 's data packet without collisions. Accordingly, it is true that FAMA-NCS allows a station to transmit a data packet only after a successful RTS-CTS exchange and no data packet collides with other transmissions. \square

Our assumption that $\epsilon > 2\tau$ is not necessary to make a FAMA protocol be correct,

but simplifies our equations and is consistent with the specifications of COTS radios and IEEE 802.11. In theory, to make the CTS dominance technique applicable to any value of $\epsilon \geq 0$, we would only need to require the sender of a data packet to wait for 2τ seconds after receiving a correct CTS, and for stations that back off to allow a possible data packet to go through to increase the back-off time by 2τ seconds.

4.2.1.1 FAMA-NBR

In FAMA-NBR the MH acts almost identically to the individual stations in FAMA-NCS; the difference is that all RTSs are directed to and received from a single station, the BS. Also, because the BS is the only station that sends data packets to the MHs within its range, it cannot interfere with any data packet transmissions to the MHs.

Theorem 11 *FAMA-NBR ensures that each new packet sent by a MH, or any of its re-transmissions, is sent to the channel within a finite time after it becomes ready for transmission, and that a data packet sent to any MH does not collide with any other transmission provided that $\gamma + 2\tau + \epsilon < \gamma' < \infty$.*

Proof:

Because the MHs follow FAMA-NPR, Theorem 10 holds for MHs receiving data packets free from collisions with packets from other MHs (even if they are not all within range of the same BS). \square

Theorem 12 *FAMA-NBR ensures that each new packet sent by a BS, or any of its re-transmissions, is sent to the channel within a finite time after it becomes ready for transmission,*

and that a data packet sent to any MH does not collide with any other transmission provided that $\gamma + 2\tau + \varepsilon < \gamma' < \infty$.

Proof:

The base stations in FAMA-NBR act almost exactly as stations in FAMA-NCS, with the exception that after hearing noise they must only wait 2τ seconds after the channel has cleared. This is because they are directly involved with any and all data packets sent from MHs within their range, and therefore cannot interfere with a data packet to one of the MHs within their range from some other hidden station. By substituting base stations BS_1 and BS_2 for stations S and Y , respectively, in Figure 4.4 it can be shown that FAMA-NBR provides collision free transmission of data packets from base stations to MHs as in Theorem 10. \square

4.2.2 Using Packet Sensing

The following example illustrates that a MAC protocol based on an RTS-CTS exchange and no carrier sensing cannot support floor acquisition efficiently in the presence of hidden terminals, because CTSs must be repeated several times to ensure that data packets never collide with other packets. We assume that RTSs and CTSs have the same duration.

Assume that Station S sends an RTS that is received correctly at Station R , then R immediately begins transmission of a CTS to S . Figure 4.5 shows two cases where the CTSs are not understood by stations in R 's neighborhood. In the first case, station X in R 's neighborhood transmits an RTS to R , blocking itself and all other stations in R 's neighborhood from understanding the first and second CTSs. In the second case, a station

in the neighborhood of X (and not R or S) transmits an RTS that blocks R 's CTS from X allowing X to transmit an RTS itself blocking additional CTSs. In either case, at least X does not understand the CTS and can transmit an RTS that collides at R with the data packet from S if not enough CTSs are sent by station R .

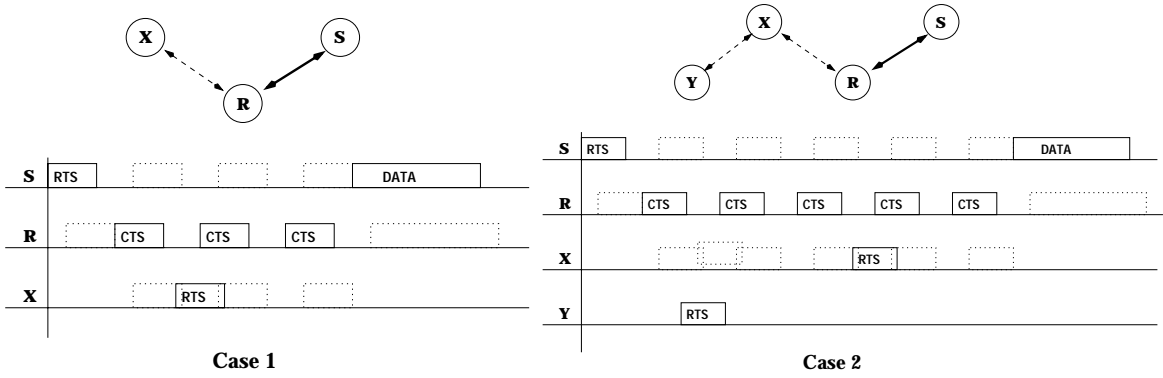


Figure 4.5: Non-persistent Packet Sensing with hidden terminals

To resolve the contention in the first case, the receiver needs to send at least three separate CTSs ($N = 1$). This is necessary, because a station considers the channel clear until any packet transmission is completely received free of error, and until that point there is no detection of traffic on the channel and transmissions are possible. Accordingly, station X can transmit its RTS just before the very end of receiving the CTS from R , and in the process also transmits over the beginning of the next CTS. X waits to get the CTS for it from R and instead sees the CTS to S , and defers further transmission.

In the second case, R must send at least five CTSs ($N = 2$). Here, the neighbor of X transmits an RTS that can collide with the first and second CTS blocking them from X , allowing it to send an RTS masking the third and fourth CTSs. The fifth CTS will be understood at X forcing it to defer after that point.

As the size of the network increases, any receiver R must send more and more CTSs to ensure that its neighbors are aware of its pending reception of a data packet, which renders the approach inefficient.

4.3 Comparative Throughput Analysis

4.3.1 Assumptions and Notations

We present an approximate throughput analysis that assumes the same traffic model first introduced for CSMA [KT75] to analyze the throughput of CSMA protocols, and the conditions for floor acquisition derived in Section 4.2.

As we have shown in Section 4.2, carrier sensing is needed to attain correct floor acquisition without sacrificing performance, which makes FAMA-NCS the only practical floor-acquisition solution; therefore, we analyze the throughput of FAMA-NCS only, and compare it against non-persistent ALOHA, CSMA, and MACA (i.e., FAMA-NPS). The throughput of non-persistent CSMA used in this analysis was reported by Tobagi and Kleinrock [KT75]. We have reported previously the throughput of FAMA-NPS [FG95b]. We compare these protocols in fully-connected networks, wireless LANs with hidden terminals, and ad-hoc networks.

We assume that there is an infinite number of stations who constitute a Poisson source sending RTS packets (for the case of FAMA), or new or retransmitted data packets (for the case of CSMA) to the the channel with an aggregate mean generation rate of λ packets per unit time. Any station can listen to the transmissions of any other station.

Each station is assumed to have at most one data block to send at any time. In all

protocols, a station transmits the entire data block as a single packet (which is the case of CSMA and MACA as it is described by Karn [Kar90]) or as multiple packets (which is the case of FAMA-NCS). The average size of a data block is δ seconds, RTSs last γ seconds, and CTSs last γ' seconds. The maximum end-to-end propagation delay of the channel is τ seconds. Collisions (e.g., RTS packets in FAMA-NCS, data packets in CSMA) can occur in the channel, and we assume that, when a station has to retransmit a packet, it does so after a random retransmission delay that is much larger than δ on the average. The average channel utilization is given by [KT75]

$$S = \frac{\bar{U}}{(\bar{B} + \bar{I})} \quad (4.1)$$

where \bar{B} is the expected duration of a busy period, defined to be a period of time during which the channel is being utilized; \bar{I} is the expected duration of an idle period, defined as the time interval between two consecutive busy periods; and \bar{U} is the time during a busy period that the channel is used for transmitting user data successfully.

The channel is assumed to introduce no errors, so packet collisions are the only source of errors, and stations detect such collisions perfectly. To further simplify the problem, we assume that two or more transmissions that overlap in time in the channel must all be retransmitted, and that a packet propagates to all stations in exactly τ seconds [KT75]. To reduce the number of variables used, we also consider that the turn-around times (ϵ) are part of the packet times, and still include the propagation delays in our computations. This provides a lower bound on the performance of the protocols we analyze.

Of course, this model is only a rough approximation of the real case, in which a finite number of stations access the same channel, stations can queue multiple packets for

transmission, and the stations' transmissions and retransmissions (of RTS or data packets) are correlated (e.g., a failed RTS is followed by another RTS within a bounded time). However, this model is a simple tool that helps us to understand why it is beneficial to listen for any type of channel activity, rather than for specific packet types, and provides additional insight on the performance of FAMA protocols and the impact of channel speed and propagation delay on the floor acquisition technique.

For the case of non-persistent CSMA, we assume [KT75] that a separate perfect channel is used for acknowledgments to let a station know when its packet was received free of collisions, and that all acknowledgments are sent reliably. Therefore, the throughput of non-persistent CSMA used for comparison with FAMA protocols is only an upper bound.

To facilitate the comparison of the various protocols numerically, the graphs showing average throughput versus traffic load normalize the results obtained for S by making $\delta = 1$ and introducing the following variables:

$$a = \tau/\delta(\text{normalized propagation delay})$$

$$G = \lambda \times \delta(\text{offered Load, normalized to data packets})$$

4.3.2 Throughput in Fully Connected Networks

Fig. 4.6 shows the transmission periods of FAMA-NCS. A transmission period begins with a source station transmitting an RTS at some time t_0 . The transmission is vulnerable for a period of τ seconds, during which another RTS from some other station may collide with it, causing the transmissions to fail. After the vulnerability period, if no other station has transmitted, all other stations will sense the channel busy, defer their transmissions, and the RTS transmission will be successful. According to FAMA-NCS, a

successful RTS is followed by the CTS response from the destination and the data packet(s) from the source. As Fig. 4.6 illustrates, because of the enforced waiting times and idle periods discussed in Section 4.1.2, a FAMA-NCS busy period is exactly one transmission period in length, either a successful or failed transmission, followed by an idle period.

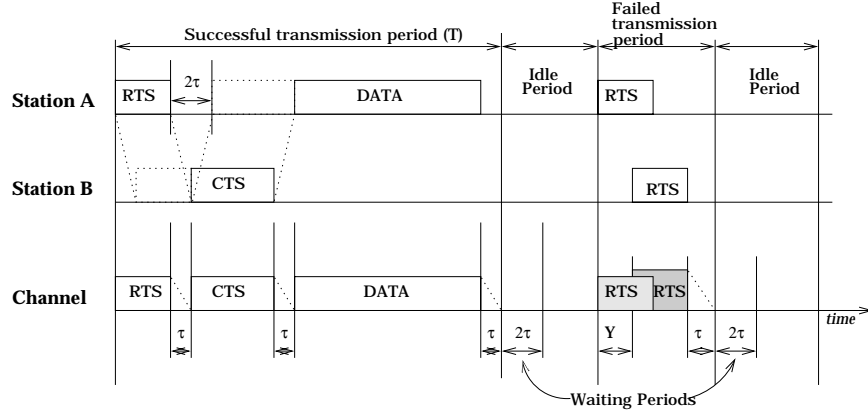


Figure 4.6: FAMA-NCS transmission periods

Theorem 13 *The throughput of FAMA-NCS is given by*

$$S = \frac{\delta}{\gamma' + \delta + 2\tau + \frac{1}{\lambda} + e^{\tau\lambda}(\gamma + 4\tau)} \quad (4.2)$$

Proof: A successful transmission consists of an RTS with one propagation delay to the intended recipient, a CTS and propagation delay back to the sender, and the data packet followed by a propagation delay. Accordingly, the time for a successful transmission, T , is

$$T = \gamma + \gamma' + 3\tau + \delta \quad (4.3)$$

Because FAMA-NCS guarantees that data packets sent after a successful RTS will not collide with any other packet, an unsuccessful transmission consists of one RTS being sent to the channel at time t_0 , followed by one or more RTSs transmitted by other

stations within a period of time of Y seconds (see Fig. 4.6), where $0 \leq Y \leq \tau$, plus one final propagation delay. Therefore, as in non-persistent CSMA, the duration of the average failed transmission period is given by [KT75] $T_{FAIL} = \gamma + \tau + \bar{Y}$. The cumulative distribution function for Y is the probability that no arrivals occur in the interval of length $\tau - y$ and equals [KT75] $F_Y(y) = e^{-\lambda(\tau-y)}$ (where $y \leq \tau$); therefore, the expected value of Y is [KT75] $\bar{Y} = \tau - (1 - e^{-\tau\lambda})/\lambda$.

The probability of success for an RTS, P_S , equals the probability that no arrivals occur in τ seconds, because there is a delay of τ seconds across the channel before all the other stations in the network detect the carrier signal. After this vulnerability period of τ seconds, all stations detect the carrier signal in the channel and defer their own transmissions. Therefore, given that the arrivals of RTSs to the channel are Poisson with parameter λ , we have

$$P_S = P\{\text{no arrivals in } \tau \text{ seconds}\} = e^{-\tau\lambda} \quad (4.4)$$

A busy period is successful with probability $e^{-\tau\lambda}$, and its length equals $(\gamma + \tau) + (\delta + \gamma' + 2\tau)$, where $\gamma + \tau$ accounts for the duration of an RTS and one propagation time, and $\delta + \gamma' + 2\tau$ accounts for the corresponding CTS, data packet, and their propagation times. As can be appreciated from Fig. 4.6, on the other hand, the length of an unsuccessful busy period equals $\gamma + \tau + Y$. Therefore, given that $y = 0$ in a successful busy period, the length of the average busy period is

$$\bar{B} = e^{-\tau\lambda}(\gamma' + \delta + 2\tau) + \gamma + 2\tau - \frac{(1 - e^{-\tau\lambda})}{\lambda} \quad (4.5)$$

The average utilization is the average amount of time during which useful data are sent during a successful busy period; therefore, we have

$$\bar{U} = \delta \cdot P_S = \delta e^{-\tau\lambda} \quad (4.6)$$

According to FAMA-NCS's definition, stations always incur a fixed time waiting period of 2τ seconds after each transmission period on the channel before making the transition to the PASSIVE or BACKOFF state (Fig. 4.2). Therefore, the average idle period can be expressed by

$$\bar{I} = \frac{1}{\lambda} + 2\tau \quad (4.7)$$

Substituting Eqs. (4.5), (4.6) and (4.7) in (4.1), we obtain Eq. (4.2). \square

We first compare the throughput of FAMA-NCS with that of non-persistent CSMA and FAMA-NPS in a fully connected network with a rate of 1 Mb/s, using both small data packets of 53 bytes (as in ATM cells) and longer packets of 400 bytes. We assume a network with a maximum diameter of 1 mile,² which gives us a propagation delay of approximately $5\mu\text{s}$. The minimum size of RTSs is 20 bytes to accommodate the use of IP addresses for destination and source, a CRC, and framing bytes. Fig. 4.7 shows the throughput (S) versus the offered load (G) for the various protocols under these conditions. These results indicate the importance of using carrier sensing as an integral part of the floor acquisition strategy. FAMA-NCS provides a much higher throughput than FAMA-NPS (i.e., MACA) or slotted FAMA-NPS. Of course, FAMA-NCS is more attractive for small values of $b = \gamma/\delta$ (Figure 4.8). It is also clear that using MACA (or its derivatives) in low or high-speed channels to transfer a single small packet for each successful RTS-CTS

²In practice, much shorter diameters are to be expected.

exchange is inefficient.

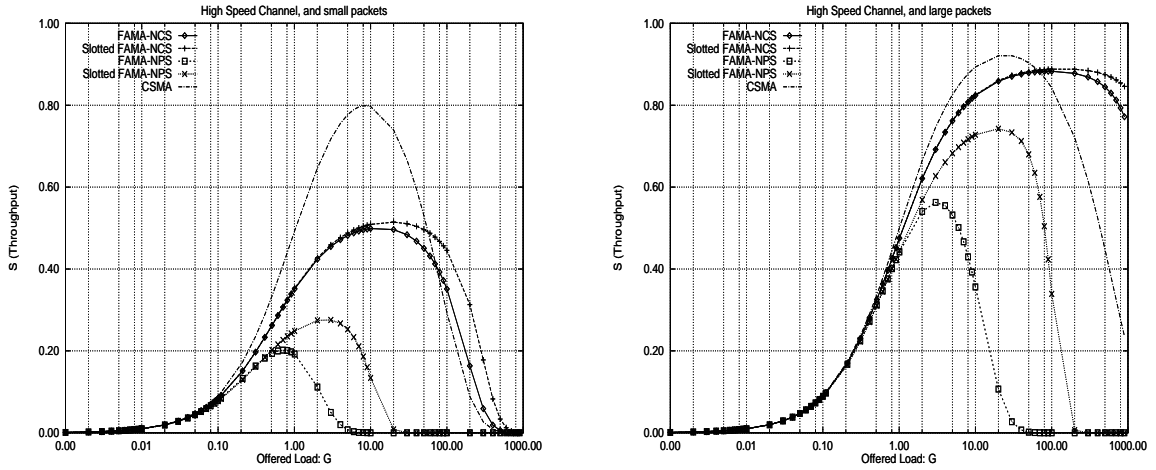


Figure 4.7: Throughput of FAMA-NCS, MACA (FAMA-NPS), and CSMA in a fully-connected network.

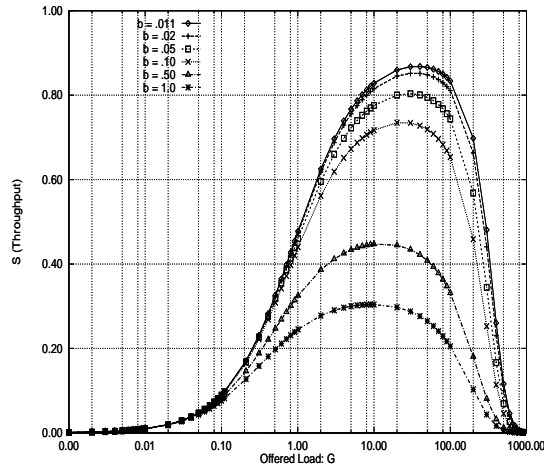


Figure 4.8: Throughput of FAMA-NCS for different values of b and $a = 0.01$.

4.3.3 Throughput in Wireless LANs

To study the performance of FAMA-NCS in wireless LANs with hidden-terminals, we adopt the same tractable model first used by Tobagi and Kleinrock [TK75] to analyze the impact of hidden terminals in CSMA. The model includes the same assumptions

made in Section 4.3.1, and a system configuration consisting of a large number of terminals communicating with a single base station over a single channel. All terminals are within line-of-sight and range of the base station, but they may be hidden from one another. The population of terminals is partitioned into N independent groups [TK75], such that all terminals within the same group can hear one another and the base station, and any two terminals from different groups are hidden from each other. Each group i consists of a large number of terminals who collectively form an independent Poisson source with an aggregate mean rate of λ_i floor requests per second, such that $\sum_i^N \lambda_i = \lambda$.

Theorem 14 *The throughput of FAMA-NCS for a system with N independent groups of hidden terminals is given by*

$$S = \frac{\delta \cdot P_S}{P_{SE} \cdot [e^{-\lambda_i \tau} \cdot (\delta + \gamma' + \gamma + 3\tau) + (1 - e^{-\lambda_i \tau}) (\gamma + \tau)] + (1 - P_{SE}) \left(\left[\frac{e^{\lambda(\gamma + \tau)} - 1 - \lambda(\gamma + \tau)}{\lambda(\gamma + \tau)(1 - e^{-\lambda(\gamma + \tau)})} \right] + (\gamma + \tau) \right) + \frac{1}{\lambda} + 2\tau \cdot P_S}$$

where $P_{SE} = \frac{1}{n} \sum_{i=1}^n \left(\prod_{j \neq i}^N e^{-\lambda_j (2\gamma)} \right)$, $P_S = \frac{1}{n} \sum_{i=1}^n \left(e^{-\lambda_i \tau} \prod_{j \neq i}^N e^{-\lambda_j (2\gamma)} \right)$ (4.8)

Proof: Consider the time line for the base station; it consists of a sequence of busy and idle periods. Because FAMA-NCS provides correct floor acquisition, collisions can occur only among RTSs. Therefore, because no successful RTS can overlap at all with any other RTS and because a successful transmission period is detected by all groups and forces an idle time of 2τ seconds, a busy period consists of either a single failed transmission period or a single successful transmission period.

An RTS originated from any node s in Group i is successful if no other RTS from any group collides with s 's RTS. Within Group i , the vulnerability period of s 's RTS is τ seconds, because all nodes in Group i can detect carrier τ seconds after the beginning of the RTS. Accordingly, an RTS is successful within its own Group i with probability $e^{-\lambda_i \tau}$.

In contrast, the vulnerability period of an RTS with respect to other groups is 2γ , because nodes in hidden groups cannot hear s 's transmissions and all transmissions take τ seconds to reach the base station. Accordingly, an RTS is successful with respect to a Group j other than its own with probability $e^{-\lambda_j(2\gamma)}$. Because all groups are independent, it follows that an RTS from Group i is successful at the base station with the following probability:

$$P_{S_i} = e^{-\lambda_i\tau} \prod_{j \neq i}^N e^{-\lambda_j(2\gamma)} \quad (4.9)$$

Therefore, the probability that an RTS from any one group is successful equals

$$P_S = \frac{1}{n} \sum_{i=1}^n \left(e^{-\lambda_i\tau} \prod_{j \neq i}^N e^{-\lambda_j(2\gamma)} \right) \quad (4.10)$$

A successful transmission period in the time line of the base station lasts T seconds, which is given in Eq. 4.3.

There are two types of failed transmission periods. If only one of the groups sends RTSs in a transmission period, its average duration in the time line of the base station equals $T_{F_1} = \gamma + \bar{Y}$, where \bar{Y} is the same as in the fully-connected network case. Note that T_{F_1} is not equal to T_{FAIL} of the fully-connected case, because nodes in a given Group i cannot hear RTSs from another group and can transmit at any instant after the end of a failed transmission period that does not involve Group i . Noting that $\bar{Y} \leq \tau$, we use the following bound for simplicity:

$$T_{F_1} \leq \gamma + \tau \quad (4.11)$$

The probability that an RTS from any given group is successful with respect to the rest of the other groups is given by

$$P_{SE} = \frac{1}{n} \sum_{i=1}^n \left(\prod_{j \neq i}^N e^{-\lambda_j(2\gamma)} \right) \quad (4.12)$$

If more than one group sends RTSs in a failed transmission period, the failed transmission period consists of multiple overlapping transmission periods with average durations of T_{F_1} seconds. Because groups are hidden and independent from each other, the length of the average failed transmission period in this case can be obtained by treating this case as an ALOHA channel with N stations, in which a station i corresponds to Group i and has an aggregate rate of λ_i . An average failed transmission period consists of a geometrically-distributed indefinite number (L) of interarrival times whose average duration is \bar{f} seconds (the average time between failed arrivals), plus the duration of an RTS (γ). The values for \bar{L} and \bar{f} are as derived by Takagi, et al. [TK85] for pure ALOHA as functions of λ and, according to our notation, δ . Substituting γ for δ in such results we obtain $e^{\lambda\gamma}$ and $(\lambda\gamma)^{-1} - e^{-\lambda\gamma}/(1 - e^{-\lambda\gamma})$, respectively. Therefore, when the first RTS of the transmission period collides with other RTSs, the average time of a failed transmission period, T_{FRTS} , equals

$$T_{FRTS} = \left[\frac{e^{\lambda\gamma} - 1 - \lambda\gamma}{\lambda\gamma(1 - e^{-\lambda\gamma})} \right] + \gamma \quad (4.13)$$

To make use of prior results, we make the simplifying assumption that N is very large. Accordingly, we approximate the average duration of the failed transmission period by substituting the upper bound of Eq. (4.11) for γ in Eq. (4.13), which yields

$$T_{F_2} = \left[\frac{e^{\lambda(\gamma+\tau)} - 1 - \lambda(\gamma+\tau)}{\lambda(\gamma+\tau)(1 - e^{-\lambda(\gamma+\tau)})} \right] + (\gamma + \tau) \quad (4.14)$$

Accordingly, the average busy period lasts

$$\bar{B} = P_{SE} \left(e^{-\lambda_i \tau} (T) + (1 - e^{-\lambda_i \tau}) (T_{F_1}) \right) + (1 - P_{SE}) (T_{F_2}) \quad (4.15)$$

Substituting Eqs. (4.12), (4.11) and (4.14) in the above Eq., we obtain

$$\begin{aligned} \bar{B} = & \frac{1}{n} \sum_{i=1}^n \left(\prod_{j \neq i}^N e^{-\lambda_j (2\gamma)} \right) \left(e^{-\lambda_i \tau} (\delta + \gamma' + \gamma + 3\tau) (1 - e^{-\lambda_i \tau}) (\gamma + \tau) \right) \\ & + \left(1 - \frac{1}{n} \sum_{i=1}^n \left(\prod_{j \neq i}^N e^{-\lambda_j (2\gamma)} \right) \right) \left(\left[\frac{e^{\lambda(\gamma+\tau)} - 1 - \lambda(\gamma + \tau)}{\lambda(\gamma + \tau)(1 - e^{-\lambda(\gamma+\tau)})} \right] + (\gamma + \tau) \right) \end{aligned} \quad (4.16)$$

The average idle period lasts 2τ seconds after every successful data packet transmission plus an average interarrival time of RTSs from all groups; therefore, we have

$$\bar{I} = \frac{1}{\lambda} + 2\tau \cdot P_S \quad (4.17)$$

The average utilization time is simply the proportion of time in which useful data are sent during a successful busy period, and

$$\bar{U} = \delta \cdot P_S \quad (4.18)$$

Substituting Eqs. 4.16, 4.17, and 4.18 into Eq. 4.1, we obtain the desired result.

□

In the limit, as $n \rightarrow \infty$, we obtain that the average throughput in any given system becomes

$$S = \frac{\delta}{\delta + \gamma' + \gamma + 5\tau + (e^{2\lambda\gamma} - 1) \left(\left[\frac{e^{\lambda(\gamma+\tau)} - 1 - \lambda(\gamma+\tau)}{\lambda(\gamma+\tau)(1 - e^{-\lambda(\gamma+\tau)})} \right] + \gamma + \tau \right) + \frac{e^{2\lambda\gamma}}{\lambda}} \quad (4.19)$$

The above result is just what should be predicted from the fact that FAMA-NCS supports correct floor acquisition. Together with Eq. 4.2, the above result indicates that,

as the number of hidden terminals increases with respect to any given group, FAMA-NCS degrades to the case in which the vulnerability period of an RTS becomes twice the length of the RTS, rather than the propagation delay. This is exactly the type of behavior of a packet-sensing FAMA protocol operating in a fully-connected network. Note that, because $\gamma \ll \delta$, this behavior is far better than the degradation experienced by CSMA, in which the vulnerability period of a packet becomes twice its length, which is the behavior of the ALOHA channel.

To visualize the above results, we compare FAMA-NCS and CSMA in wireless LANs with independent groups hidden from one another, and with one common central station. This type of experiment is similar to the ones used by Tobagi and Kleinrock [TK75].

Fig. 4.9 shows the maximum attainable throughput of ALOHA, slotted ALOHA, non-persistent CSMA, and FAMA-NCS versus an increasing numbers of independent groups (N). The results indicate that, FAMA-NCS's performance under hidden terminals becomes that of a packet-sensing FAMA protocol operating in a fully connected network, which is exactly the desired result. In contrast, as has been reported by Tobagi and Kleinrock [TK75], CSMA quickly degrades to ALOHA.

Another way to look at the behavior of FAMA-NCS in a wireless LAN with hidden terminals is by considering a complimentary-couple configuration. In this configuration, a fraction of the population is hidden from the rest. We use two independent groups ($N = 2$) and vary the size of one group versus the other, such that $S_1 = \alpha \cdot S$ and $S_2 = (1 - \alpha) \cdot S$. The total average arrival rate of RTSs is set to $G = 5.0$, which corresponds to the arrival rate at which the maximum throughput is obtained when $\alpha = 1/2$. Figure 4.10 shows the maximum

attainable throughput of FAMA-NCS versus α ; it is clear from the figure that FAMA-NCS suffers much smaller performance degradation with hidden terminals than CSMA does.

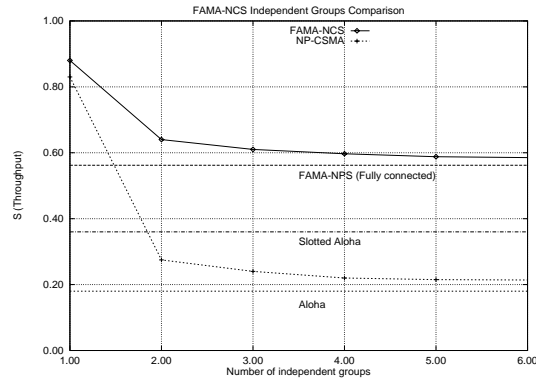


Figure 4.9: Throughput of FAMA protocols for increasing numbers of independent groups

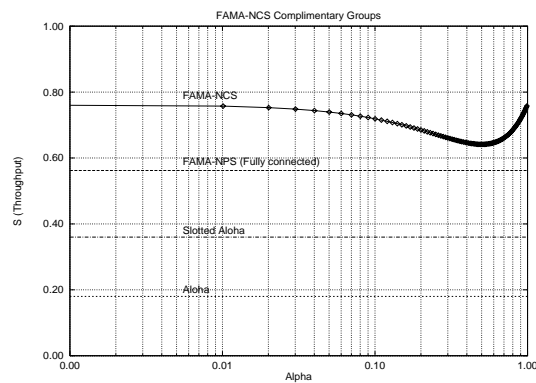


Figure 4.10: Throughput of FAMA-NCS in the complimentary couple configuration

4.3.4 Throughput in Ad-Hoc Networks

We analyze the throughput achieved at a given node in a single-channel ad-hoc network in which the neighbors of a node may be hidden from one another. For simplicity, we assume that any given node, w , has N neighbors. All nodes communicate over a single channel. The chosen node, w , is a Poisson source of RTSs with a mean rate of λ' floor

requests (RTS packets). Additionally, each neighboring node is an independent Poisson source of RTSs as well, with a mean rate of λ' RTSs per second, such that the total floor requests seen in the channel at w equals $\lambda = \sum_1^{N+1} \lambda'$. The traffic generated at each node has its destination determined by a uniform distribution of the node's neighbors, i.e., for N neighbors, a node directs $1/N$ of its RTSs to each of those neighbors. This assumption makes our analysis independent of the specific routing choices made at each node. Again, each node is assumed to have at most one data block to send at any given time. A packet sent by any node propagates to all its neighbors (nodes in line of sight) in exactly τ seconds. All neighbor nodes are within line-of-sight and range of node w , but may be hidden from one another. The population of neighbors at node w is partitioned such that, for any given neighbor of w , there are Q neighbors that are hidden from it, and $N - 1 - Q$ neighbors that it can hear in addition to w . The rest of our assumptions are the same as in the two previous cases.

Assuming the above model, the following theorem gives the throughput for a given node.

Theorem 15 *The throughput of FAMA-NCS for a system with N independent neighbors, of which Q are hidden from the others is given by*

$$S = \frac{\delta \cdot e^{-\lambda'(N\tau + Q(2\gamma - \tau))}}{e^{-\lambda'(N\tau + Q(2\gamma - \tau))} [(3Q + N + 4)\tau + (Q - N)\gamma + (Q + 1)(\gamma + \delta)] + (N + 1) \cdot [T_{F_2} + (\gamma + \tau - T_{F_2}) \cdot e^{-\lambda'2Q\gamma + \frac{1}{\lambda}}] + N \cdot (\gamma + \gamma' + \delta 3\tau)}$$

Where T_{F_2} equals: $\left[\frac{e^{\lambda Q \gamma} - 1 - \lambda Q \gamma}{\lambda Q \gamma (1 - e^{-\lambda Q \gamma})} \right] + \gamma$ (4.20)

An RTS originated from any node A is successful at w if no other RTS from any other node in w 's neighborhood collides with A 's RTS. At node w and the $N - 1 - Q$ neighbors that hear A the vulnerability period of A 's RTS is τ seconds because these nodes

can detect carrier τ seconds after the beginning of the RTS. Accordingly, because of the independence of the nodes transmissions, an RTS is successful within the $N - 1 - Q$ nodes and w with probability

$$\prod_1^{N-Q} e^{-\lambda'\tau} \quad (4.21)$$

In contrast, the vulnerability period of an RTS with respect to the other Q hidden nodes is 2γ because these nodes cannot hear A 's transmissions. Accordingly, an RTS is successful with respect to any of the other Q nodes with probability $e^{-\lambda'(2\gamma)}$. Because all nodes are identical and independent, it follows that an RTS from a node A is successful at node w with the following probability:

$$P_S = \prod_1^{(N-Q)} e^{-\lambda'\tau} \prod_1^Q e^{-\lambda'(2\gamma)} = e^{-\lambda'(N\tau+Q(2\gamma-\tau))} \quad (4.22)$$

The probability that a given RTS arrival was generated by node w is $(\lambda'/\lambda) \equiv \frac{1}{N+1}$. The traffic directed to w comes from each of w 's N neighbors, who generate RTSs at a rate of λ'/s with $1/N$ destined for w . Accordingly, the probability that a given RTS arrival is meant for w is

$$\frac{\sum_{i=1}^N \left[\frac{1}{N} \cdot \lambda' \right]}{\lambda} = \lambda'/\lambda = \frac{1}{N+1} \quad (4.23)$$

A successful transmission period in the time line of node w includes the RTS with a one-way propagation delay, followed by the return CTS with a one-way propagation delay and lastly the data packet with a one-way propagation delay for a total of $T = \gamma + \gamma' + \delta + 3\tau$.

There are two types of failed transmission periods for RTSs. If only the nodes that can hear each other send RTSs in a transmission period, the average duration of the period in the time line of node w equals $T_{F_1} = \gamma + \bar{Y}$, where \bar{Y} is the same as in the fully-connected

network case [FG95b]. Noting that $\bar{Y} \leq \tau$, given that $\tau \ll \gamma$, we use the bound of Eq. 4.11 for T_{F_1} .

If one or more of the hidden nodes send RTSs in a failed transmission period, the failed transmission period consists of multiple overlapping RTSs with durations of γ seconds. Because these nodes are hidden and independent of one another, the length of the average failed transmission period in this case can be obtained by treating this case as an ALOHA channel with Q different nodes, in which a node A has a rate of λ' . To make use of prior results, we make the simplifying assumption that N and Q are very large. The aggregate RTS arrivals for these nodes is $\lambda_Q = \sum_1^Q \lambda'$. Substituting λ_Q for λ and γ for δ in the expression of \bar{F} derived by Takagi, et al. [TK85] we obtain

$$T_{F_2} = \left[\frac{e^{\lambda_Q \gamma} - 1 - \lambda_Q \gamma}{\lambda_Q \gamma (1 - e^{-\lambda_Q \gamma})} \right] + \gamma \quad (4.24)$$

The busy period can be partitioned into three categories – periods of successful transmissions at node w , periods of failed transmissions at node w , and periods when node w is deferred due to neighbors receiving data from nodes hidden from w .

At node w , successful transmission periods can consist of transmissions from w , transmissions to w , or successful transmission overheard by w . The probability that a given RTS is from w itself is λ'/λ , and the probability that it is successful is P_S . Therefore, the time allotted to successful transmission for w is $(\lambda'/\lambda) \cdot P_S \cdot T$. The probability of a transmission directed to w from a given neighbor is $\frac{1}{N}\lambda'$ and with probability P_S such a transmission is successful. Therefore, the total time for successful transmissions to w from its neighbors is $(\lambda'/\lambda) \cdot P_S \cdot T$.

We say that a transmission is overheard by node w if it is sent by a neighbor

of w , but is not meant for w . For any given neighbor, this is equal to the remaining number of transmissions not meant for w , i.e., $\frac{N-1}{N}\lambda$. The total overheard transmissions from all neighbors is then $((N-1)\lambda')/\lambda$. Therefore, the total time for overheard successful transmissions is $[((N-1)\lambda')/\lambda] \cdot P_S \cdot T$. Adding the time spent in successful transmissions for w and overheard successful transmissions together, we obtain that the total time for successful transmissions at node w is $P_S \cdot T$.

The average failed period consists of failed transmissions involving the neighbors of node w that are hidden from each other, and those nodes that are fully connected at w .

The probability that none of the hidden neighbors of node w are involved in a failed transmission at w is the probability that they have no arrivals whose transmission would interfere with the RTS at w , which is $e^{-\lambda'2Q\gamma}$. The probability that an RTS fails because of interference from one of the $(N-Q)$ fully connected nodes around w is $(1 - e^{-\lambda'(N-Q)\tau})$. Therefore, the probability of having a failed transmission period at node w involving only fully-connected neighbors of w is the probability that no hidden neighbor transmits within the duration of an RTS and some of the connected neighbors transmit within τ seconds of an RTS. Because nodes send RTSs independently of one another, this probability equals $(e^{-\lambda' \cdot 2Q\gamma} - e^{-\lambda'(N\tau + Q(2\gamma - \tau))})$.

Similarly, the probability of having a failed transmission period at node w involving one or more of the neighbor nodes hidden from one another is $(1 - e^{-\lambda' \cdot 2Q\gamma})$. It then follows that the average failed transmission period lasts

$$T_{FAIL} = (e^{-\lambda' \cdot 2Q\gamma} - e^{-\lambda'(N\tau + Q(2\gamma - \tau))}) \cdot T_{F_1} + (1 - e^{-\lambda' \cdot 2Q\gamma}) \cdot T_{F_2} \quad (4.25)$$

We say that node w is deferred when one of its neighbors is receiving a data

packet and after hearing noise on the channel (i.e., a failed period in which w was not directly involved). In both instances, w defers for a maximum packet transmission time, T . The probability that w hears noise from its neighbors' transmissions and defers is $((N \cdot \lambda')/\lambda) \cdot (1 - P_S)$. Each of w 's neighbors is identical to w and as such receives packets directed to it at a rate of λ' . Of that traffic sent by the nodes hidden from w , w can only hear the neighbor's CTS and will defer. As such, the probability that w is deferred by a neighbor receiving traffic not overheard by w is $((Q \cdot \lambda')/\lambda) \cdot P_S$. Therefore, the average time during which w is deferred by neighbor traffic (either noise or successful data) is

$$\begin{aligned} T_{DEFER} &= [((N \cdot \lambda')/\lambda) \cdot (1 - P_S) + ((Q \cdot \lambda')/\lambda) \cdot P_S] \cdot T \\ &= \left[\frac{N}{N+1} + \frac{Q-N}{N+1} \cdot P_S \right] \cdot T \end{aligned} \quad (4.26)$$

Accordingly, the average busy period lasts

$$\bar{B} = P_S \cdot T + T_{FAIL} + T_{DEFER} \quad (4.27)$$

Substituting Eqs. (4.25) and (4.26) in the above Eq., we obtain

$$\begin{aligned} \bar{B} &= \left(e^{-\lambda' \cdot 2Q\gamma} - e^{-\lambda'(N\tau + Q(2\gamma - \tau))} \right) \cdot T_{F_1} + (1 - e^{-\lambda' \cdot 2Q\gamma}) \cdot T_{F_2} \\ &\quad + T \cdot \left[\frac{N}{N+1} + \frac{Q+1}{N+1} \cdot P_S \right] \end{aligned} \quad (4.28)$$

The average idle period lasts 2τ seconds after every successful data packet transmission plus an average interarrival time of RTSs from all groups; therefore, we have

$$\bar{I} = \frac{1}{\lambda} + 2\tau \cdot P_S \quad (4.29)$$

The average utilization time at node w is simply the proportion of time in which useful data are sent during a successful busy period in w 's proximity, and

$$\bar{U} = \frac{1}{(N+1)} \cdot \delta \cdot P_S \quad (4.30)$$

Substituting Eqs. (4.28), (4.29) and (4.30) into Eq. (4.1) we obtain Eq. (4.20). \square

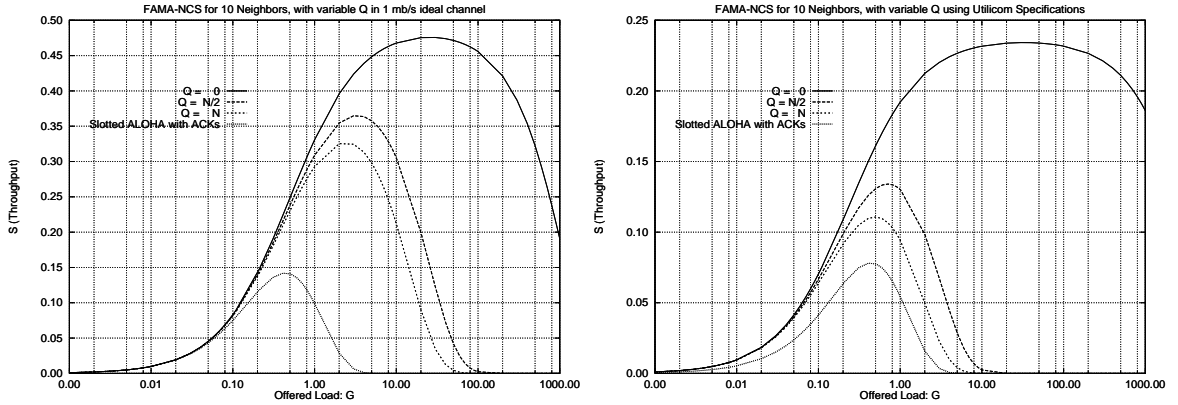


Figure 4.11: Throughput versus load for various Q in FAMA-NCS network

To compare the performance of FAMA-NCS in a multihop network, we use slotted ALOHA using non-priority acknowledgments. This choice is driven by the following considerations: (a) with hidden terminals, CSMA degrades to pure ALOHA; (b) implementing ALOHA with a single channel requires the use of acknowledgments to let the senders' of packets know if they need to retransmit; (c) implementing priority acknowledgment schemes (e.g., the schemes proposed for ALOHA and CSMA by Tobagi and Kleinrock [TK78]) do not work well with hidden terminals; and (d) slotted ALOHA has better performance than pure ALOHA.

Figure 4.11 shows the results of this comparison using the value of throughput derived by Tobagi, et al. [TK78] for slotted ALOHA with acknowledgments. We assume a network where each node has 10 neighbors for varying values of Q : 0, where the network is fully connected; $N/2$, where half the neighbors of any node are hidden from a neighbors' neighbors; and $(N-1)$, where all the nodes are hidden from their neighbors' neighborhoods. The latter case corresponds to a hypercube topology. Additionally, we assume a network

with a propagation delay of $6\mu\text{s}$ (one mile), 500-byte data packets, 25-byte RTS, 50-byte CTS. We show results for both a 1 Mb/s channel with zero preamble and processing overhead, and a 298 Kb/s channel with processing and preamble based on the specifications of the Utilicom model 2020 radio transceiver. The Utilicom radio has a turn-around time of about 5ms to ramp up the transmitter, and about the same to ramp down and be ready to receive again. At 298 Kb/s, the RTS itself is about 1 ms long; therefore, it takes about 11ms for a transmitter to send an RTS and be ready to receive a CTS. Together with its transmission overhead, a 500-byte data packet then becomes about 25ms in length. This makes the ratio of RTS to data about 0.47, which severely degrades the performance of FAMA protocols [FG95b].

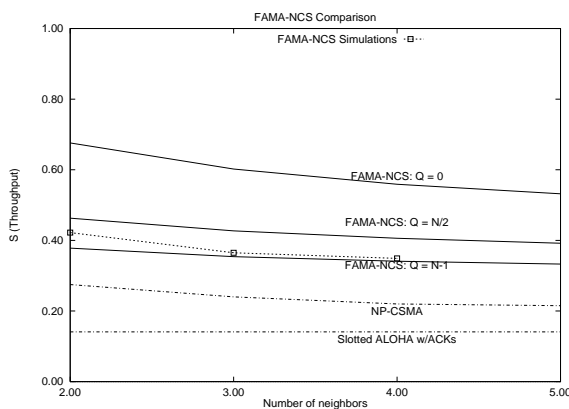


Figure 4.12: Throughput versus degree of node in ad-hoc network

Figure 4.12 shows the maximum throughput for FAMA-NCS in multihop networks versus the number of neighbors per node for the values of $Q(0, \frac{N}{2}$ and $N - 1$). The analysis assumes a 1Mb/s wireless network with a diameter of 1 mile (propagation delay of about $6\mu\text{s}$). The size of data packets RTSs used were 500 and 25 bytes, respectively. The figure also shows the throughput for slotted ALOHA with acknowledgments, which reflects the ex-

pected behavior of both ALOHA and CSMA protocols operating over a single channel when hidden terminals abound. We include the performance of non-persistent CSMA predicted by the model developed by Tobagi and Kleinrock when hidden terminals exist [TK75]; this value of throughput is only an upper bound on CSMA, because it assumes a separate ideal channel over which acknowledgments to packets are sent correctly in zero time.

The above results clearly show the importance of floor acquisition, which makes FAMA-NCS a far better choice than CSMA for a multihop setting. The results also indicate the important role that radio parameters play in the overall performance of an ad-hoc network. Without good carrier sensing and turn-around times, throughput degrades substantially.

4.4 Simulation Results

To validate our results on sufficient conditions for floor acquisition and the approximations made in our performance analysis of FAMA-NCS, we carried out a number of simulations³. The simulations ran the actual code used to implement the MAC protocols in embedded systems and, for the case of FAMA-NCS, this code is based on the specifications shown in Figure 4.2.

In the first set of experiments, we assumed single-channel spread spectrum radios capable of transmitting at 256 Kbs. The stations are within four miles of each other, giving a maximum propagation delay of approximately 20 microseconds. We present results for FAMA-NCS using single packet transmissions as well as packet trains. Figure 4.13 shows the various topologies used by these simulation experiments. Table 4.1 show the results for

³We thank Rooftop Communications Corp. for donating the C++ Protocol Toolkit (CPT) simulator.

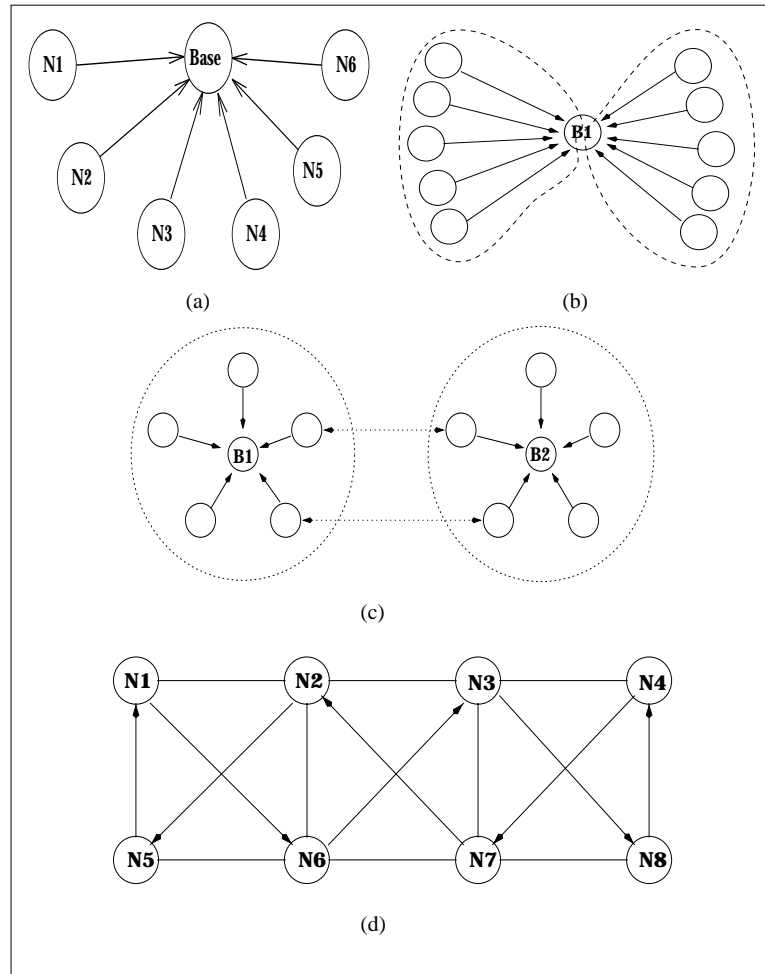


Figure 4.13: Simulation topologies used in testing FAMA-NCS protocols in hidden terminal environments

FAMA-NCS as compared to MACAW⁴ [BDSZ94].

To illustrate the importance of carrier sensing, we chose to compare FAMA-NCS against MACAW instead of FAMA-NPS, because MACAW uses packet-sensing and RTS-CTS handshakes and its performance has been reported before by Bhargavan, et al. [BDSZ94]. The physical parameters of the radio assumed a null transmit-to-receive turnaround time and transmitter ramp-up time, we also assumed transmission preamble

⁴We thank Ted Goodman for the use of his implementation of MACAW in CPT for our comparisons.

Configuration	<i>FAMA-NCS</i>	<i>FAMA-NCS train</i>	<i>MACAW</i>
(a)	.78	.89	.63
(b)	.58	.81	.49
(c) B1	.75	.88	.45
(c) B2	.75	.88	.39
(d) average	.49	.67	.06
(d) N1,4,5,8	.57	.81	.07
(d) N2,3,6,7	.42	.54	.05

Table 4.1: Throughput results for various configurations

and framing of 0 bits. These parameters were chosen in order to obtain the same results for MACAW that have been reported previously [BDSZ94]. Our results are only meant for comparative purposes.

In configuration (a) of Fig. 4.13 all stations are within range of all others (no hidden terminals). Traffic was generated at each node (N1 - N6) directed to the base station. Configuration (b) has two groups of five nodes that can hear the nodes in their own group, but are hidden from nodes in the other group. Traffic is generated from each node in each group directed to the central base station, *B1*. Configuration (c) has two base stations each with a group of five nodes sending traffic to it. The two groups cannot hear each other except for two nodes in each group that interfere with corresponding nodes in the other group (represented by the dashed arrows in the figure). Configuration (d) represents a multihop network of eight nodes. The lines between the nodes represent the radio connectivity of the network. The lines with arrows depict the flow of traffic from one node to another. Each node is generating a traffic stream to another node that at least three other nodes can hear, and is hidden from at least two of the other nodes in the network.

The traffic delivered to the nodes was sent at a constant rate with a packet size

of 512 bytes on the channel (including all headers and framing). The maximum capacity of the channel at this bandwidth and packet size is approximately 63 packets per second. Table 4.1 reports the maximum throughput achieved by each of the protocols.

FAMA-NCS achieves a higher throughput than that of MACAW in all cases. For the case of a fully connected network (configuration (a)), FAMA-NCS attains a maximum throughput of 78%, while MACAW achieves a 63% throughput. These results are as predicted by our approximate analysis of Section 4.3. For the case of MACAW, our simulation leads to a slotted behavior in which a slot lasts the duration of an RTS plus a maximum round-trip time. For the case of two independent groups competing for the same base station, FAMA-NCS has a maximum throughput of 58%, while MACAW's achieves 49% maximum throughput. However, for the case of the two base stations with a small number of interfering nodes (configuration (c)), FAMA-NCS achieves a throughput of nearly twice that of MACAW, and in fact shows very little loss in overall throughput from interference due to hidden terminals (78% without interference, 75% with interference).

In the multihop-network example (d) FAMA-NCS achieves an average throughput of 49%, with the nodes on the corners (N1,N4,N5,N8) reaching 57%, and the inside nodes reaching 42%. This is somewhat lower than predicted by our analysis of wireless LANs, and is expected because the analysis assumes a base station that does not transmit data packets. In this network MACAW achieves a much lower throughput of 6% on the average, achieving 7% at the corner nodes, and 5% on the inside nodes.

Additionally, fairness is not an issue in FAMA-NCS. Even the simple uniformly distributed backoff scheme gives all stations basically an equal share of the channel without the complex housekeeping suggested in MACAW [BDSZ94].

As expected, FAMA-NCS with packet trains of up to five packets in a train improves over single-packet transmissions by about 14% in the fully connected network and 17% for the two-base station configuration. In the case of two independent groups sending to one central base station, the improvement is almost 40%. For the multihop network FAMA-NCS packet trains provide an average throughput improvement of about 36%.

The poor performance of MACAW with hidden terminals is a direct consequence of the fact that data packets can collide with other packets, i.e., that it cannot enforce “floor acquisition” in the presence of hidden terminals and emphasizes the benefits of using carrier sensing.

In the second set of experiments, A 1Mb/s wireless network was modeled with stations at one mile from neighbors (propagation delay of approximately $6\mu s$). Data packet size was 500 bytes, and RTS and CTS were 25 and 48 bytes respectively. As Figure 4.12 illustrates, the simulation results are almost identical to the analytical results for the case in which $Q = (N - 1)$ (all neighbors hidden from each other), which validates the approximations used to make our model tractable.

Avg. Rate Pkts. Received	<i>FAMA-NCS</i> (2KB pkts)	<i>IEEE 802.11</i> (2KB pkts)	<i>MACAW</i> (1KB pkts)
Avg. Total Input	36.0KB/s	17.0KB/s	–
Avg. Local Input	15.3KB/s	8.4KB/s	1.1KB/s
Avg. at N1 & N4	15.5KB/s	5.5KB/s	2.1KB/s
Avg. for others	15.2KB/s	9.3KB/s	0.8KB/s

Table 4.2: Throughput comprison of FAMA-NCS, IEEE 802.11 and MACAW

In the third set of experiments, we assumed a 1 Mbps network with the same topology of Configuration (d) in Figure 4.13. However, traffic was only between $N1$ and its

neighbors, and between N_4 and its neighbors. Table 4.2 lists the results for FAMA-NCS, IEEE 802.11 DFWMAC, and MACAW. In the table, “total input” refers to traffic correctly received and meant for any node; “local input” refers to traffic correctly received and meant for the receiving node. The results illustrate that making the CTSs dominate the RTSs, i.e., enforcing floor acquisition, is important for throughput in ad-hoc networks.

4.5 Related Work

There are several prior proposals for single-channel MAC protocols similar to the FAMA protocols we have discussed. The IEEE 802.11 DFWMAC protocol [IEE97] uses an RTS-CTS exchange with non-persistent carrier sensing; however, it fails to provide correct floor acquisition with hidden terminals, because the duration of RTSs is actually longer than the CTS duration. Another difference with FAMA-NCS is that IEEE 802.11 does not support packet trains. The IEEE 802.11 DFWMAC protocol has been analyzed in the recent past [WSFW97, CG97]; our analysis complements this prior work by showing the impact of floor acquisition.

Lo [LM84] and Rom [Rom86] have proposed protocols similar to non-persistent CSMA that detect collisions by means of pauses. A station that senses the channel busy defers transmission, a transmitter that senses the channel idle starts transmitting but pauses during transmission and senses the channel. If the channel is sensed idle, the sender completes its transmission; otherwise, the sender continues to transmit for a minimum transmission duration (called the collision detection interval or CDI). Unfortunately, this protocol does not guarantee that a station can sense all collisions [Rom86].

Another CSMA-like protocol based on the idea of sending a request signal and pausing to sense collisions was proposed by Colvin [Col83] and analyzed by Brewster [BG87]. This protocol, however, was designed for LANs in which stations can sense the channel while transmitting.

A number of techniques have been proposed by Bharghavan, et al. [BDSZ94] to improve the performance of MACA [Kar90], which constitutes a variant of FAMA using RTS-CTS exchange and no carrier sensing. These techniques consist of different retransmission strategies and additional handshaking between sender and receiver. The resulting protocol is called MACAW. Like MACA, MACAW is based on the basic premise that collisions are detected not by sensing the channel, but by the receivers being able to understand the transmissions they receive. Given that we have assumed the minimum RTS-CTS handshake of MACA and full connectivity in our analysis, our results on FAMA-NPS provide an upper bound on MACAW's throughput.

4.6 Concluding Remarks

We have introduced the FAMA-NCS protocol for single-channel wireless networks with hidden terminals. FAMA-NCS permits a sender to acquire control of the channel in the vicinity of a receiver dynamically before transmitting data packets. The floor acquisition strategy uses an RTS-CTS handshake and is based on a few simple principles: (a) making the senders listen to the channel before transmitting RTSs; (b) implementing a busy-tone mechanism using a single channel and half-duplex radios by making the receiver send CTSs that last long enough for the hidden senders to realize that they must back off; and (c)

providing priority to those stations who successfully complete a handshake.

Although many MAC protocols have been introduced in the past based on RTS-CTS exchanges, we prove, for the first time, sufficient conditions under which an RTS-CTS dialogue becomes a floor acquisition strategy (i.e., one with which data packets are sent without ever colliding with other transmissions) with carrier sensing. Contrary to the conjectures made in prior work on MAC protocols based on collision avoidance [BDSZ94, Kar90], our verification and throughput analysis demonstrates that carrier sensing should be used in single channel networks because it substantially improves performance by enabling floor acquisition in the presence of hidden terminals.

We have shown through our analysis and supported by simulations that FAMA-NCS solves the hidden terminal problems of CSMA [TK75] in wireless LANS with hidden terminals and ad-hoc networks, because it is able to enforce floor acquisition. Our analysis illustrates the performance improvement obtained by allowing the transmission of packet trains in the clear, and a method to enable packet trains even with hidden terminals.

FAMA-NCS has been successfully implemented and demonstrated in actual packet radios for ad-hoc networks [WIN] built using commercial direct-sequence spread-spectrum radios and controllers.

Chapter 5

Collision Avoidance for Multichannel Networks

In this chapter we discuss collision avoidance techniques for networks with multiple channels. These channels can be derived using unique orthogonal codes (i.e., as in direct sequence spread spectrum (DSSS) or code division multiple access (CDMA)), or they can be established from a larger frequency domain that is split into several smaller non-overlapping subchannels within that domain (i.e., frequency domain multiple access (FDMA) and frequency hopping spread spectrum (FHSS)).

Multichannel networks exhibit several properties which differentiate them from single-channel networks. First, with orthogonal channels, multiple concurrent data packets can be successfully transmitted at a given instant. Secondly, delay characteristics (average delay as well as variance) have been shown to improve markedly with multiple channels [MN91, Leu90, Lof96]. Additionally, multiple channels gives the network better fault tol-

erance against noise and fading in the radio channel [Chu91, Lof96]. One caveat is that multiple channel networks can be inefficient because not all channels are fully used at all times.

In the past multichannel networks have been constructed using multiple transceivers operating on separate fixed channels [SK87]. As such, they were expensive to construct, and complicated to maintain and operate. Current transceiver technology has made radio devices available today that are inexpensive (on the order of \$100), medium speed (1Mb/s) and narrow band (1MHz) with upwards of 80 channels available. This hardware is also capable of being controlled by software to make channel changes quickly (on the order of $1\mu\text{s}$). This allows multichannel networks to be constructed inexpensively using a single device at each station.

Early work in protocol design for multichannel networks used CSMA or ALOHA protocols in slotted multiple channels [MR83]. A reservation protocol over multiple channels has been investigated for satellite communication systems [Leu90]. A sequential multichannel system using CSMA/CA on each channel to assign stations to channels dynamically has been presented by Brewster [BG89]. Analysis of multihop multichannel networks using CDMA in sparse networks with receiver-based, transmitter-based, pairwise-based, and common channel assignment is presented by Hu [Hu91].

In this chapter we present the FAMA-MC (multichannel) protocol that allows for concurrent data transmissions on orthogonal channels. FAMA-MC uses an RTS-CTS handshake on a common channel to establish the channel for data packet transmissions. The channel assignment can be either receiver-based or transmitter-based. In this discussion we limit ourselves to transmitter-based channel assignment. Section 5.1 describes in detail the

```

Variable Definitions
   $T_{PROP}$  = Maximum channel propagation delay
   $T_{RTS}$  = Transmission time of an RTS packet
   $T_{CTS}$  = Transmission time of a CTS packet
   $T_{DATA}$  = Transmission time of a DATA packet
   $T_{TR}$  = Time to transition from transmit to receive

Procedure START()
Begin
  Timer  $\leftarrow T_{RTS} + 2T_{PROP}$ 
  While(Timer not expired)
    wait
    call PASSIVE()
  End

Procedure PASSIVE()
Begin
  Change channel  $\leftarrow$  common channel
  While(No Local RTS Received  $\wedge$  No Local Packet to Send)
    wait
  If(Local RTS received) Then call SENDCTS()
  Else If(Local packet to send) Then call SENDRTS()
End

Procedure SENDRTS()
Begin
  Transmit RTS
  Change channel  $\leftarrow$  receivers data channel
  Timer  $\leftarrow T_{CTS} + T_{TR} + 2T_{PROP}$ 
  While(Timer not expired  $\wedge$  No Packet Received)
    wait
  If(Timer expired) Then call BACKOFF()
  Else DO CASE of (received packet type)
  Begin
    Local CTS: call XMIT()
    Default: call BACKOFF()
  End
End

Procedure SENDCTS()
Begin
  Change channel  $\leftarrow$  local data channel
  Transmit CTS
  Call REMOTE
End

Procedure BACKOFF()
Begin
  Change channel  $\leftarrow$  common channel
  Timer  $\leftarrow$  RANDOM(0,  $10 \times T_{DATA}$ )
  While(Timer not expired  $\wedge$  No Local RTS Received)
    wait
  If(Local RTS) Then call SENDCTS()
  Else call PASSIVE()
End

Procedure XMIT()
Begin
  Change channel  $\leftarrow$  local data channel
  While(Number of pkts for destination  $> 1$ )
  begin
    Set MORE flag
    Transmit Data Packet
  end
  Transmit Data Packet
  call PASSIVE()
End

Procedure REMOTE()
Begin
  Change channel  $\leftarrow$  senders data channel
  timer  $\leftarrow T_{DATA} + T_{TR} + 2T_{PROP}$ 
  While(Timer not expired  $\wedge$  No Packet Received)
    wait
  If(received packet  $\wedge$  MORE flag)
    Then call REMOTE()
  Change channel  $\leftarrow$  Common Channel
  If(Local packet) Then call BACKOFF()
  Else call PASSIVE()
End

```

Figure 5.1: FAMA-MC Specification

operation of FAMA-MC.

In section 5.2 we present the proof that FAMA-MC provides correct floor acquisition. The analysis of FAMA-MC in fully-connected networks is shown in section 5.3. Our analytical results show that FAMA-MC achieves average throughput of over 45% of the channel per station for carrier sensing devices, and over 40% per station with packet sensing devices.

Simulation results for FAMA-MC performance in both fully-connected and multi-hop networks are shown in section 5.4.

Section 5.5 presents our conclusions.

5.1 FAMA-MC Protocol

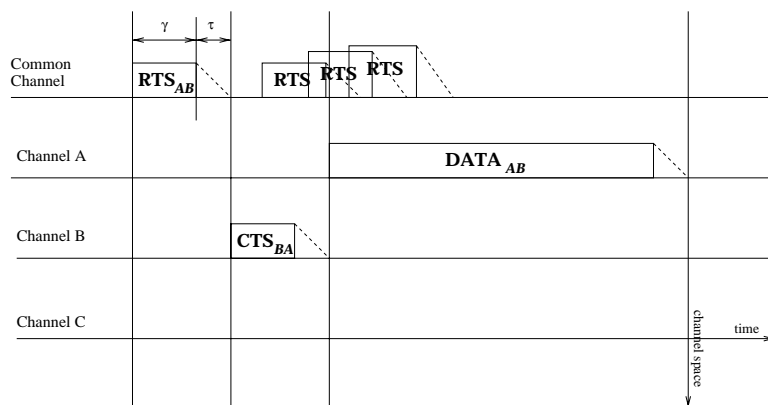


Figure 5.2: FAMA-MC transmission periods

FAMA-MC utilizes multiple channels in combination with an RTS-CTS handshake. FAMA-MC can operate with either carrier sensing or packet sensing devices. For this description we only discuss packet sensing devices.

A common channel is used to send RTS control packets only. Each station is required to have a channel unique from all other stations within a two hop radius to transmit data packets and CTS control packets to neighbors.

The RTS control packet contains the senders identification information as well as the identification of intended destination. The RTS also contains the data channel of the sender, which is used by the destination to receive data packets. The data channel is also used to send the CTS packets on, and stations must record this information about their neighbors to know the channel on which they will receive the CTS from this neighbor.

Figure 5.1 specifies FAMA-MC in detail. A station that has just been initialized starts by listening to the common channel for at least the maximum round-trip time across

the channel plus the time it takes to transmit an RTS. This allows any RTS in progress to have time to complete un-obstructed by the new station. After waiting the station transitions to the PASSIVE state. Stations in the PASSIVE state only listen to the common channel.

A station that is in the PASSIVE state and understands an RTS packet addressed to itself as the destination transitions to the SENDCTS state. On the other hand, if a station receives a packet to send in the PASSIVE state transmits an RTS on the common channel, transitions to the SENDRTS state and begins to listen to the destination's data channel. The sender waits long enough for the destination to send a complete CTS. If the CTS is not received within the required time, the sender transitions to the BACKOFF state and again listens to the common channel. Otherwise, upon receiving the CTS, the sender changes to its own data channel and transmits its data packet. After sending the data packet the sender changes to the common channel and transitions to the PASSIVE state.

In the SENDCTS state a station has just received an RTS packet addressed to itself on the common channel. It changes to its own data channel and sends a CTS packet. It then changes to the data channel of the sender as specified in the RTS. Once the CTS is transmitted and channel change complete, the station transitions to the REMOTE state to receive the data packet.

In the BACKOFF state, provided no RTS packet is received with the local station ID as destination, the station sends its RTS and transitions to the SENDRTS state. Otherwise, upon understanding an RTS packet for itself, the station transitions to the SENDCTS state.

A station in the REMOTE state waits on the senders data channel to receive a data

packet transmitted following the CTS just sent. Upon receiving the data packet the station transitions to either the BACKOFF state (if it has a packet waiting to be transmitted) or the PASSIVE state otherwise. Because it is possible for the CTS packet to be corrupted at the sender (i.e., due to channel fading) if, after waiting the maximum time for a data packet to be transmitted, the destination station does not receive the data, it transitions out of the REMOTE state.

If a sending station has multiple packets to send to a given destination it can set a MORE flag in the data packet header and the destination station will stay in the REMOTE state and continue to listen to the senders data channel instead of transitioning to another state after receiving the first data packet. The sender signals the last packet in the sequence by not setting the flag, and both the sender and destination transition after the final packet.

An example of FAMA-MC transmission periods is shown in Figure 5.2.

5.2 Correct Floor Acquisition in fully connected multichannel networks

To provide correct floor acquisition FAMA-MC must guarantee that data packets are sent free from collisions with any other packets, and within a finite time after becoming available for transmission.

Theorem 16 shows this under the following assumptions:

- A1) A packet sent over the channel that does not collide with other transmissions gets delivered free of errors to a station with probability $p > 0$.
- A2) A station transmits an RTS that does not collide with other transmission with prob-

ability $q > 0$.

- A3) All stations execute a FAMA protocol correctly.
- A4) The channel assigned to a given station is unique to any other station within a two hop radius.
- A5) A channel exists that is orthogonal to all other channels and is common to all stations.
- A5) The transmission time of an RTS or CTS packet is γ , the transmission time of a data packet is δ , and the processing time is t_p , where $\gamma \leq \delta < \infty$, and $t_p < \infty$.
- A6) The maximum end-to-end propagation time in the channel is $\tau < \infty$.

Theorem 16 *FAMA-MC provides correct floor acquisition in an ad-hoc network provided each station has a channel unique to itself within a two hop radius along with an orthogonal channel common to all stations.*

Proof:

Because RTS packets are only sent on the common channel, and the common channel is orthogonal to all of the data channels, no RTS packet can interfere with any data packet.

Similarly, CTS packets are only transmitted on a destination's own unique data channel. Because all data channels are unique (i.e., orthogonal) within a two hop radius, and CTS packets are only transmitted on the local data channel they cannot interfere with any data packets from or to other stations.

Finally, data packets are always transmitted on the senders own unique data channel, and as such can not interfere with any other stations data transmissions.

It then follows that all data packets are sent free from collision with other packets, and the theorem is true. \square .

5.3 Approximate Throughput Analysis

In our analysis of FAMA-MC we consider the case in which all stations have identical traffic loads, and equal numbers of neighbors. In addition, the destination for each packet sent from a station is chosen randomly from a uniform distribution of the station's immediate neighbors.

Because the destination for a packet is chosen randomly at time of transmission, the destination station may be on a data channel and not be able to hear the RTS packet sent on the common channel. As such, it is necessary to determine the average number of neighboring nodes available to receive a packet at any given instant in order to calculate the throughput of FAMA-MC. We use a Birth/Death Markov process to model the operation of the data channels, where each state k of the process represents the number of channels being used to transmit data packets at a particular point in time. Based on this model we find the expected value for k , $E[k]$, to use in our throughput analysis. In our model the value of k is bounded by 0 and $\lfloor \frac{N}{2} \rfloor$.

In our model we make a simplifying assumption that the arrival of RTS packets and the completion of data packets on the continuous time line can only occur as single events. This allows us to keep our analysis tractable, and we provide simulation results to validate our analysis. Figure 5.3 shows an example Markov process for a fully connected network of four nodes.

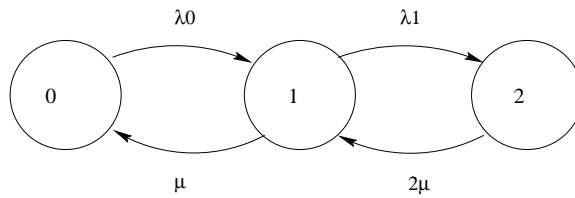


Figure 5.3: Markov Process for FAMA-MC Multichannel data channel behavior

In addition to the particular case we are considering in our analysis, we make the following assumptions:

- Finite population of nodes, N .
- Arrival of RTSs at each node is λ
- Each data packet is sent to a neighbor chosen from a uniform distribution of total neighbors
- The time to service (i.e., transmit on the channel) a data packet is exponentially distributed with an average length of δ
- Arrival of RTS packets and completion of data packets are unique events, and the Markov process may only advance or decline by at most one state on any given event.

We let k represent the value of the current state the process is in. In a fully connected network this implies that at state k , $2k$ stations are involved in data transmissions (either sending or receiving) on another channel, leaving $N - 2k$ stations operating in the common channel. As such, with no data channels busy, $k = 0$, and the maximum number of data channels that can be actively transferring at any give time is limited by the number of pairs of stations available, therefore $0 \leq k \leq \lfloor \frac{N}{2} \rfloor$.

The arrival rate of RTSs at any given state, k of the Markov process is given by,

$$\lambda_k = \lambda \cdot (N - 2k) \quad (5.1)$$

The probability of success for an RTS at any given state, k , in the Markov process is equal the probability that no other RTS arrives during the vulnerability period and the intended receiver is currently listening to the channel. For fully connected networks using carrier sensing the vulnerability period is the channel propagation delay, τ . Therefore

$$P_{S_k} = e^{-(\lambda_k)\tau} \cdot \left(\frac{(N-1) - 2k}{(N-1)} \right) \quad (5.2)$$

The average rate of successful RTS packets that will change the state of the process from $k \rightarrow k + 1$ is then

$$L(k) = \lambda_k \cdot P_{S_k} \quad (5.3)$$

Because we make the assumption that only one data packet may depart at any given instant, the average rate of departure for data packets at a given state k , μ_k equals $k \cdot 1/\delta$.

With the arrival and departure rate we define a ratio $r_k = L(k-1)/\mu_k$. Furthermore, we define

$$R_k = r_k \cdot r_{k-1} \cdots r_1 \quad (5.4)$$

$$R_0 = 1 \quad (5.5)$$

The probability of being in a particular state of the birth-death process at any given moment [Leo94] is then

$$P_k = \frac{R_k}{\sum_{i=0}^{\lfloor \frac{N}{2} \rfloor} R_i} \quad (5.6)$$

At steady state the expected value value of k is the average number of channels sending data at any given time

$$E[k] = \sum_{k=0}^{\lfloor \frac{N}{2} \rfloor} k \cdot P_k \quad (5.7)$$

Because traffic is uniformly distributed across all stations the average capacity each station achieves transmitting on it's channel equals $E[k]/N$. However, some of the transmission time is used for the CTS and one round-trip propagation delay, and the average throughput is therefore

$$S = \frac{E[k]}{N} \cdot \frac{\delta}{\delta + \gamma + 2\tau} \quad (5.8)$$

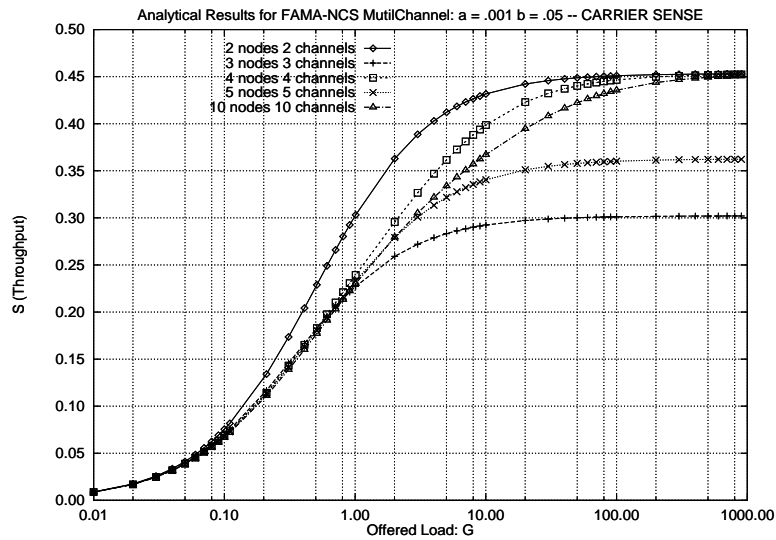


Figure 5.4: Analytical throughput per channel for FAMA Multichannel – Carrier Sensing

5.4 Simulation Results

Figures 5.4 and 5.5 shows the analytical results for fully connected networks of 2,3,4,5 and 10 stations. Figure 5.4 show results if carrier sense is used in the common

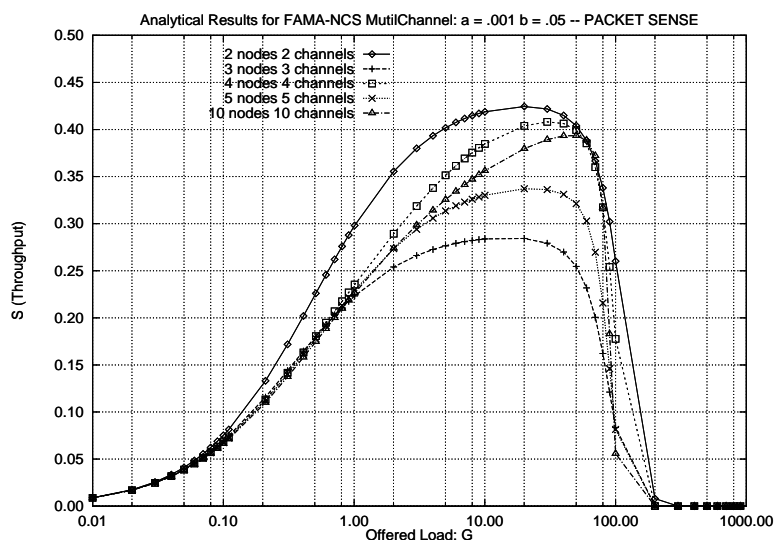


Figure 5.5: Analytical throughput per channel for FAMA Multichannel – Packet Sensing channel, and Figure 5.5 shows the use of packet sensing in the common channel. The propagation delay was equal to $.001\delta$, and the RTS and CTS size was equal to $.05\delta$.

A few key observations should be noted here. First, with uniform traffic assumed between all nodes, the highest throughput for transmitted traffic will not exceed 50% per node (because stations must communicate in pairs). As such, the graphs presented in Figures 5.4, 5.5, 5.6 and 5.8 only scale to 50% for throughput. Second with small numbers of nodes (less than 10) networks with an odd number of stations show lower capacities than those with even numbers of stations. This is because with an odd number of stations there is never a time when all stations are active on data channels at once and there will always be at least one lone node, which lowers the average capacity used by each station on its data channel. The extreme case of this is when there are exactly three nodes and the maximum average throughput per node is 33% instead of 50%.

Figure 5.6 compares the simulation results obtained for FAMA-MC and the an-

alytical calculations shown in Figure 5.5 (for packet sensing). The simulations modeled a 1Mb/s radio channel with a unique frequency for each station plus one additional 1Mb/s radio channel assigned as the common channel for all nodes. The network was fully connected, with a propagation delay of $6\mu s$ (≈ 1 mile). The simulator models a radio that does only packet sensing, no carrier sense was used. Note that the graphs confirm the anomaly that odd numbered small networks having lower capacities.

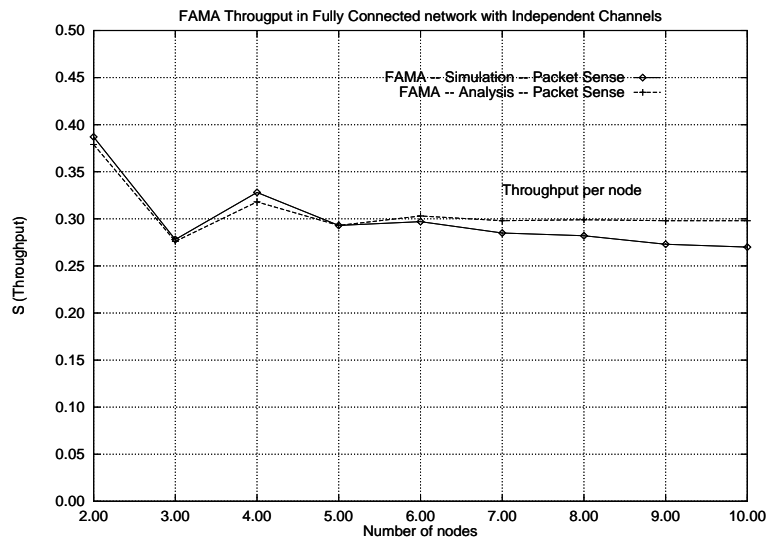


Figure 5.6: Comparison of Throughput for FAMA Multichannel simulations at maximum offered load

To compare FAMA-MC operation in ad-hoc networks we simulated two interfering networks with the topology shown in Figure 5.7. The results are compared to FAMA-NCS, IEEE 802.11, and MACAW in Table 5.1. In addition we simulated FAMA-MC operation in hypercube topologies and these results are shown in Figure 5.8 compared with a fully connected topology with the same number of neighbors.

Avg. Rate Pkts. Received	<i>FAMA-MC</i> (2KB pkts)	<i>FAMA-NCS</i> (2KB pkts)	<i>IEEE 802.11</i> (2KB pkts)	<i>MACAW</i> (1KB pkts)
Avg. Total Input	---	36.0KB/s	17.0KB/s	—
Avg. Local Input	27.1KB/s	15.3KB/s	8.4KB/s	1.1KB/s
Avg. at N1 & N4	72.4KB/s	15.5KB/s	5.5KB/s	2.1KB/s
Avg. for others	12.0KB/s	15.2KB/s	9.3KB/s	0.8KB/s

Table 5.1: Results for FAMA-NCS in an ad-hoc network

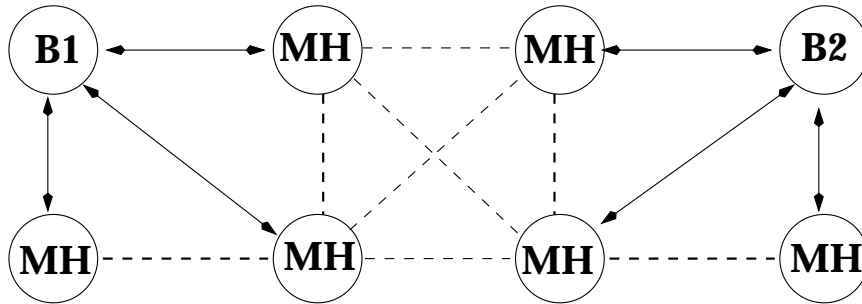


Figure 5.7: Multihop topology for FAMA-MC simulations

5.5 Conclusion

We have presented a new protocol that provides floor acquisition in multichannel networks called FAMA-MC. FAMA-MC achieves a channel utilization of over 45% per node for carrier sensing and over 40% utilization per node for packet sensing devices.

We give simulation results to validate our analysis. The results show that the throughput in multi-hop networks with hidden terminals

We have shown that FAMA-MC preserves the floor acquisition property, and that no data packets can collide with any other packets, as long as stations have data channels that are unique within a two-hop neighborhood.

FAMA-MC has been implemented in an operational ad-hoc packet radio network

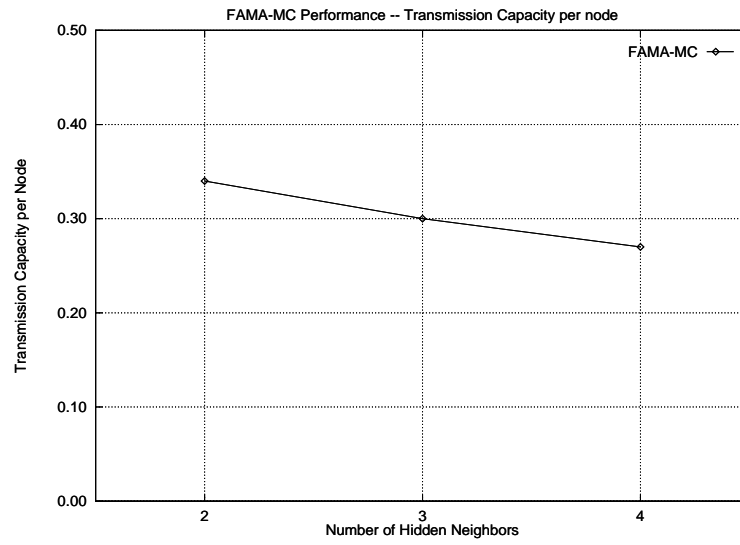


Figure 5.8: Comparison of Throughput for FAMA-MC in multihop topology

using commercially available 1Mb/s slow frequency hopping radios.

Chapter 6

Implementation: WINGs for Internets

Multihop packet-radio networks (or ad-hoc networks) are an ideal technology to establish an “instant communication infrastructure” for military and civilian applications (e.g., ad-hoc networks for disaster areas resulting from flood, earthquake, hurricane, or fire) in which both hosts and routers are mobile and can have multiple points of attachment to the global IP Internet. In multihop packet-radio networks, there are no dedicated base stations as in commercial cellular networks, and all nodes interact as peers for packet forwarding. This distributed nature eliminates single points of failure and makes packet-radio networks much more robust and survivable than commercial cellular networks. Furthermore, because packet-radio networks can be entirely deployed and operated by the end-users, there is no reliance on a wireless service-provider or a stable backbone infrastructure.

The DARPA packet radio and SURAN programs [sur90, pr-89] demonstrated the

basic capabilities of ad-hoc networking. However, the ad-hoc networks proposed and implemented to date [LNT87, ABB⁺96] have been designed as opaque subnetworks using an intranet protocol for packet forwarding that enables packets to flow from one packet radio to the other and from one entry point of the ad-hoc network to an exit point. When the ad-hoc network is used as a subnet in an IP internet, one or more of the packet radios connect to the rest of the IP internet through IP routers in order to provide end-to-end connectivity. IP packets are encapsulated in intranet-level packets, and the routing functions within the ad-hoc network are carried out below the IP routing layer.

Over the past two years, the Wireless Internet Gateways project (WINGS) has introduced and demonstrated an architecture and protocols for *mobile wireless internet-working*, in which packet-radio nodes are wireless IP routers and the global IP Internet is extended to the mobile wireless environment in a seamless manner. Within the WINGS project, Wireless Internet Gateway (WING) prototypes were built to demonstrate the concept, architecture, and protocols for wireless mobile internetworking. A novel feature of the WINGs is that the same protocol code used to debug and analyze new protocols within a Unix simulation environment is also used to control the operation of the actual WING prototypes. The WINGS project is part of the DARPA Global Mobile (GloMo) Information Systems program [glo].

WINGs are wireless IP routers designed to extend the global IP Internet to ad-hoc networking environments. Like an IP router, a WING accomplishes its routing functions at the IP layer; however, in contrast to wired IP routers, WINGs must also adapt to the dynamics of an ad-hoc network in which routers can move frequently, and must schedule their transmissions to maximize utilization of the available spectrum, while avoiding interference

with other transmissions that they may not even be able to detect (the *hidden terminal* problem).

Section 6.1 describes our protocol architecture to support wireless mobile internet-working using WINGs. Section 6.2 describes the FAMA-NCS protocol (for floor acquisition multiple access with non-persistent carrier sensing), which eliminates the hidden-terminal problems of CSMA in single-channel networks [FG97b]. Section 6.3 describes the wireless internet routing protocol (WIRP), which supports internet routing in the wireless mobile environment. Section 6.4 presents the results of a number of simulation experiments designed to show the performance of the entire WING protocol stack. Section 6.5 describes the software and hardware configuration used to build the WING-I prototype, as well as field demonstrations of ad-hoc networks based on WINGs.

6.1 WING Protocol Architecture

Figure 1 shows a high-level description of the WING protocol architecture that includes only the main protocols implemented for the WINGs when they operate over a single channel. The key differences between a WING and a traditional router are that: (a) we have improved upon traditional internet routing protocols like RIP and RIPv2 with WIRP, which can far more effectively handle the topological dynamics and broadcast radio channel of the wireless links; (b) the routing protocol interacts with the link-layer protocols in order to reduce control traffic needed to maintain routing tables; (c) we use a new set of protocols for link control and channel access designed for ad-hoc networks with hidden terminals.

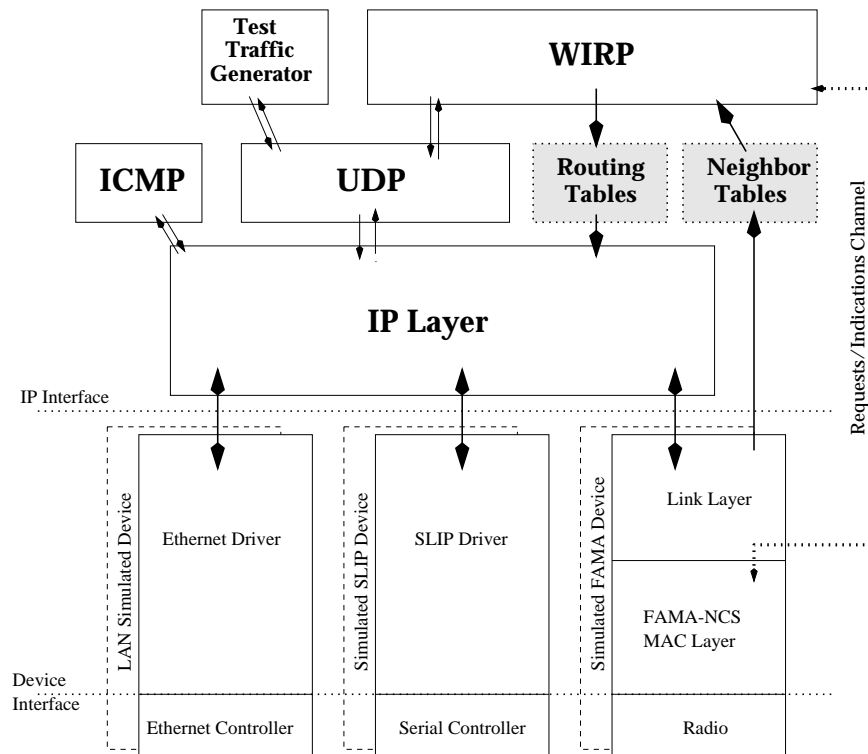


Figure 6.1: WINGS I Protocol Architecture

An internal traffic generator (TG), which uses the User Datagram Protocol (UDP), is part of the basic architecture and is used extensively in our simulations and testing of WING prototypes.

The Internet Protocol (IP) uses a standard set of interface functions to access the routing table and to obtain routing instructions for packets being forwarded. The IP protocol's interface to the table is the same regardless of what network protocol is used to update the routing table. Similarly, all protocol modules that are connected to the bottom of the IP modules present the same standard IP interface (IpIf) to allow new protocol interface stacks to be easily added or swapped for existing ones.

The WING currently supports three interface protocol stacks for interfaces to an

Ethernet LAN, a SLIP link, and a digital radio device. The FAMA-NCS protocol and a radio link-layer protocol are used to control the underlying radio device. An Ethernet protocol module which includes the Internet standard Address Resolution Protocol (ARP) is used to control the Ethernet device. A simple SLIP protocol module is used to control the underlying serial communications device. A common device applications programmer interface (API) provides a consistent interface structure between the control protocols and each of these interface devices. This API divides the protocol-to-device interface into three fundamental types of primitives: commands, variables, and signals. In addition, this Device API allows the developer to swap an actual interface device driver for one that simulates the communication channel with no changes required of the interface control protocols. For instance, unbeknownst to the MAC and logical link control protocols, the device driver for the radio used in the WING prototype (the Utilicom LongRanger radio) can be swapped for a module that simulates the radio channel in a simulation environment.

Because of its particular importance for developing open-architecture wireless internetwork systems, special attention was given to the definition of the interface between the protocol software and the digital radio modem. This has resulted in the emergence of a pair of standard interface specifications: the *Radio Device API* [BFLL96] and the *Physical Radio Interface* [Bey97]. The Radio Device API defines the software interface between the MAC-layer protocols and the “transceiver frame controller” which converts a packet buffers to/from a synchronous bit stream. The Physical Radio Interface defines the lower-layer interface between this transceiver frame controller and the digital radio modem, and consists of a synchronous serial “Data Port” and an abstract “Command Port.” The Command Port includes a set of variables, commands, and signals, most of which are also made available

to the protocols over the Radio Device API, for controlling and accessing the frequency, RSSI, transmit power, receiver carrier state, and others. The intent of these Radio APIs is to facilitate both collaboration and independent development of the network protocols and digital radio modem hardware which can be easily mixed and matched into well-integrated systems.

6.2 FAMA-NCS

FAMA-NCS is similar to the protocol use in the IEEE 802.11 standard [IEE97]. However, this and prior protocols based on handshakes (also called collision avoidance) and carrier sensing or packet sensing do not provide floor acquisition in networks with hidden terminals [FG97b].

A station that has just been initialized must wait the time it takes to transmit the maximum-size data packet in the channel plus one maximum round-trip time across the channel. This allows any neighboring station involved in the process of receiving a data packet to complete the reception un-obstructed. The initialization time also gives the station the ability to learn of any local traffic in progress. If no carrier is detected during the initialization period, the station transitions to the `PASSIVE` state. Otherwise, it transitions to the `REMOTE` state. A station can only be in the `PASSIVE` state if it is properly initialized (i.e., has no packet to send, and senses an idle channel). In all other states, the station must have listened to the channel for a time period that is sufficiently long for any neighbor involved in receiving data to have finished.

A station that is in the `PASSIVE` state and senses carrier transitions to the `RE-`

MOTE state. On the other hand, a station that receives a packet to send in the PASSIVE state transmits a request-to-send (RTS) and transitions to the RTS state. The sending station waits long enough for the destination to send the clear-to-send (CTS) to the RTS. If the CTS is not received within the time allowed, the sender transitions to the BACKOFF state. If the sender hears noise on the channel after its RTS, it assumes a collision with a neighbor's dominating CTS and waits long enough for a maximum-length data packet to be received. Otherwise, upon receiving the CTS, the sender transmits its data packet. Because the CTS could be corrupted at the sender, once the destination station sends its CTS, it only needs to wait one maximum round-trip time to sense the beginning of the data packet from the source. If the data packet does not begin, the destination transitions either to the BACKOFF state (if it has traffic pending) or to the PASSIVE state.

In the BACKOFF state, if no carrier is detected during the entire backoff waiting period computed by the station, the station transmits an RTS and transitions to the RTS state as before. Otherwise, upon sensing carrier the station transitions to the REMOTE state. Any passive station that detects carrier transitions to the REMOTE state, and waiting periods are enforced after the channel clears based on what the station last heard (noise, a control packet, a data packet). Such waiting periods allow RTS/CTS exchanges and packet trains to terminate [FG97b].

The channel becomes idle when all stations are in either the PASSIVE or BACK-OFF state. The next access to the channel is driven by the arrival of new packets to the network and retransmission of packets that have been backed off.

The length of a CTS is larger than the aggregate of the length of an RTS plus one maximum round trip time across the channel, the transmit to receive turn around

time, and any processing time; the length of an RTS is larger than the maximum channel propagation delay plus the transmit-to-receive turn around time and any processing time. This is required to avoid one station hearing a complete RTS before another has started to receive it. The relationship of the size of the CTS to the RTS gives the CTS *dominance* over the RTS in the channel. Once a station has begun transmission of a CTS, any other station within range of it that transmits an RTS within one propagation delay of the beginning of the CTS hears at least a portion of the dominating CTS and backs off, thereby letting the data packet that will follow to arrive free from collision. The dominating CTS of FAMA-NCS plays the role of a busy tone sent in the same channel as data packets.

To increase the efficiency of the channel, a station that has successfully acquired the floor can dynamically send multiple packets together in a train, bounded by an upper limit. The signaling required to support packet trains with hidden terminals has been previously discussed [FG97b].

6.3 WIRP

The Wireless Internet Routing Protocol (WIRP) was designed for an IP internet in which topology changes are the rule, rather than the exception, and where control traffic must be limited. It runs on top of UDP and it can be functionally divided into three main components: Reliable exchange of updates, neighbor discovery mechanism, and its path-finding routing algorithm (PFA).

6.3.1 Reliable Transmission of Updates

Reliable transmission of update messages is implemented by means of multicasting of update messages that are acknowledged with update messages carrying both updates and acknowledgments to one or more other messages.

After receiving an update message free of errors, a node is required to acknowledge, which indicates that there is good connectivity with the neighbor and that the neighbor has processed the update message.

An update message is retransmitted if acknowledgments are missing after a finite time and specifies which neighbors should acknowledge. A WING keeps a Message Retransmission List (MRL) with the neighbors whose acknowledgments are still missing [MG96].

6.3.2 Neighbor Discovery Mechanism

Every WING checks the connectivity with its neighbors periodically. A WING transmits a HELLO packet if it does not have any data packet or routing-table update message to transmit during a HELLO interval. In the current implementation, the HELLO interval is set to 3 seconds.

To interoperate, WIRP and FAMA-NCS share a *Neighbor Information Table* (NIT) and a *Subnet Activity Table* (SAT). The NIT table contains an entry for each neighbor with a flag and a counter. FAMA-NCS sets the flag for a particular neighbor every time it hears a packet (control or data) with that neighbor as the source station. WIRP periodically scans the table to increment the counters and reset the neighbor flags to 0. The SAT table contains an entry for each subnet attached to the FAMA-NCS domain with a flag. FAMA-

NCS sets the flag every time it sends a data packet to a particular subnet. WIRP also periodically scans this table and resets the flags.

In addition to these tables, a message channel is used to send requests and indications between WIRP and FAMA-NCS. When FAMA-NCS cannot successfully send a packet to any given destination (i.e., no CTS response is received after several RTS transmission to the destination) an indication is sent to WIRP informing it that a packet was dropped for the destination. WIRP can also send requests to FAMA-NCS. WIRP can tell FAMA-NCS which *proxy* address to use for broadcast packets at any given time, and to send *explicit* HELLOs when WIRP deems it necessary. FAMA-NCS sends explicit HELLOs by sending an RTS with a special destination address (different from the proxy for broadcast address) which no station will respond to directly, but will still send the source address up to its own WIRP layer as an implicit HELLO simply by having heard the control packet.

6.3.3 Wireless Internet Routing

The basic design concept in WIRP is simple. Each WING communicates to its neighbors a hierarchical routing tree in an incremental fashion. The hierarchical routing tree reported by a WING consists of all the WING's preferred shortest paths to each known IP network and IP host, where an IP host is typically a WING. An entire remote IP network is simply a node in the routing tree.

WINGs exchange their hierarchical routing trees incrementally by communicating only the distance and second-to-last hop (predecessor) to each destination. In the case of destinations within a WING's own IP network, the second-to-last hop consists of a WING (i.e., a host-level IP Address). In the case of a remote IP network known to the WING,

the predecessor consists of another IP network. Hence, internet routing in WIRP does not require a WING to store more routing-table entries than an Internet routing protocol like RIPv2 would, for example.

In essence, WIRP implements Dijkstra's shortest-path algorithm distributed over a hierarchical graph representing the connectivity of IP networks as well as the connectivity of the WING's own IP network(s). The algorithm used for this purpose is a modification of the path-finding algorithm (PFA) [MG96].

The entry for destination j in WING i 's routing table consists of the destination's IP address, the distance to the destination (D_j^i), the successor (s_j^i), and the predecessor (p_j^i) along the preferred path (shortest path) to the destination. Routing information is exchanged among neighboring WINGs by means of update messages. An update message from WING i consists of a vector of entries reporting incremental updates of its routing table; each entry specifies a destination j (i.e., an individual host or an IP network), the reported distance to that destination, and the reported predecessor (individual host address or an IP network) in the path to the destination.

Because every WING reports to its neighbors the second-to-last hop in the shortest path to the destination, the complete path to any destination (called the implicit path to the destination) is known by the WING's neighbors. This is done by a path traversal on the predecessor entries reported by the WING. This accounts for the elimination of counting to infinity problems in WIRP that plague RIP.

6.4 Simulation Experiments

The average throughput of FAMA-NCS and the effectiveness of WIRP in providing new paths after topology changes were analyzed by simulation using the C++ Protocol Toolkit (CPT) [BN95] on a Sun Ultra II Sparc workstation.

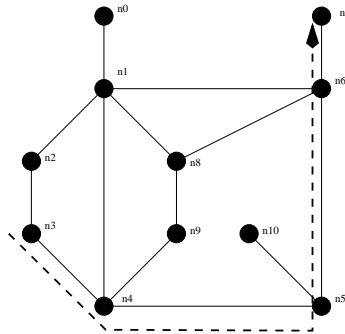


Figure 6.2: WIRP routes established during simulation at startup

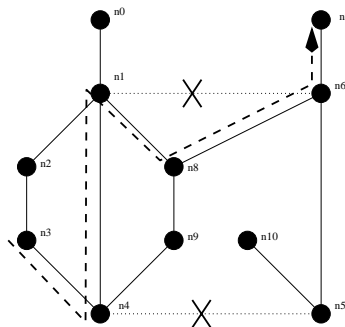


Figure 6.3: WIRP routes established during simulation after links are lost

Figure 6.2 shows the “Los Nettos” network topology used in the simulations; the average degree of nodes in this topology is approximately three. We used two different types of channels for our simulations. In the first case, the nodes were capable of a maximum transmission rate of 1Mb/s and a zero transmit to receive turnaround time, with no pream-

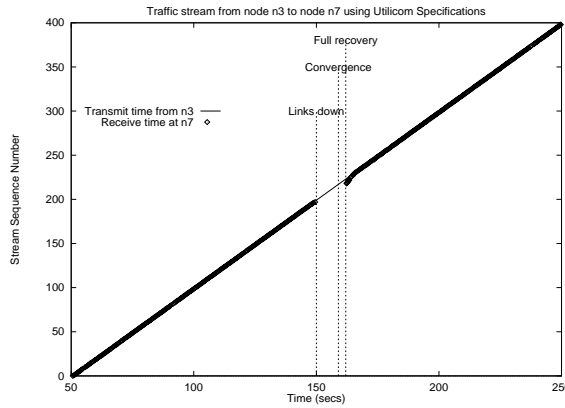


Figure 6.4: Sequence ID versus time for stream from n3 to n7 for Utilicom radios

ble or processing time included. In the second case, we simulated the parameters of the Utilicom model 2020 radio device, which is the current platform of the WING I prototype. The Utilicom radio introduces a 5ms transmit ramp up time and a 5ms ramp down time; this includes a 745-bit preamble (for capture) and a 3ms capture release delay. Finally, the Utilicom radios do not provide true carrier sensing of the channel, and provide only a capture detection signal. As such, these devices cannot *hear* noise and signal the MAC layer of activity on the channel (i.e., a dominant CTS, or collisions by other nodes).

Nodes were separated by a distance of approximately one mile from each other, giving a propagation delay of about $6\mu\text{s}$. In addition, each node had a single 20-packet output buffer at the MAC layer for all data packets, and a separate queue for control packets. FAMA-NCS attempted 10 transmissions to deliver a packet to the radio channel before giving up.

To test the convergence capabilities of WIRP, a single stream was initiated between two nodes on opposite sides of the network. The nodes were started with empty routing tables and allowed to find each other and stabilize for 50 seconds of simulation time. A

UDP traffic stream was then started using the test traffic generator (TG) from Node 3 to Node 7 sending packets of 500 bytes at a rate of two packets per second. After the stream had been flowing for 100 seconds the links between Node 1 and Node 6 and between Node 4 and Node 5 were blocked (as in a long period of fading, or obstruction in the path). The simulation was allowed to run another 100 seconds.

Figures 6.2 and 6.3 show the route established through WIRP before and after the links were blocked. Figure 6.4 shows the arrival of packets at the receiver by sequence number versus time for a network using the Utilicom parameters. Also shown are the point where the links were broken in the topology, the point where Node 1 converged to the new route, and the point of full recovery when the stream was again delivered to the destination. The time for Node 1 to converge with a new route for the stream to Node 7 was 9.1 seconds, full recovery of the stream at Node 7 took an additional 3.2 seconds for a total of 12.3 seconds, with a loss of 21 consecutive packets out of the stream.

To verify attainable throughput, we ran simulations using both the 1Mb/s and 298Kb/s channels with 500 byte data packets and 25 byte RTS. A set of traffic streams (from the TG) was started from edge nodes n0, n7, and n10, and inside nodes n2, n3 and n9. Each set consisted of a stream from the node to each of the other nodes in the network, and an echo back of the test packet. Nodes n1, n4, n5, n6 and n8 did not originate any streams (however, they did echo test packets as well as forward traffic to others).

Figures 6.5 and 6.6 shows the average throughput per node over one second intervals. Figure 6.7 shows the average delay for all packets received during a given interval (one second). The delay for the 298Kb/s channel is an order of magnitude greater than that for the 1Mb/s channel; as the throughput seen for the 1Mb/s channel is an order of magnitude

greater than that of the 298Kb/s channel, this is an expected result. These simulation results agree with the performance predicted by the analytical model of FAMA-NCS with hidden terminals [FG97b].

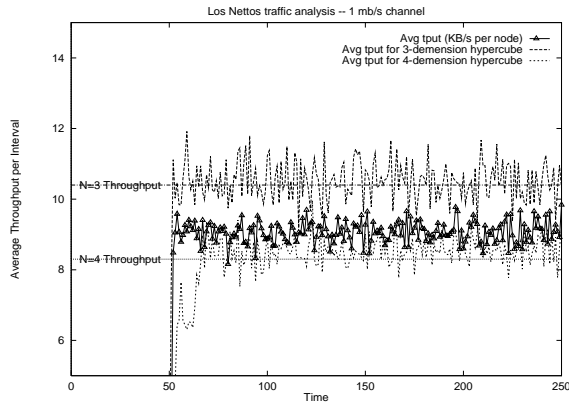


Figure 6.5: Average throughput per node in the Los Nettos topology with ideal radios

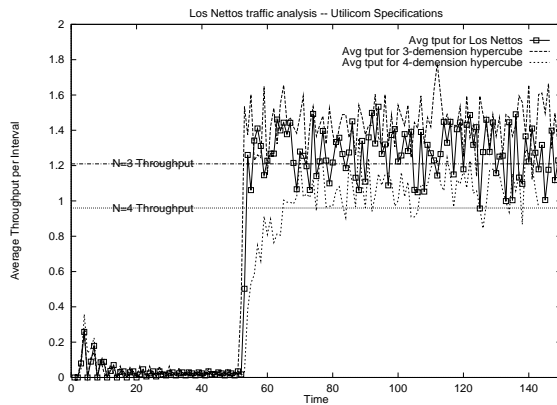


Figure 6.6: Average throughput per node in the Los Nettos topology with Utilicom radios

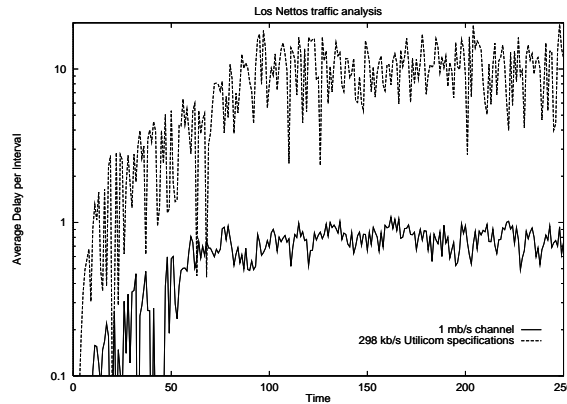


Figure 6.7: Average delay per node in the Los Nettos throughput tests

6.5 Implementation Experience

6.5.1 WING Software Approach

A well-defined, object-oriented framework for linking protocol modules was created within the project to allow the individual protocols to be coded and tested independently by multiple developers, and then easily integrated, swapped, or added into complete protocol stacks. This framework consisted of C++, Application Programming Interfaces (APIs) defined at key protocol boundaries and tables in the WINGS protocol stack. The design of this framework was facilitated by the use of the core protocol library objects available in the C++ Protocol Toolkit (CPT) [BN95] discussed below.

Development of the WINGs protocols has been facilitated through the use of CPT. CPT was created to support the efficient development, testing, and analysis of protocol software within a realistic simulation environment, and then allow the seamless transition of this same protocol software into an embedded hardware system. This support for seamless transition of the protocols into an embedded system is in stark contrast to the traditional

two-phase approach where protocols are first developed and tested on simulation systems, and then re-implemented for a target embedded system. In particular, the development of the WINGs protocols benefited from the following key capabilities of the CPT:

- **Rapid and Reliable Transition to Embedded Systems.** Minimal, well-defined interfaces to device modules and the hardware system platform, allow protocol software to be transitioned from a simulation environment to an embedded system simply by recompiling and relinking with a new platform wrapper and device drivers libraries.
- **The CPT Protocol Framework.** The object-oriented, highly-instrumented, and robust CPT Protocol Framework library speeds development of network protocols by presenting the developer with standard, protocol-relevant objects such as packets, queues, timers, protocol modules, and state machines. Also, this framework provides a consistent protocol structure to permit the mixing-and-matching of protocol stacks.
- **Realistic Simulations.** A realistic simulation capability, particularly well-suited for wireless networks, allows the performance and reliability of the network protocols to be extensively tested in a highly-instrumented simulation environment prior to field testing.
- **Public-Domain Graphical Analysis Tools.** The performance and behavior of CPT simulations and operational networks can be analyzed using a suite of public-domain filtering and graphic visualization tools including the NetViz network animation tool [Bey].

6.5.2 WING Hardware Configuration

The hardware platform for the WING prototypes are based on a Motorola, 68360-based controllers for running the protocols and supplying the serial communication channels for communicating with digital radio modems (over the Physical Radio Interface [Bey97]). The WING I prototype uses a 298-Kbps, direct-sequence spreading radio from Utilicom Inc. Table I provides the specifications for the WING I prototype.

However, it is important to note that, due to the flexibility of the 68360's communication capabilities and the growing acceptance of the Radio API specifications, the WING controller can be used in conjunction with a variety of other radios. For example, during the WINGS project and related efforts, the WING controller has been effectively integrated with two other commercial radios (one being a 1-Mbps, frequency-hopping radio by Netwave), and plans are currently being made to integrate the WING controller with radios being developed as part of the GloMo Program by UCLA, Virginia Tech, and ISI.

6.5.3 Wireless Internetworking Demonstrations

The CPT simulator was incorporated into the WINGs from its inception in November 1995. The baseline protocols were completed and installed on the first embedded system in May, 1996. In July, 1996 a WING ad-hoc network was demonstrated to the GloMo community at the CalNeva Lodge in Lake Tahoe, California. One WING was connected through a SLIP link to a local ISP, and three more were setup though the lodge to form a three-hop network connecting to a laptop running WWW sessions. In a second demonstration a satellite feed from Hughes Research Labs (HRL) was sent over a WaveLan link to a commercial router connected to a WING router and to the laptop via a single-hop WING network.

Protocol Processor	33 MHz Motorola 68360
Memory	4 MByte RAM, 1 MByte Flash ROM
Wired Interfaces	Host Port: SLIP RS-232, 57.6 Kbps max rate LAN Port: 10 Mbps 10BaseT Ethernet Console: RS-232, 115 Kbps max bit rate
RF Frequency Range	905 to 925 MHz center frequency software selectable by 100 kHz increments.
RF Modulation Type	QPSK direct sequence
RF Output Power	800 mW (29 dBm) maximum software controlled for lower power settings.
RF Radiated Power	4 Watts (36 dBm) at maximum output power with a 7 dBi-gain antenna, neglecting cable loss.
Receiver Sensitivity	-92dBm at 10^{-6} bit error rate (BER) (at the code length and modulation used by the WING I)
Approximate Link Range	7 miles multipoint-to-multipoint max. 15 miles point-to-multipoint max. 30 miles point-to-point max.
PN Code Rate	4.6 Mchip/second
PN Code Length	31 chips/symbol 2 bits/symbol (15.5 chips/bit)
Channel Bit Rate	298 Kbps
Power Requirements	12 VDC at 1.1 Amps (11 Watts), receiving 1.25 Amps (15 Watts) transmitting
Dimensions	Controller: 7.25" W x 1.5" H x 6.5" D Radio: 4.125" W x 1.5" H x 6.5" D

Table 6.1: WING I Hardware Specifications

The WIRP and FAMA protocols were installed and operational on the WINGs in November 1996. In February 1997, these WINGs were demonstrated at the GloMo PI meeting. The network configuration consisted of a hub connected to the UCLA campus network. One WING was connected to the hub and served as the border router for the rest of the WING and their respective clients. Two additional WINGs, each with a FreeBSD client attached to the Ethernet port, were operational in the network. Three internetworking demonstrations were accomplished. A video stream was sent between the two WING clients

running FreeBSD and using the VIC Mbone tool over the WING link. Rates of eight to ten frames per second were shown. HRL again provided a satellite video feed as in the Tahoe demonstration, this time to the local subnet. A live video transmission was received and shown also at eight to ten frames per second. The WING router was instantiated in the UCLA routers to the DARTNET connection, and clients on the WING subnet were able to access and download files across DARTNET (i.e, clients were able to connect to SRI International's HTTP server to download files from it).

6.6 Conclusions

We have presented the architecture, main protocols, and implementation of Wireless Internet Gateways (WINGS), wireless IP routers designed to provide wireless mobile internetworking over ad-hoc networks.

The WINGs and the basic concept of achieving mobile wireless internetworking have been demonstrated successfully in the DARPA GloMo program, and our work continues to analyze improvements on the initial protocols being used in the WINGs today. In particular, analyzing the performance of different types of routing and channel access protocols capable of using multiple channels as well as applying intelligent control of other link characteristics is an attractive area of research.

We have shown that using the FAMA-NCS protocol, a given station and its neighbors are able to utilize at least one third of the channel capacity in the worst case (with all neighbors hidden from each other). This is in remarkable contrast with CSMA, whose behavior degrades to the basic ALOHA protocol under hidden terminals, which renders

throughputs smaller than 18% because of the need to relay and acknowledge packets. The simulation results obtained using the parameters of the Utilicom radio also show the importance of carrier sensing; because the Utilicom radio does not provide true carrier sensing, the performance of FAMA-NCS degrades substantially, as predicted by the theory [FG95b, FG97b].

We have also shown that WIRP provides internet routing in the ad-hoc network environment and converges efficiently, even when competing with heavy traffic for bandwidth to send its routing-table update information. The simulation results presented assumed *single-path routing*, in which the protocol provided a single path to each destination. A new version of WIRP provides multiple paths, and we are developing new queueing schemes for the WINGs to establish a late binding of packets to their next hops, so that packets can be rerouted around failures more efficiently.

Implementing the WINGs has been simplified by our use of CPT, which allowed us to carry out simulations of large network topologies using the complete WING protocol suite, with each protocol being implemented exactly as it would be running in a WING, and then use the very same code written for our simulations in the actual prototype by simply recompiling. This eliminated the time needed to rewrite the protocols, as well as the associated recoding errors.

Chapter 7

Conclusion

We have introduced new channel access methodology we call *Floor Acquisition Multiple Access* (FAMA) that provides better throughput performance in ad-hoc networks than has been previously shown. We analyze and compare this methodology in fully-connected and multi-hop networks. We show results for both single-channel and multiple channel devices. With these results we provide a formal verification of collision avoidance techniques in ad-hoc networks.

Our first contribution is to show the importance of carrier sensing in single-channel networks. We give results demonstrating that carrier sensing provides higher throughput than packet sensing in fully-connected networks.

Our second contribution shows that carrier sensing used in conjunction with a CTS packet that overlaps an RTS packet (which we define as *CTS dominance*) is sufficient to provide collision avoidance in single-channel ad-hoc networks.

Our third contribution is the first average throughput analysis of medium access control protocols based on “floor acquisition” in both full-connected and ad-hoc networks.

Because collision detection is not practical in single-channel radio devices, our fourth contribution is the design and analysis of a channel access method that employs passive jamming (FAMA-PJ), emulating collision detection in fully-connected networks.

Our final contribution is the introduction of a channel access method for multiple channel devices (FAMA-MC) which provides a floor for each network channel.

New radio devices are being developed that have the capability to receive multiple streams of data concurrently, as well as the ability to transmit and receive concurrently. Our future work will continue to extend this methodology to these and other new types of radio devices as they become available.

Bibliography

- [ABB⁺96] A. Alwan, R. Bagrodia, N. Bambos, M. Gerla, L. Kleinrock, J. Short, and J. Villasenor. Adaptive Mobile Multimedia Networks. *IEEE Personal Communications*, pages 34–51, 1996.
- [Abr70] N. Abramson. The ALOHA System - Another Alternative for Computer Communications. In *Proceedings of the Fall Joint Computer Conference*, pages 281–85, 1970.
- [BDSZ94] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang. MACAW: A Media Access Protocol for Wireless LAN's. In *Proceedings of ACM SIGCOMM '94*, pages 212–25. ACM, 1994.
- [Bey] D. Beyer. Network visualizer (netviz) network animation tool. www.rooftop.com – Research and Development page.
- [Bey97] D. Beyer. *Physical Radio Interface*. Rooftop Communications, 1997. www.rooftop.com – Radio Interface page.
- [BFL96] D. Beyer, T. Frivold, D. Lancaster, and M. Lewis. *Radio Device API*. Rooftop Communications, 1996. www.rooftop.com – Radio Interface page.
- [BG87] R. L. Brewster and A. M. Glass. Throughput Analysis of Non-Persistent and Slotted Non-Persistent CSMA/CA Protocols. In *4th International Conference on Land Mobile Radio*, pages 231–6. Institution of Electronic and Radio Engineers, 1987.
- [BG89] R. L. Brewster and A. M. Glass. Simulation model of sequential multichannel network and its throughput determination. *Electronic Letters*, 25(15):941–2, 1989.
- [Bib92] Ken Biba. A Hybrid Wireless MAC Protocol Supporting Asynchronous and Synchronous MSDU Delivery Services. Technical Report Paper 802.11/91-92, IEEE 802.11 Working Group, 1992.
- [BN95] D. Beyer and B. Nguyen. *The C++ Protocol Toolkit: Overview*. Rooftop Communications, 1995. Also see www.rooftop.com – Research and Development page.

- [BVG97] D. Beyer, M. Vestrich, and J. J. Garcia-Luna-Aceves. The Rooftop Community Network: Free, High-speed Network Access for Communities. *Harvard University's Information Infrastructure Project*, 1997.
- [CFL79] Imrich Chlamtac, William R. Franta, and K. Dan Levin. BRAM: The Broadcast Recognizing Access Method. *IEEE Trans. Commun.*, COM-27:1183–89, 1979.
- [CG97] H. S. Chhaya and S. Gupta. Performance modeling of asynchronous data transfer methods of ieee 802.11 mac protocol. *Wireless Networks*, 3(3):217–34, 1997.
- [Che94] Kweng-Cheng Chen. Medium access control of wireless lans for mobile computing. *IEEE Network*, 8(5):50–63, 1994.
- [Chu91] K. C. Chua. Performance analysis of multichannel CSMA/CD network with noisy channels. In *IEEE ICC'91*, 1991.
- [Col83] Alama Colvin. CSMA with collision avoidance. *Computer Communications*, 6(5):227–35, 1983.
- [FG95a] Chane L. Fullmer and J. J. Garcia-Luna-Aceves. FAMA-PJ: A Channel Access Protocol for Wireless LANs. In *Proceedings of ACM Mobile Computing and Networking '95*. ACM, 1995.
- [FG95b] Chane L. Fullmer and J. J. Garcia-Luna-Aceves. Floor Acquisition Multiple Access (FAMA) for Packet-Radio Networks. In *Proceedings of ACM SIGCOMM '95*. ACM, 1995.
- [FG97a] Chane L. Fullmer and J. J. Garcia-Luna-Aceves. Complete Single-Channel Solutions to Hidden Terminal Problems in Wireless LANs. In *Proceedings of ICC '97*. IEEE, 1997.
- [FG97b] Chane L. Fullmer and J. J. Garcia-Luna-Aceves. Solutions to Hidden Terminal Problems in Wireless Networks. In *Proceedings of ACM SIGCOMM '97*. ACM, 1997.
- [GBF⁺97] J. J. Garcia-Luna-Aceves, Dave Beyer, Chane L. Fullmer, Thane Frivold, and Ewerton Madruga. Wireless Internet Gateways (WINGS). In *Proceedings of IEEE MILCOM '97*. IEEE, 1997.
- [GF98] J. J. Garcia-Luna-Aceves and Chane L. Fullmer. Floor Acquisition Multiple Access (FAMA) in Single-Channel Wireless Networks. *Journal of Special Topics in Mobile Networks and Applications (MONET) – Special Issue on Ad-Hoc Networks*, 1998.
- [glo] DARPA *darpa global mobile glomo information systems program*. www.glomo.sri.com.
- [Hu91] L. Hu. Local throughput performance of packet radio networks with transmitting power control. In *IEEE ICC'91*, 1991.

- [IEE97] *Wireless LAN Medium Access Control (MAC) and Physical Specifications, IEEE std. 802.11.* IEEE, January 1997.
- [Kar90] Phil Karn. MACA - a new channel access method for packet radio. In *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pages 134–40. ARRL, 1990.
- [KS80] Leonard Kleinrock and Michel O. Scholl. Packet Switching in Radio Channels: New Conflict-Free Multiple Access Schemes. *IEEE Trans. Commun.*, COM-28:1015–29, 1980.
- [KT75] Leonard Kleinrock and Fouad A. Tobagi. Packet switching in radio channels: Part I - carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Trans. Commun.*, COM-23(12):1400–1416, 1975.
- [Leo94] Alberto Leon-Garcia. *Probability and Random Processes for Electrical Engineering.* Addison-Wesley, 1994.
- [Leu90] V. C. M. Leung. Multichannel reservation protocol for packet multiple-access communications. *Electronic Letters*, 26(20):1637–8, 1990.
- [LM84] W. F. Lo and H. T. Mouftah. Carrier Sense Multiple Access with Collision Detection for Radio Channels. In *IEEE 13th International Communications and Energy Conference*, pages 244–47. IEEE, 1984.
- [LNT87] B. M. Leiner, D. L. Nielson, and F. A. Tobagi, editors. *Proceedings of the IEEE*, volume 75. IEEE, January 1987.
- [Lof96] C.-G. Lof. Packet Delay for CSMA and Multi-channel ALOHA Multicast Schemes in WLANs with Fading and Co-channel Interference. In *IEEE Int. Conference on Universal Personal Communications*, 1996.
- [MB76] R. M. Metcalfe and D. R. Boggs. ETHERNET: Distributed packet switching for local computer networks. *Communications of the ACM*, 19(7):395 – 403, 1976.
- [MG96] S. Murthy and J. J. Garcia-Luna-Aceves. An Efficient Routing Protocol for Wireless Networks. *ACM Mobile Networks and Applications Journal*, 1(2), 1996.
- [ML87] J.S. Meditch and C.T.A. Lea. Stability and Optimization of CSMA and CSMA/CD Channels. In *Advances in Local Area Networks*, pages 340–362. IEEE Press, 1987.
- [MN91] A. M. Marsan and F. Neri. A Simulation Study of Delay in Multichannel CSMA/CD Protocols. *IEEE Trans. on Comm.*, 39(11), 1991.
- [MR83] A. M. Marsan and D. Roffinella. Multichannel local area network protocols. *IEEE JSAC*, SAC-1(5):885–97, 1983.

- [pr-89] *Packet Radio Network Research, Development, and Application*, 1989.
- [Rom86] Raphael Rom. Collision Detection in Radio Channels. In *Local area and multiple access networks*, pages 235–49. Computer Science Press, 1986.
- [RS90] Raphael Rom and Moshe Sidi. *Multiple Access Protocols Performance and Analysis*. Springer-Verlag, 1990.
- [SAO90] Gursharan S. Sidhu, Richard F. Andrews, and Alan B. Oppenheimer. *Inside AppleTalk, Second Edition*. Addison-Wesley Publishing Company, Inc., 1990.
- [SK87] N. Shacham and P. King. Architectures and Performance of Multichannel Multihop Packet Radio Networks. *IEEE JSAC*, SAC-5(6):1013–1025, 1987.
- [sur90] *Accomplishments of the DARPA SURAN Program*, 1990.
- [TK75] Fouad A. Tobagi and Leonard Kleinrock. Packet switching in radio channels: Part II - the hidden terminal problem in carrier sense multiple-access modes and the busy-tone solution. *IEEE Trans. Commun.*, COM-23(12):1417–1433, 1975.
- [TK76] Fouad A. Tobagi and Leonard Kleinrock. Packet switching in radio channels: Part III - polling and (dynamic) split-channel reservation multiple access. *IEEE Trans. Commun.*, COM-24(8):832–845, 1976.
- [TK78] Fouad A. Tobagi and Leonard Kleinrock. The Effect of Acknowledgment Traffic on the Capacity of Packet-Switched Radio Channels. *IEEE Trans. Commun.*, COM-26(6):815–826, 1978.
- [TK85] Hideaki Takagi and Leonard Kleinrock. Output processes in contention packet broadcasting systems. *IEEE Trans. Commun.*, COM-33(11):1191–9, 1985.
- [Tob74] Fouad Tobagi. *Random Access Techniques for data transmission over packet switched radio networks*. PhD thesis, University of California, Los Angeles, Computer Science Department, School of Engineering and Applied Science, 1974. Report UCLA-ENG 7499.
- [Tri88] Kishor S. Trivedi. *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*. Prentice-Hall, 1988.
- [Vad95] Bharghavan Vaduvur. *Access, Addressing and Security in Wireless Packet Networks*. PhD thesis, University of California, Berkeley, Computer Science Department, 1995.
- [Wei93] M. Weiser. Some computer science issues in ubiquitous computing. *Communications of the ACM*, pages 74–84, 1993.
- [WIN] Wings for internets project. <http://www.cse.ucsc.edu/research/ccrg/projects/wings.html>.

- [WL87] C. Wu and V. O. K. Li. Receiver-initiated busy-tone multiple access in packet radio networks. In *ACM SIGCOMM 87 Workshop: Frontiers in Computer Communications Technology*, 1987.
- [WSFW97] J. Weinmiller, M. Schlager, A. Festag, and A. Wolisz. Performance study of access control in wireless lans—ieee 802.11 dfwmac and etsi res 10 hiperlan. *Mobile Networks and Applications*, 2(1):55–68, 1997.