UNIVERSITY OF CALIFORNIA

Los Angeles

Medium Access Control Protocols for Satellite Networks

A dissertation submitted in partial satisfaction of the

requirement for the degree of Doctor of Philosophy

in Electrical Engineering

by

Dennis Patrick Connors

2000

The dissertation of Dennis Patrick Connors is approved.

Kirby A. Baker

Richard D. Wesel

Kung Yao

Gregory J. Pottie, Committee Chair

University of California, Los Angeles

2000

This work is dedicated to my mother, Geraldine Ann Connors, who sacrificed her life so that I might have mine.

Contents

List of Figures

List of Tables

ACKNOWLEDGEMENTS

I would like to first give thanks to Almighty God for giving me the ability and giving me the perseverance to complete this degree.

I would like to acknowledge and thank my advisor, Professor Gregory J. Pottie for his support and patience throughout my doctoral studies. I would also like to thank my dissertation committee members: Dr. Kirby A. Baker, Dr. Richard D. Wesel, and Dr. Kung Yao for their help in serving on my committee

I would next express my gratitude and thanks to all my graduate student friends who made my tenure at UCLA enjoyable. First and foremost to Bruce Kwan, a friend the likes of which I will most likely never find again. Also I would like to mention and thank Dengwei Fu, Jane Yu, Michelle Li, Jeffrey Putnam, Larry Wasserman, John Leete, Kathy Sohrabi, Chris Hansen, George Kondylis, Heung-No Lee, Victor Lin, Keith Scott, Jay Gao, John Siwko, Charles Wang, and Tommy Yu. I apologize in advance if I have forgotten anyone.

I would also like to acknowledge my colleagues at HRL Laboratories (formerly Hughes Research Laboratories) whom I formed a close bond with. They are Chuck Wu, Srikanth Krishnamurthy, Andy Davis, Dante De Lucia, Greg Kaestle, Eddie Shek, Asha Velliakal, Jihoon Yang, Wensheng Zhou, Bikram Singh, Bong Ryu, and Yongguang Zhang. I am indebted to Son K. Dao for hiring me and giving me the chance to prove myself in the research lab environment.

I extend my deepest gratitude to Ms. Daphne Chi for being a true friend throughout all of our time in Los Angeles and for being a pillar of strength when things were so dark.

Lastly I would like to thank my beloved Emily Dizon Lacson for putting up with a man who could not give her the time she rightfully deserved.

VITA

November 26, 1969	Born, Meiringen, Switzerland
1992	B.S. Electrical Engineering San Jose State University San Jose, California
1994	M.S. Electrical Engineering University of California, Los Angeles Los Angeles, California
Summer 1994,95	Communications Systems Engineer Digital Radio Corporation Redondo Beach, California
1994-97	Research Assistant Electrical Engineering Department University of California, Los Angeles
1997-2000	Research Staff Member H.R.L. Laboratories Malibu, California (formerly Hughes Research Labs.)
2000-Present	Director of Data Link Control and Networking Magis Networks Incorporated San Diego, CA

PUBLICATIONS

D. P. Connors and B. Ryu, **Performance Evaluation of Satellite MAC protocols with QoS Guarantees**, *Proc. 2nd Workshop on Satellite Based Information Services, (WOSBIS'97)*, October 1997.

D. P. Connors, B. Ryu, and S.K. Dao, **Modeling and Simulation of Broadband Satellite Networks Part I: Medium Access Control for QoS Provisioning**, *IEEE Communications Magazine*, March 1999.

J. Ling and D. P. Connors, **Medium Access Control in Satellite Networks with non-Stationary Spot Beams**, *Proc. Fourth Annual Workshop on Satellite Based Information Services, (WOSBIS'99)*, December 1999.

D. P. Connors and G. J. Pottie, **The Performance of HTTP over Satellite Random Access Channels**, *Communications Networks and Distributed Systems Modeling and Simulation*, January 2000.

I. Koutsopoulos and D. P. Connors, **Intra-Team Multi-Hop Broadcasting: A MAC Layer Protocol for Efficient Control Signaling in Wireless Ad-Hoc Networks**, *Proc. International Conference on Communications (ICC 2000)*, June 2000.

T. ElBatt, S. V. Krishnamurthy, and D. P. Connors, **Power Management for Throughput Enhancement in Wireless Ad-Hoc Networks**, *Proc. International Conference on Communications (ICC 2000)*, June 2000.

D. P. Connors, G. J. Pottie, B. Ryu, and S.K. Dao, **A Medium Access Control Protocol for Real Time Video over High Latency Satellite Channels**, *Baltzer / ACM Journal on Special Topics in Mobile Networking Applications (MONET)*, To Appear Fall 2000.

ABSTRACT OF THE DISSERTATION

Medium Access Control Protocols for Satellite Networks

by

Dennis Patrick Connors

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2000

Professor Gregory J. Pottie, Chair

The means of achieving full duplex broadband access to either the home or office has been the focus of much research and entrepreneurial development in the past five years. This is the so-called “last mile” problem. One approach is duplex access to a geostationary earth orbit satellite through a relatively small satellite dish located at each home or office. This scenario gives rise to a multiple access problem in the uplink (i.e. terrestrial earth terminal to satellite) channel, since this channel is considered *many-to-one*.

We first focused on real-time medium quality video transmission. To efficiently transport this medium, we developed a new medium access control protocol, *Random Access / Demand Assigned Multiple Access* (RA/DAMA). The RA/DAMA medium access

technique uses a time division multiple access Physical Layer and seeks to adaptively minimize the delay of each Network Layer packet that arrives to the output queue by transmitting packets on either a collision free demand assigned channel or on a collision possible random access channel. Combined with this dual channel transmission method is a new technique for acquiring demand assigned bandwidth, called a packet flow rate metric. This metric seeks to track the slow time behavior of video traffic, leading to a significant reduction in the amount of demand assigned multiple access signaling. We conducted extensive simulations of this protocol, using the network simulator **ns**. Our results show that if light packet loss is tolerable (less than three percent), then significantly lower delays and higher link utilization can be achieved.

We next turned our attention to Hyper Text Transfer Protocol based World Wide Web browsing, bulk file transfers, and interactive computing. To accommodate these traffic types, we developed another medium access technique dubbed *Response Initiated Multiple Access* (RIMA). RIMA also uses a time division multiple access Physical Layer, but relies on replies from a remote server to instantiate collision free uplink bandwidth for clients attached to terrestrial earth terminals. We again conducted extensive simulations using **ns** and showed that for all three types of traffic under consideration, RIMA significantly out performs previously proposed medium access control protocols designed for computer data applications.

Chapter 1

Introduction

We are at the dawn of a new millennium and communications technology is drastically affecting not only how we conduct our daily affairs, but also significantly influencing our culture. Within the general realm of communications, the two areas that are seeing explosive growth are wireless communications and the packet switched Internet. Wireless communications has now taken many forms in the commercial arena. Currently the most dominant usage of this technology is in satellite media distribution, broadcast television and radio, satellite paging, cellular telephony, cordless in-home telephony, and to a limited degree, wireless local area networks (LANs). On the horizon are such technologies as wireless local loop (often referred to as fixed wireless access), high speed wireless LANs, telephony via low earth orbit (LEO) satellites, and high speed optical wireless communications. The Internet, which is supported by a concatenation of many forms of communication, is having a profound effect on our way of life. Mail can now be sent electronically, with images and sound attached, information can be shared and disseminated, stocks can be traded by amateurs, and even the age-old profession of news gathering and reporting is being supplanted by “Internet Journalists”. These two explosive technologies are in the process of colliding. Wireless access to the Internet and all its wealth is now becoming a reality. This dissertation deals with one aspect of this inevitable convergence, that being the efficient delivery of Internet content via satellite wireless technology.

1.1 A Brief History of Satellite Communications and Inter-Networking

Satellite communications has rapidly evolved since the first commercial deployment of satellites in 1965. The first satellites deployed were geosynchronous earth orbit (GEO) satellites. These satellites orbit high enough above the earth’s surface that they appear stationary from the ground. Recently there has been activity and initiative in the area on non-geostationary orbit (non-GSO) satellites [51, 22]. These types of satellites are referred to as low earth orbit (LEO) or medium earth orbit (MEO), depending upon the orbital altitude. Along with launched technology (often referred to as payload), there has been much innovation towards the goal of reducing the size of the ground station, often referred to as the *earth terminal*. When satellites were first deployed, the earth terminal antennas required for duplex transmission were upwards of 30 meters in diameter [33]. Now, with very small aperture terminal (VSAT) technology, duplex transmission can now be accomplished with a 1.2-1.8 meter antenna, with receive only antennas on the order of 60 cm. A Japanese company has recently announced the development of a 75 cm VSAT terminal capable of two way communication with a GEO satellite [30]. This technological advancement has lead to Internet access and broadcast video dissemination to dispersed end users almost anywhere on the globe.

The growth and adoption of the Internet and its packet switched technology followed a different path than that of satellite technology. The technology behind the Internet began in the mid-1960's, ironically around the same time as the launch of the first satellites. The original goal was for the Advanced Research Projects Agency (ARPA) to develop a packet switched command-and-control network capable of surviving a nuclear war. The work was originally contracted out to BBN, a consulting firm in Cambridge, Massachusetts. This network was called the ARPANET [52]. In 1969, the experimental ARPANET went on-line with four nodes, at UCLA, University of California Santa Barbara, SRI Inc., and the University of Utah. The ARPANET quickly grew, spanning the United States through a connection of various university campuses and research facilities. The first application of wireless technology to the ARPANET occurred in the early 1970s when a successful experiment was conducted wherein a truck was connected, via a wireless packet network, to SRI, which forwarded all incoming packets via the ARPANET to the East Coast where they were transmitted via satellite to University College in London. In 1974, the Transmission Control Protocol / Internet Protocol (TCP/IP) suite of protocols was developed. This protocol suite allowed for heterogeneous computers to communicate via a common protocol stack. In 1984, the National Science Foundation (NSF) decided to build a successor network, using the same packet switched technology as the ARPANET, but using TCP/IP from the very beginning. This network, NSFNET, was primarily academic in nature as it connected most American universities. The line speed of the NSFNET was gradually upgraded from 448 kb/s, to 1.5 Mb/s, to 45 Mb/s and gradually began to be referred to as the *Internet*. Eventually private companies began to develop TCP/IP networks, so the Internet in time became less supported by the government and more by several private companies. By 1992, the one millionth host was added to the Internet. Throughout all this time, the Internet was mostly used by research institutions and academia. The World knew what satellites were and those TV addicts or those in remote areas had individual satellite dishes so as to receive hundreds of programming channels. Hardly anyone outside of academia knew what the Internet was or the wealth of information it could provide. That changed in 1993 when the Hypertext Transfer Protocol (HTTP) was introduced. This Session Layer protocol still utilized the underlying TCP/IP protocol suite, while greatly simplifying the procedures for retrieving and forwarding information. This along with the introduction of the World Wide Web lead in short time to rapid public acceptance of the Internet and the current “Dot Com Gold Rush” that we see today.

1.2 Scope of the Dissertation

The goal of our work is to develop and evaluate medium access protocols, for satellite communication channels, that are optimized to *effectively* and *efficiently* deliver multimedia Internet content. Medium access control (MAC) is a mechanism whereby a shared wireless channel is managed. A brief introduction to MAC protocols will be given in Chapter 3. By effectively, we mean that the end user, located at a earth terminal, achieves a suitable *Quality of Service* for the applications that the user is running. Likewise we take efficiently to mean that the satellite link is fully utilized, thus maximizing the revenue that the operator of the satellite network can generate. This joint optimization often involves many trade-offs, which will be expanded upon in the subsequent chapters. Since both satellite networks and Internet applications are very diverse, we will narrow our focus. The network infrastructure will consist of an earth terminal, capable of duplex communication with a GEO satellite, attached directly to a host. This scenario, referred to as the "personal earth terminal" model, will be described in more detail in Chapter 3. The Internet applications that we will focus our attention on are uplink streaming of real-time video, World Wide Web browsing, bulk file transfers from client to server, and interactive computing.

This dissertation will be organized as follows. Chapter 2 will give the necessary background information to achieve an appropriate understanding of both a layered communication network and of a satellite network in particular. Chapter 3 will give an outline of medium access control in the general sense and then go into more detail on MAC protocol design in the context of satellite networks. This will be followed in Chapter 4 by a detailed presentation of a medium access control protocol, *random access / demand assigned multiple access* (RA/DAMA), which seeks to minimize delay and maximize capacity for video sources accessing a GEO satellite uplink channel. Chapter 5 will present a second medium access technique named *response initiated multiple access* (RIMA). RIMA is a medium access scheme, also designed for a GEO channel which, seeks to minimize the file transfer times for terrestrial World Wide Web users, while at the same time maximizing the number of users that can be supported in a given portion of uplink bandwidth. Finally Chapter 6 will conclude with both an outline for implementation of RA/DAMA and RIMA on a common time division multiple access (TDMA) satellite channel and some thoughts for future work.

Chapter 2

Background

This dissertation will detail two resource management algorithms. The resource under consideration is wireless radio frequency bandwidth. Wireless communication between two entities is made possible when there is a wireless “link” available. This link can be either available or unavailable, and when available its quality can vary. Considering now only available wireless links, the source of the varying quality can be many factors, among them loss of transmitted energy due to distance or obstructions, multiple reflections of the transmitted signal, thermal noise, and interference from the transmitted energy of other communicating entities. Medium Access Control (MAC) protocols pertain to the last form of quality variation, that being interference.

When entities communicate, they utilize a certain amount of bandwidth (i.e. a portion of radio frequency spectrum) during transmission. Since communication requires at least two communicating entities (i.e. a sender and receiver(s)), if two separate communicating pairs choose to communicate at the same time, using the same bandwidth, in the same relative proximity, the receivers in each pair will experience interference from the other pairs' sender. This is commonly referred to as a *collision*. In order to eliminate, or at least minimize, these collision events, the wireless resource (bandwidth) must be managed. The means by which this management is accomplished is through a medium access control protocol. Chapters 4 and 5 detail two MAC protocols designed for bandwidth management in satellite communication networks.

2.1 Layering in Communication Networks

Before continuing into a detailed description of the two MAC protocols, a brief overview of a communication network will be given. Traditionally two communicating entities (CE) will have information that needs to be shared. Referring to Figure 2.1, this information is communicated through a network which (usually) adheres to the open systems interconnection (OSI) layering model[7]. At each layer, the information will be represented in a different manner. At the Application Layer for instance, information is presented in the most user appropriate manner, an example of which is a Web Page as seen through a browser. At the Transport Layer, information is received in messages and delivered to the Network Layer as packets. The Network Layer determines how to "route" a packet received from the Transport Layer and appends this *routing information* to the packet. Routing is necessary in what is called a *multi-hop network*. Multi-hop networks exist when the link between two CEs is actually a concatenation of several links connecting multiple CEs, all of these links forming a route from the source CE to the destination CE. The most widely known multi-hop network is the Internet. At each stage in the layering process, the information stream is fragmented and additional information (sometimes redundant) is added. When finally the information transmitted at the Physical Layer, it is represented as continuous *channel symbols*, with additional redundancy to combat the previously mentioned sources of channel degradation. Following this layering model, research can be conducted focusing on one layer while ignoring, or at least making simplified assumptions, about the other layers. The whole idea of OSI layering in communication networks is to isolate each layer from the next so as to make a system that is not dependent on a particular application or a particular communication physical link. Each layer would pass information to adjacent layers using what is called an application program interface, or API. Employing this techniques has its benefits, since communications networks are very complex, however as will be shown later in this dissertation, when addressing a problem particular to one layer, considering the interaction / effects of other layers can lead to much better solutions.

2.2 Physical Layer

At the Physical (PHY) Layer, information is represented as analog channel symbols. These symbols are subject to degradation as they are transmitted across the *channel*. For wire-

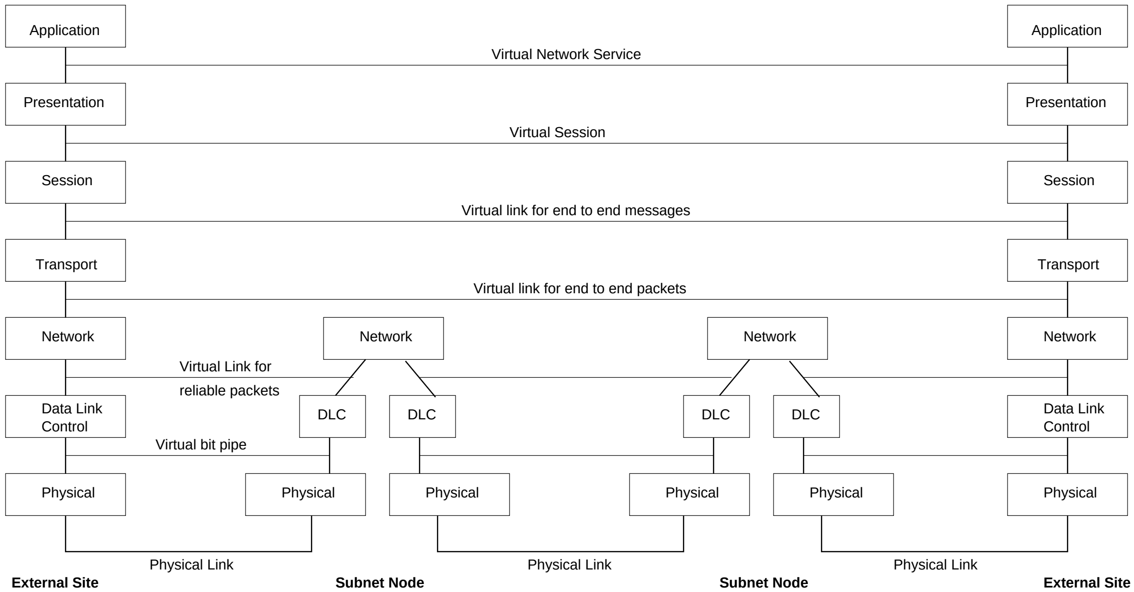

Figure 2.1: Seven Layer OSI Network Architecture

less channels this degradation can take the form of path loss, shadowing loss, multi-path interference, thermal noise, and multiple access interference. Interference due to multiple communicating entities accessing the same bandwidth (forthwith referred to as multiple access interference, or MAI) is to some degree controllable, however the other forms of link quality variation are usually not. In order to combat these “link degradations”, many different techniques (i.e. channel coding, coherent demodulation, adaptive equalization, smart antennas, power control, etc.) have been developed [44]. The layering architecture presented in Figure 2.1 made no presuppositions about the nature of the *physical link*, thus making it applicable to both wireless, wire-line, and hybrid wireless-wireline networks.

2.3 Data Link Control Layer

The primary role of the Data Link Control (DLC) Layer is to transform the potentially unreliable physical link into a reliable "virtual pipe" between two hops on a communication network. This is accomplished by a combination of scheduling, framing, error detection, and retransmission. The scheduling aspect of the DLC in a multi user environment is what constitutes a multiple access protocol.

CEs employ multiple access when the physical link in Figure 2.1 is shared by other CEs. The most common occurrence of this in wireline networks is with Ethernet. There are a plethora of examples of multiple access in wireless networks, since radio frequency bandwidth can be accessed by any CE with the proper equipment (i.e. FDMA, CDMA, TDMA in cellular telephony, IEEE 802.11 in wireless LANs, etc). When multiple access is necessary in a communication network, a medium access control (MAC) mechanism must be in place in the layering hierarchy. Traditionally the MAC layer is considered a *sub-layer* of the DLC Layer, residing between the Data Link Control Layer and the Physical Layer as shown in Figure 2.2.

At the DLC, the physical links of a wireless network can fall into two general categories: point-to-point links and point-to-multipoint links. With point-to-point links, a CE simply communicates with its peer CE. Usually in this case, multiple access is simple and the MAC sub-layer is trivial or sometimes absent. Point-to-multipoint links on the other hand require sometimes very innovative approaches to medium access control, since there are

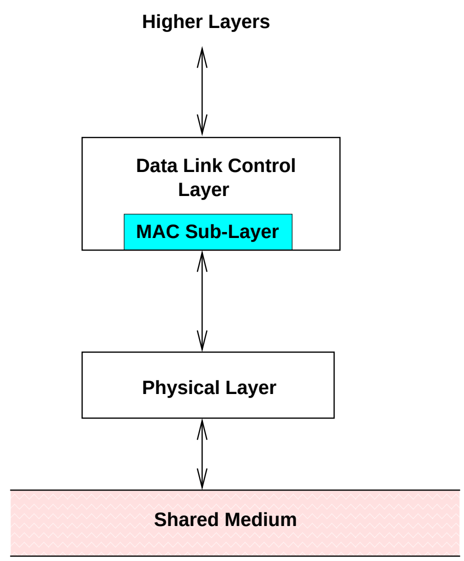

Figure 2.2: The MAC Sub-Layer incorporated into the Physical Layer

multiple CEs attempting to gain access to the shared communication medium (i.e. bandwidth). Point-to-multipoint links can further be categorized into two different topologies: *star* and *ad-hoc*. In a *star* topology, there is one principal CE which will be referred to as an *arbitrating agent*, or AA (this will also be called an *allocating* agent). The role this AA plays is to allocate bandwidth to the communicating CEs and to coordinate to some degree their communication behavior. An example of this is a cellular telephone system, where the AA is referred to as a *base station*. Usually the AA is stationary while the other CEs can be mobile. However there are some systems where even the AA is mobile [27, 51]. In an *ad-hoc* topology there is no AA and the CEs are often referred to as “nodes” in the network. Often ad-hoc networks are mobile and find most of their use in environments where a star topology “infrastructure” has not been established due to time constraints (i.e.

military applications) or when there is no economic reason for doing so. Since coordination amongst ad-hoc nodes must be done in a distributed fashion, many interesting research problems, relating to medium access control, routing, and multicasting, still exist.

For the purposes of our work, we will be focusing on geosynchronous earth orbit (GEO) satellites forming a star topology with terrestrial earth terminals. A GEO satellite resides in the neighborhood of 35,800 km above the earth's surface and orbits the earth in the same direction that the earth rotates, so as to appear stationary from the point of reference of one residing on the ground. This characteristic is attractive for the reason that the satellite can be communicated with at all times by any earth terminal within the satellites' "spot" beam. Figure 2.3 shows how two earth terminals would communicate in a GEO satellite network. It should be clear that the challenges of multiple access occur on the *uplink* since that channel is a many-to-one channel. The *downlink* is a one-to-many channel and as such has no complicated multiple access issues.

2.4 Network Layer

The Network Layer performs the function of routing and flow control in a multi-hop network. When frames of information arrive at the DLC, it removes any DLC specific information and passes the information up to the Network Layer. These units of information are commonly referred to as packets once they have arrived at the Network Layer. The Network Layer can then either pass the packet up to the Transport Layer, if the destination

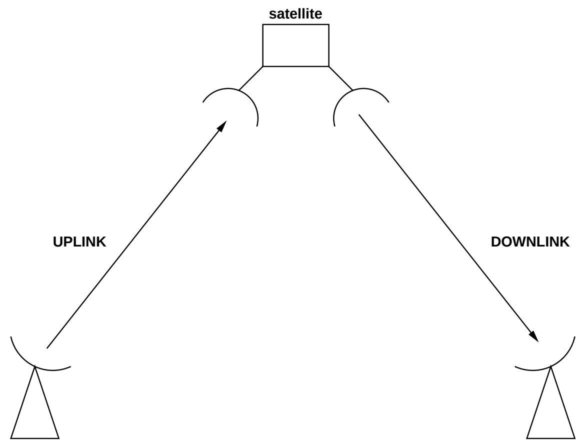

Figure 2.3: An Illustration of Earth Terminals Communicating in a GEO Network

of the packet is a host residing on this hop, or forward the packet back down to the appropriate outgoing DLC Layer, depending upon the routing information contained within the packet. If the packet is initially received from the Transport Layer, then the Transport Layer messages have been segmented into packets and routing information has been attached prior to its delivery to the Network Layer. In order for the Network Layer to route packets, it must have additional routing information and a routing protocol. In the Internet, routing information is contained in *route tables* and the routing protocol used is a *link-state* protocol called Open Shortest Path First (OSPF) [37].

For completeness sake, two common networking terms will be defined : *unicast* and *multicast*. Unicasting takes place when there is a duplex end-to-end connection between two CEs existing at the transport layer. This does not preclude a multi-hop connection, where packets must be stored and forwarded from multiple CEs in order to reach the destination.

It simply means that when a CE's transport layer emits a packet, it has only one destination. TCP is traditionally the Transport Layer protocol of choice for unicasting. With multicasting, there are potentially several sources and destinations that make up a *multicast group*. Many times in multicasting, there is a multicast sender (or source) which is transmitting packets to a multicast group, considered multicast receivers. Each node (CE) in the network forwards and possibly replicates the packet according to the routes to each member of the multicast group. This is shown pictorially in Figure 2.4. There exists in the Internet, the Multicast backBone, or MBone [31], which uses Internet multicasting techniques to accomplish multicasting. For a more thorough explanation of multicasting in the Internet, see [15]. Multicasting can also be comprised of multiple senders and receivers, as in a collaborative video / audio conference [14].

2.5 Transport Layer

Referring back to Figure 2.1, the Transport Layer is the first layer (considering the Physical Layer to be layer 1) that considers the communication link an *end-to-end* connection. The transport layer receives messages from the higher layers and translates these messages into "packets". These packets are then sent according to the particular transport layer protocol. This can take many forms, but for the purposes of this research, we have chosen to focus on one suite of protocols, that being the **TCP/IP** suite [50]. The TCP/IP suite of protocols focuses on the Session, Transport, and Network Layers of Figure 2.1. For more information

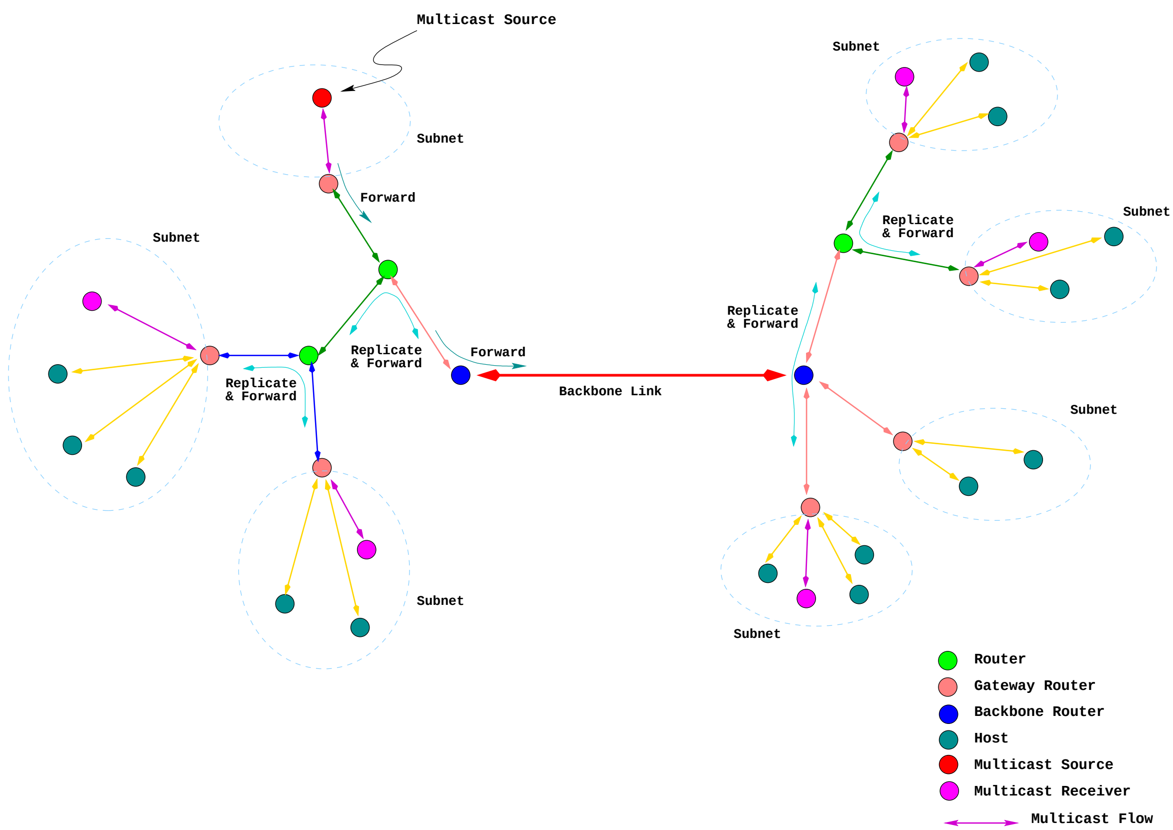

Figure 2.4: A Multi-Hop Network Performing a Multicast

on TCP/IP, please refer to [50]. It was designed primarily with the underlying assumption of a wireline or high reliability wireless link existing at the physical layer. The two dominant transport layer protocols of the TCP/IP suite are the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP).

2.5.1 Transmission Control Protocol

The Transmission Control Protocol (TCP) [43] is a reliable, connection-oriented Transport Layer protocol. TCP uses a window based flow control scheme to adapt its bandwidth usage and achieves its reliability via end-to-end *acknowledgements*. TCP was designed and subsequently modified [24] to adapt to network congestion, and not to packet loss due to an unreliable physical link. Several attempts have been made to make TCP performance improve over wireless channels [5], with moderate degrees of success. However TCP as it is now implemented on millions of hosts that make up the Internet has not incorporated any of these enhancements. The responsibility to make a high throughput connection then falls on the Data Link Control, MAC, and Physical Layers. Work to be presented in Chapter 5 deals with a high throughput, high capacity MAC protocol for World Wide Web (WWW) browsing over satellite. In that chapter some of the more subtle aspects of TCP will be described.

2.5.2 User Datagram Protocol

Aside from TCP, there exists another TCP/IP Transport Layer protocol in high usage in the Internet, and that is the *user data-gram protocol* or UDP [42]. Unlike TCP, UDP is considered unreliable and connectionless. UDP simply receives messages from the Session Layer and produces Network Layer packets according to the maximum transmit unit (MTU) size. There are no end-to-end acknowledgements and no network congestion adaptability in UDP. It simply transmits packets without regard to whether or not they are indeed arriving at the intended destination(s). UDP finds most of its use in Domain Name Server (DNS) lookups and in real-time audio / video Internet applications. Work to be presented in Chapter 4 details a low delay, high capacity MAC layer protocol suitable for real-time video applications using UDP over geostationary earth orbit (GEO) satellite channels.

2.6 Session Layer and Above

The remaining layers are not as clearly understood nor as important to this work, however for completeness sake a brief description of each will be given. A Session Layer protocol is a technique for managing multiple Transport Layer streams to form a session. The Web browsing protocol HTTP [18] is a Session Layer protocol which runs “on top of TCP”. Depending upon which variant of HTTP is in use, potentially 4 to 7 TCP connections will be used to make up one HTTP session. The Presentation Layer performs data encryption, data compression, and code conversion [7]. The Application Layer presents the received data in a manner that is most useful to the end user. The most wide spread application is the Mosaic Viewer, written at the National Center for Supercomputer Applications and subsequently commercialized in the form of Microsoft's Internet Explorer and the Netscape Navigator.

Chapter 3

Medium Access Control Protocols and Satellite Networks

Achieving broadband access to the home or office will usher in a new era of accessibility to Internet / multimedia services. The main obstacle to this becoming a reality is that in most cases, the only connection that exists between the home or office and the high speed fiber optic backbone is a low bandwidth copper wire. Several solutions to this “last mile” problem are taking shape. They include cable modems[8], digital subscriber lines (xDSL)[34], fixed wireless access (sometimes referred to as LMDS or MMDS)[48], and duplex satellite access [30]. Each of these techniques has their strong and weak points. Ignoring for the moment which, if any, of these techniques will eventually dominate, this work is focused on the duplex satellite access (DSA) model. Referring to the satellite dish residing at either the home or office as an *earth terminal*, if DSA becomes a reality, there will exist thousands (or perhaps millions for GEO) of earth terminals requiring network services residing within a satellite's spot beam. There are many different scenarios which are well served by the DSA model. This chapter is organized as follows. We will first outline some of the inherent benefits of satellite networks and what factors make satellite networking different from terrestrial networking. Next we will present an overview of medium access control in communications, then explain some of the details and challenges of MAC protocol development in the context of satellite networks. Finally we will describe the simulation tools we have developed for rapid medium access control protocol development and validation.

3.1 Applications for Satellite Networks

Satellite networks have characteristics unique from terrestrial fiber networks. These characteristics make it very suitable for some applications and less suitable for others. The following is a overview of many common multimedia applications and their QoS requirements.

- Interactive Computing: An example of an interactive computing application is `Telnet`. In this case, a user is typing in data and expecting a prompt response from the other end of the connection. For a high latency satellite network, this will be difficult to implement.

- Bulk Transfers: This is a broad term for an application where the total amount of data to be transferred is known a priori. These applications tend not to be as sensitive to delay and delay variation as interactive applications, however delay is still an issue. Ignoring flow and congestion control mechanisms, the total transmission time is the propagation delay plus the product of file size and the reciprocal of the bandwidth. Thus as the size of the file to be transferred grows, bandwidth becomes more important than delay. Therefore applications where the file sizes are small suffer more from the effects of latency. An example of this is HTTP, the hyper text transfer protocol. Since the amount of data transferred in web pages tends to be small [2], its performance over satellite, due to TCP slow start, can be worse than terrestrial links.

- Information Dissemination: Due to the star topology of satellite networks, information dissemination can be accomplished simply and efficiently. Examples of this include situation awareness data, stock markets numbers, and medical data.

- Video Broadcasts: As with information dissemination, the broadcast nature of satellite networks make them ideally suited for video broadcasts. This application is sensitive to delay variation and not as affected by delay itself. This is because successful playback of the received sequence requires each frame to be equally spaced. Large variations in arrival times of packets degrades the quality of playback. A satellite network could be employed to broadcast video efficiently and with strong QoS guarantees.

- Video Conferencing: This application can have many forms. If very high quality (i.e. high bandwidth) point-to-point video with extremely low delays are required, then

GEO satellite systems will have a hard time providing this, due to the high latencies. However if packet loss, delays, and delay variations are not as strict, and the video conference session is multipoint-to-multipoint, then satellites are ideal. Currently the MBone [31] provides this kind of service over the terrestrial Internet. We believe MBone service over a satellite network will deliver superior performance.

3.2 Medium Access Control

Medium access control is at its essence a means of scheduling access to a shared medium for a set of "competing" communicating entities. A seminal paper / protocol on medium access control was written / developed by Professor Norman Abramson in 1970 called ALOHA [1]. Since then much work has been done on this topic, including a very informative book [45]. What follows will be a brief introduction to challenges involved with multiple access and the various medium access techniques.

3.2.1 An Overview of Medium Access Control

The goal when designing medium access control (MAC) protocols has always been to maximize the utilization of the shared channel while at the same time providing a suitable quality of service (QoS) for those users accessing the channel. For the purposes of this research, at the MAC sub-layer, we will define QoS to be what performance a system delivers in terms of packet loss, delay, and delay variation (often referred to as packet jitter). As stated previously, medium access control is only necessary when multiple CEs are trying to access the shared bandwidth. A satellite system is a point-to-mulitpoint communications network, and therefore medium access is only an issue on the uplink (i.e. the link from the earth terminal to the satellite), as is illustrated in Figure 3.1.

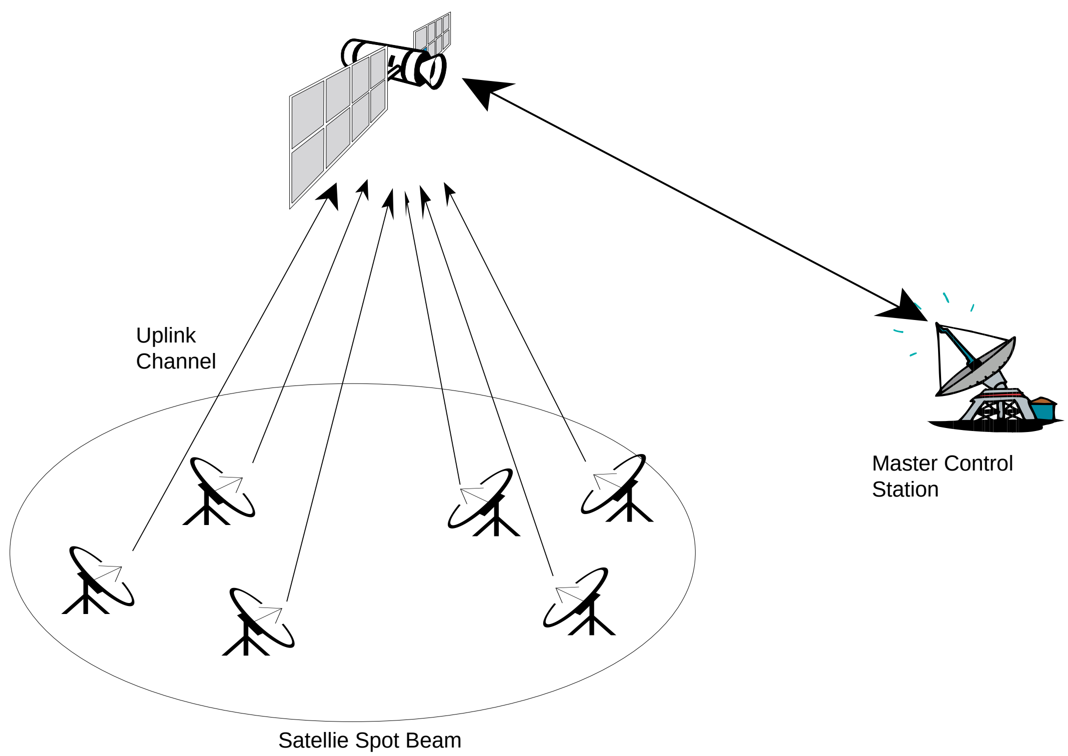

Figure 3.1: An Illustration of a Generic Packet Switched Satellite Network with Multiple Access on the Uplink Channel

For the purposes of this overview, a star topology (as mentioned in Chapter 2, Section 2.3) will be assumed for the communicating network. Medium access techniques can be crudely categorized into four access methods; random access, reservation random access, demand assigned multiple access, and fixed access. In random access, the dispersed users sharing the channel transmit packets in an uncoordinated manner to the arbitrating agent (for instance the satellite). The ALOHA protocol [1] and its variants fall into this category. Since collision-free channel resources cannot be guaranteed with random access methods, QoS guarantees, in terms of packet loss and delay, are very weak. The benefits of random, uncoordinated transmission are reduced control signaling and algorithmic overhead, and in ease of implementation. Random access MAC techniques are traditionally employed when the network traffic is unpredictable and bursty. Reservation Random Access (RRA) protocols are similar to random access techniques, except that once a packet is received at the arbitrating agent, subsequent channel slots can be *implicitly* reserved. This allows for bursts or streams of packets to be delivered in a more efficient manner than random access. Reservation ALOHA [13], and Packet Reserved Multiple Access (PRMA) are the most widely known RRA protocols. Fixed access, or fixed bandwidth allocation (FBA), lies at the other end of the medium access spectrum. In fixed access, each channel user is given dedicated bandwidth for the life of the connection. With fixed bandwidth allocation, medium access is accomplished at connection set-up. A terminal acquires a fixed amount of channel resources and maintains this resource for the life of the connection. The only time the amount of channel resource may change is when the connection is preempted by another higher priority connection. The traditional mobile cellular access techniques TDMA, FDMA, and CDMA fall into this category. The QoS guarantees that fixed access is capable of delivering are much stronger than random access, but at the expense of network capacity. If the peak rate of the source is known, then simply acquiring channel bandwidth greater than or equal to this rate will give a strong upper bound on the delays seen by packets entering the network. The problem with this technique is that for variable bit rate (VBR) sources often times significant channel bandwidth goes unused. If a traffic source has a high peak to average ratio and if the bandwidth allocated is the peak rate, then when the source is producing traffic below this peak, valuable network bandwidth is wasted. This poor link utilization leads to low network capacity, i.e. the number of CEs that can be supported within a given amount of uplink bandwidth. Demand assigned multiple access (DAMA) seeks to achieve the QoS guarantees of fixed access while achieving the statistical multiplexing gain of random access [12]. DAMA protocols consist of a phase for bandwidth request and a phase for actual data transmission. The request phase occurs in a region disjoint from the data phase in the uplink. In the request phase, data bandwidth is reserved by the terminal forming and transmitting a metric that represents its current desired bandwidth. The request phase allows terminals to communicate their instantaneous bandwidth needs to the arbitrating (allocating) agent (AA). This allows the users to *explicitly* reserve precise amounts of future bandwidth. This request phase could be itself random access. In a satellite network the AA resides either at the satellite or at a terrestrial master control station (MCS), as indicated in Figure 3.1. Bandwidth proportional to the need of an individual terminal is then alloted, subject to any shortage. In the context of TDMA systems this means that time is set aside, either in designated control slots (mini-slots) or in a "piggy-backing" manner, for the terminal to reserve time slots for transmission in future TDMA time frames. We should note that DAMA does not presuppose any particular Physical Layer transmission format. It can be implemented using TDMA, FDMA, and CDMA. What this MAC technique seeks to do is to move any possible collisions to the request phase and assign channel resources in the data phase to those terminals who have packets to send. This MAC scheme results in a system wherein the time varying bandwidth needs of any terminal can be accommodated. The only time insufficient bandwidth allocated is when several terminals are producing traffic at close to their peak rates. When this occurs, some terminals may not get all of the channel bandwidth they requested. Because of this possibility, the QoS guarantees, in terms of delay bounds, are not as strong as the fixed bandwidth allocation case. In considering DAMA, one needs to determine the distribution of packet sizes that will be transmitted in the uplink channel. Demand assigned multiple access works well for bursty sources only if the sizes of the data packets far exceed the size of the DAMA request packets. If this is not the case, then DAMA performs no better than random access.

3.2.2 The Author's Design Methodology for Medium Access Control

In the spring of 1997, during an invited talk organized by the UCLA's Computer Science Distinguished Lecture Series, Professor Abramson stated that in terms of multiple access, "You just cannot do better than ALOHA." In a limited sense, this statement is true. If the sources (applications) are generating traffic (packets) whose interarrival times are exponentially distributed and certain conditions exist at the Physical Layer, then ALOHA is optimal. However, as has been repeatedly shown in study after study, nothing in the Internet, neither at its core nor at its edges, is Poisson in nature [41, 47, 55, 6, 54]. With this in mind the following methodology will be put forth as to how to do optimal medium access control design. When attempting to develop a medium access control protocol for the MAC sub-layer of any communication system three things must be taken into account : the nature of the traffic (i.e. packet flow) that an application produces, the effect that the Transport Layer has on this flow of packets, and the constraints imposed by the Physical Layer on packet delivery.

Since this is stated in a rather abstract manner, some examples will be given so as to clarify.

- **Application Example:** A constant bit rate (CBR) application is not well served by a ALOHA MAC sub-layer since packets are arriving in a deterministic manner. A simple TDMA MAC sub-layer is appropriate.

- **Physical Layer Example:** In a high symbol rate indoor radio system, where intersymbol interference (ISI) is severe, adaptive algorithms (i.e. adaptive equalizers, adaptive antenna arrays) are required at the Physical Layer. This imposes the Physical Layer constraint that many known channel symbols must be inserted at the beginning of every transmitted packet. If the application using this radio system transmits very small packets, then an ALOHA MAC sub-layer will lead to very poor spectral utilization, due to the per packet training symbol overhead. A MAC method where these packets are accumulated and transmitted as a "bundle" of packets, with a single training period would be appropriate.

- **Application / Physical Layer Example:** An application, operating over a high latency link (for instance a GEO satellite channel), which generates delay, but not loss, sensitive packets would not be well served by a DAMA based MAC sub-layer, due to the DAMA metric "handshake". An ALOHA scheme without acknowledgement would be appropriate.

- **Transport Layer Example:** A bulk transfer application, using TCP as the Transport Layer protocol requires reliable delivery of TCP *acknowledgements* in order to increase the TCP window size, therefore increasing throughput and minimizing transfer time. A random access MAC sub-layer which cannot deliver packets reliably would not server this application well, rather a DAMA technique would be appropriate.

3.3 Medium Access Control in Satellite Communications

In satellite networks, MAC protocols must be in place for the uplink (earth terminal to satellite) channel. Since the uplink is a shared channel, the MAC protocol can have a significant effect on the QoS the network is capable of delivering. The main characteristic that distinguishes satellite networks from terrestrial networks is in transmission delay (sometimes referred to as latency). With random access, this delay makes for a long time between burst transmissions and acknowledgements. PRMA and similar techniques that require instantaneous acknowledgements are rendered unworkable. Satellite systems impose additional challenges to DAMA techniques also. Due to the high latency, passing DAMA request information between an earth terminal and the satellite is a very slow process. Let D represent the one hop delay from earth terminal to satellite. For GEO systems, D is on the order of 135 milliseconds. If the allocating agent is located at the satellite, the time between a packet's arrival at the terminal output queue and its actual start of transmission on the uplink channel is lower bounded by 2D. If the allocating agent is located at the terrestrial MCS, this time is now lower bounded by 4D. Since minimizing end to end delay is crucial in satellite systems, the link utilization benefits of DAMA bring with it a substantial increase in end-to-end delay. More detail on the variants of these various MAC techniques will be expounded upon in subsequent chapters as will be required for comparison purposes. For the purposes of this work, we shall assume that the satellite is capable of *on-board decisions* (i.e. switching, scheduling, etc.) and is not simply a "bent pipe". It is in this context that our MAC ideas take shape.

Before developing a medium access control methodology, the potential *system scenarios* must be established. By system scenarios, we mean the degree of traffic aggregation present at each earth terminal within a satellite's spot beam. Figure 3.2 illustrates this. We shall refer to all terrestrial stations that communicate with a satellite as *earth terminals*, even though the QoS requirements and the volume of packets transmitted could be vastly different. Figure 3.2 indicates that all of these earth terminals could be sharing the uplink channel, but the means by which they access it will differ by the QoS contract and the nature of the traffic which each terminal is offering to the satellite. We shall refer to *source traffic* as the packets that are transmitted by the earth terminal over the uplink channel to the satellite. We refer to these packets as source traffic since they are entering the satellite network from a particular earth terminal. We shall refer to the earth terminal as the *effective source*. The source traffic leaving an effective source many times may not be originating from the same *actual source*, since it is possible that many packet sources (i.e. computers, tele/video phones, etc) may be attached to one earth terminal. The number of actual sources (i.e. end users) generating the packets will vary among types of earth terminals in the satellite network. For instance if the earth terminal is a node on an Internet backbone, the number of actual sources that are in some way connected to the backbone is very large whereas in the end user scenario, the number of actual sources attached to the earth terminal could conceivably be only one. As shown in Figure 3.3, this can be modeled as a single effective source attached to an earth terminal. The probability distribution function (PDF) of packet sizes and inter arrival times would be what actually describes the traffic characteristics of the effective source. This PDF represents the combined statistics of all the actual sources connected to the earth terminal.

3.3.1 Very High Source Aggregation Scenario

If a satellite link is used as an Internet Backbone, it will most likely be connecting two continents, or at least two points where a wire line connection is not feasible. This type

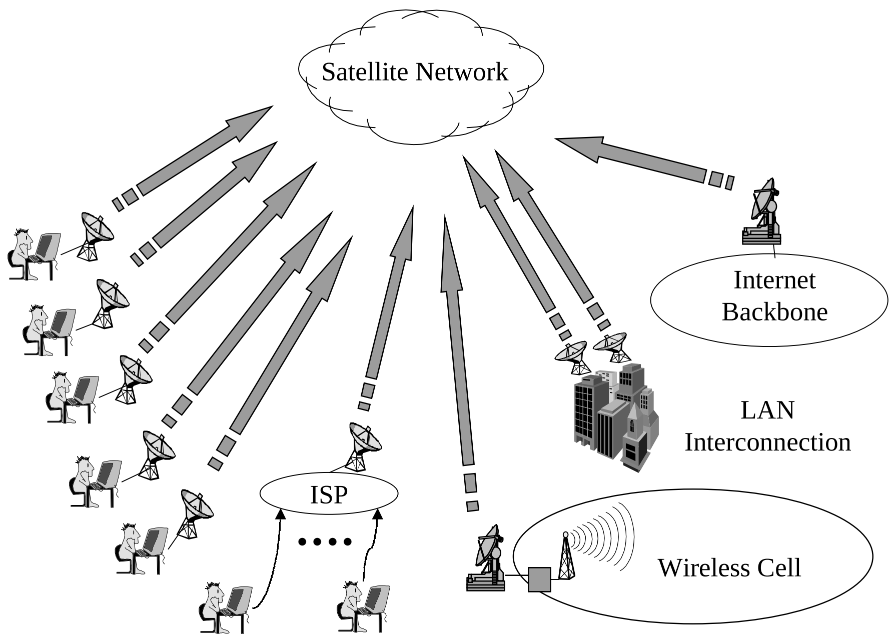

Figure 3.2: System Environment

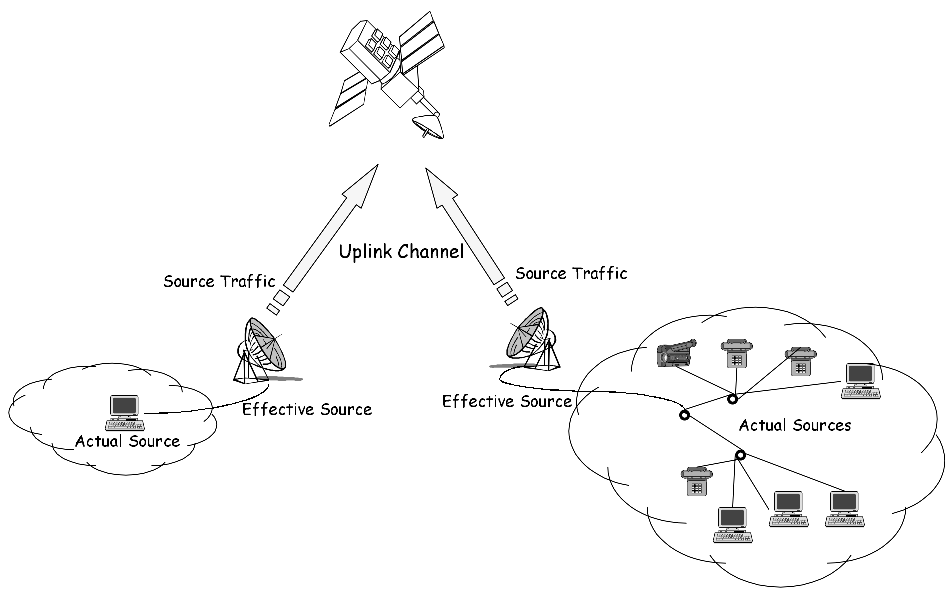

Figure 3.3: Earth Terminal Source Model

of connection will require a high bandwidth uplink, with latency close as possible to the propagation time. The offered traffic load will be an aggregation of a seemingly infinite number of independent actual sources. At short time scales this traffic will always be bursty but since there is a high degree of statistical multiplexing, the traffic may be become sufficiently smooth at larger time scales. The most appropriate MAC solution will depend upon the delay requirements, the smoothness of the traffic, and the price the backbone provider is willing to pay. If delay is critical then the most appropriate method of medium access would be a dedicated channel. The total bandwidth allocated could be chosen either to match that of the terrestrial links or based upon an estimate of the peak traffic load. Using a fixed bandwidth connection for bursty source traffic will always lead to some loss in system capacity, but if there is enough aggregation at the uplink point, the source traffic will be sufficiently smooth such that it will efficiently utilize a fixed bandwidth link (we are not disputing claims that Internet traffic is *self-similar* [16, 17, 55], just that it will efficiently use a dedicated link when considering large time scales). If required delays are close to what a DAMA scheme will deliver or if the backbone provider cannot afford to waste bandwidth, then demand assigned techniques can be employed.

3.3.2 High Source Aggregation Scenario

A company with two or more dispersed campuses may use a satellite link to connect their local area networks (LAN). In this case the desired bit rates will be in the 10 - 100 Mb/s range. Since the number of sources will be considerably smaller than the backbone scenario, the source traffic will be very bursty. Another situation similar to this is an Internet Service Provider's (ISP) connection to the terrestrial Internet. The similarity arises in that the degree of aggregation will be in the same range. The actual traffic statistics may differ significantly because the applications common to companies interconnecting LANs are different from the applications running over an ISP, which tend to connect households to the Internet. As in the previous very high aggregation case, the choice of medium access technique will depend on the delay requirements and pricing. However in this scenario, the burstiness of the source traffic will make fixed bandwidth connections very inefficient from a system capacity standpoint.

3.3.3 Medium Source Aggregation Scenario

When the bandwidth demands of the source traffic drop below 10 Mb/s, DAMA based MAC protocols become feasible for both GEO and non-GSO systems. In a wireless scenario, where a satellite link is used to connect a wireless cell with a wide area network backbone, the offered source traffic will always be less than a LAN interconnect, even for the same mix of source applications. This is due to limitations in the wireless channel which cause all applications to run at bandwidths significantly less than their wire line counterparts. In a scenario where end users are directly connecting to an earth terminal, the bandwidths can potentially be very low and the burstiness very high. Currently, individuals can purchase a relatively small satellite dish capable of two way communication with a GEO satellite. In future non-GSO systems, these dishes can be made even smaller. These can be used by individual households, companies, universities, and many other “users” who do not have access to a broadband wired infrastructure. This brings on the specter of many low bit-rate terminals sharing a common uplink channel. We refer to this scenario as a “personal earth terminal” model. The personal earth terminal model scenario will be the environment where the MAC protocols detailed in Chapters 4 and 5 are primarily intended to operate. In this case, the MAC protocol will play an important role in determining what QoS the end users will see and what the capacity of the uplink channel will be.

3.4 Simulation Tools for Protocol Evaluation

In order to test and verify any MAC design, a simulation tool is required. Our requirements were to have a simulation test-bed which had popular Transport Layer protocols, such as TCP, UDP, etc., already implemented and the ability to be modified, so as to allow for satellite network elements. We chose to build our satellite simulation environment upon the discrete event simulator **ns** [35]. **ns**, developed at the University of California, Berkeley, is a discrete event simulator primarily used in Internet research. Since this is non-commercial software, we were able to add on additional network elements, so as to produce a satellite simulation test-bed. We have verified our satellite MAC simulator by implementing and evaluating two previously proposed MAC protocols [4] [21] developed for VBR traffic over GEO satellite systems. The results of these simulations appeared in [11]. In this section we will describe in some detail the new satellite network elements that were added to **ns** and their functionality. This will be followed by an explanation of how these elements are assembled to make a working emulated network and how these elements are extended to add additional functionality.

ns is a discrete event simulator initially developed to simulate the performance of queueing disciplines (then called the REAL simulator). Since then it has undergone many changes and improvements and is now one of the primary simulation tools of the Internet research community. Version 2.0, which is the version our additions are built into, uses C++ as the primary language for creating new network elements and uses Tcl, the tool command language [39] as a means of creating simulation scripts. **ns** creates a network topology by declaring basic network elements, created in C++, and "connecting them", using Tcl, to generate compound network elements. For example, a link connecting two IP nodes would be generated by choosing the type of queue desired and connecting it to a delay object. This compound object then has its parameters (such as latency and bandwidth) set, so as to model the desired link. All network elements in **ns** are designed to use C++ object oriented features, primarily inheritance. This allows for a simple base class, and more complicated derived classes. This is helpful for this type of simulator since most research ideas are built upon improvements to basic network elements, such as transport protocols, routers, and queues.

To simulate uplink channel MAC protocols, new network elements were needed. Since *ns* is primarily a Network Layer simulator (layer 3 and above), many new Data Link Control (DLC) and Physical Layer (PHY) elements had to be generated. The following is a list of satellite network elements and their primary function.

TDM Interface: Since the underlying PHY protocol would be time division multiple access (TDMA), there needed to exist a means of synchronously switching cells onto a TDMA link. Here we make a distinction between *packets* and *cells*. We consider packets to be those datagrams of information passed into the DLC from the Network Layer. Sometime these will be referred to as *Network Layer Packets*. Cells are the fixed length basic unit of information that is transmitted over the wireless channel. Packets are segmented (fragmented) to form one or more cells. Since packets are often variable length, the number of cells produced from a packet will vary also. These cells are switched at synchronous instants in time. This represented a significant departure from the conventional **ns** scenario, where a packet arriving to a queue was the event that instigated transmission on a link. In a time division multiplexed (TDM) link, cells are placed on a link at synchronous instances in time. Thus a packet could arrive at a link and have to wait in the queue until the next synchronous instance. The TDM Interface generated for our purposes can be though of as a switch that synchronously switches cells from output queues to slots in the TDM channel. The exact mapping that the TDM Interface follows in performing this switching is given to it at fixed intervals called frames. The originator of this mapping is a TDM Controller, to be described next. Thus every frame, the TDM Interface receives a mapping plan which it follows in moving cells from queues to TDM slots throughout the entire frame.

TDM Controller: This element along with its derived classes make up the heart of satellite MAC implementation. The TDM Controller generates the mapping from output queues to

TDM slots. The method it follows in generating this mapping is the essence of the MAC protocol. In most cases the mapping generated by the TDM controller at the earth terminal is generated in conjunction with the TDM Controller at the satellite. Another role of the TDM Controller is in measuring one or more metrics from the output queues. The ability to monitor the output queues allows for generation of request information. This is essential for many request-based MAC protocols.

Satellite Queue: Since queues are an essential part of any network simulator, **ns** comes with queues of many different flavors. We needed some additional functionality not already present in **ns**, namely fragmentation. Fragmentation is needed when the packet size at the Network Layer does not match the cell size at the Physical Layer. This fragmentation is best accomplished at the output queues immediately before the TDM Interface. When a packet is queued at the Satellite Queue, it is queued as a Network Layer packet. When the TDM Interface dequeues it for transport over the TDM link, it is dequeued as a Physical Layer cell. Since **ns** has a queue base class, our Satellite Queue is simply a derived class with our additions.

Defragmentation Queue: Since fragmentation is occurring at the transmitter, defragmentation must occur at the receiver. Defragmenatation Queues simply collect PHY cells and hold them until the entire packet it received. Once this occurs, the original Network Layer packet is forwarded on. If some cells of a Network Layer packet are lost at the Physical Layer, due to collisions, then the packet can never be forwarded on. To accomplish this, the Defragmentation Queue periodically inspects its queue. If it finds a cell that has been sitting in the queue beyond a fixed amount of time, it drops it from the queue. This has the effect of gradually purging packets from the Defragmentation Queue that will never be reconstructed.

Traffic Type Classifier: Routers are another essential ingredient in a network simulator. In **ns**, routers are referred to as classifiers. For our satellite network simulation, we needed a classifier that would route packets based upon the traffic type. We implemented a derived class Traffic Type Classifier that had 6 categories of packets; constant bit rate (CBR), real-time variable bit rate (rt-VBR), non-real-time VBR (nrt-VBR), unspecified bit rate (UBR), available bit rate (ABR), and none of the above (NOTA). This breakdown follows the service classification of ATM networks. Our MAC work is independent of the Network Layer protocols. However we chose the ATM convention because it was comprehensive.

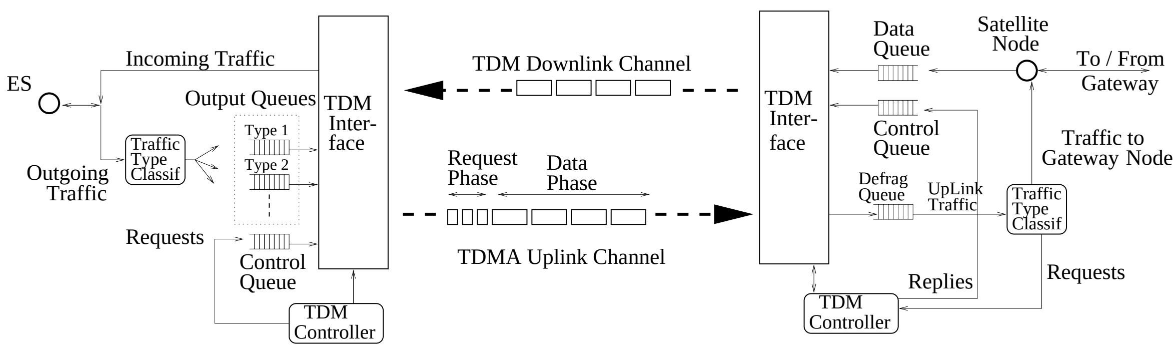

Figure 3.4: Block diagram of a generic satellite MAC protocol implementation in ns-2.0.

Figure 3.4 shows how these network elements are integrated to make a working simulation. Figure 3.4 shows a duplex link between an earth terminal and a satellite. Keep in mind that there are several earth terminals communicating with one satellite, so in each simulation, there are as many duplex links as there are earth terminals. To avoid confusion, all of our satellite network elements exist at the “satellite layer”. The layers above the satellite layer will simply be called the Network Layer. Packets arrive from the Network Layer to the satellite layer and enter the Traffic Type Classifier (TTC). These packets are routed to the appropriate Satellite Queue based upon the packet type. The TDM Controller is constantly monitoring the queues. Most MAC protocols depend upon some request metric to be transmitted to the TDM Controller at the satellite. The TDM Controller then, in a unique MAC dependent manner, generates the mapping for the TDM Interface to follow when switching cells to the outgoing link. This mapping is passed to the TDM Interface and cells are moved to the outgoing link. When packets are de-queued for transport, they are fragmented to the size of the PHY cells. When these cells arrive at the satellite, they are passed through the satellite’s TDM Interface and then placed in the Defragmentation Queue. In this queue the cells wait until all of the fragments have been successfully received. The original Network Layer packet is then forwarded to a Traffic Type Classifier at the satellite. This classifier then routes the packets, either to the downlink or to the TDM Controller at the satellite. Any sort of data packet will be routed to the downlink. The only type of packet routed to the TDM Classifier would be a request packet generated by an earth terminal’s TDM Controller. In this case the packet would be considered as “none of the above” (NOTA). The satellite TDM Controller would process these requests and then produce a response that would be transmitted in the downlink to the earth terminals. In this manner, any form of satellite MAC protocol can be implemented. Since the essence of the MAC is implemented in the TDM Controller, each MAC protocol we implemented was realized by generating a derived class of the base class TDM Controller. This allowed us to keep many of the same functions, while adding new features as needed by any particular MAC.

Chapter 4

A MAC Protocol for Streaming Media over Satellite Channels

In Chapter 1 we laid out our goal of developing a suite of medium access control protocols which would facilitate the transmission and reception of common Internet application data for terrestrial users in a personal earth terminal environment. This chapter presents a new hybrid demand assigned multiple access (DAMA) medium access control protocol which is designed to deliver real-time content over a GEO satellite uplink channel. Our proposed protocol uses a DAMA request metric which attempts to measure the slow time behavior [25] of source coded video. This request metric along with a dual random access (RA) / DAMA channel make for a MAC technique which seeks to solve the delay / capacity trade-off by introducing slight packet loss. Some real-time multimedia applications (voice and video) can tolerate light to moderate packet loss. Take for instance the video conferencing tools running over the Internet MBone. The MBone [31] is a virtual network running over the physical Internet and it serves as a test bed for delivering multimedia services to many users via IP multicast. Currently the Internet provides only *best effort* delivery of packets which leads to no QoS guarantees. This causes current real-time MBone applications to see heavy packet loss under certain congestion conditions. We will define at this point *medium quality video* to be interactive applications which produce traffic at average rates less than 500 kb/s and can tolerate packet loss of 5 percent or less. This definition describes most interactive applications currently using the MBone.

This chapter will be organized as follows. Section 4.1 will present some hybrid DAMA techniques for real-time uplink multiple access. This will be followed by Section 4.2 which explains the basis for our RA/DAMA MAC. Section 4.3 provides some analysis of source coded video traffic and describes our methodology for tracking its slow time behavior. Section 4.4 describes in detail the RA/DAMA MAC. Section 4.5 will present some simulation work done on RA/DAMA using the extendable simulation tool built upon ***ns***, previously described in Section 3.4. Section 4.6 will conclude and outline some of the limitations and open integration issues for our RA/DAMA technique. These open issues will be address in the final chapter.

4.1 Hybrid DAMA Techniques for Real Time Sources

Calling the demand assigned multiple access of Chapter 3, Section 3.2.1 "pure-DAMA", hybrid DAMA techniques were developed to improve on the delay performance of pure DAMA, without sacrificing the channel utilization benefits. Hybrid DAMA medium access schemes are MAC protocols in which a portion of the bandwidth is allocated in a demand assigned manner [29, 21, 28, 10]. From now on we will be considering DAMA protocols in the context of time division multiplexing (TDM) although many of the techniques described here can be applied to other Physical Layer access methods. As stated in Section 3.3, packets crossing DAMA systems capable of on-board switching will wait in the output queue of the earth terminal a minimum of 2D. This implies that the time to cross the satellite portion of the end-to-end path is lower bounded by 4D. Therefore this 4D delay is the mark by which hybrids of DAMA must be measured. When considering real-time sources, this 4D lower bound (on the order of 540 milliseconds for GEO systems) can prove to be prohibitive in providing suitable QoS. Real time sources for the purposes of this paper will consist of source coded video. Let us now define a *sustained stream* to be the term that describes source coded variable bit-rate (VBR) video. This means that the video source has an average bit rate and the instantaneous bit rate deviates about this average. The bit rate never drops to zero. Recognizing this, Hung *et al* [21] proposed a hybrid DAMA MAC. Let B_T represent the total uplink bandwidth and N_{ET} be the number of earth terminals with ongoing sustained stream connections. In their technique, each earth terminal is allocated in every TDMA uplink frame, a fixed bandwidth (slots) ρ^{fx}. The bandwidth that remains, $B - N_{ET} \cdot \rho^{fx}$ is made available on a demand assigned basis. In each TDMA frame, control slots are set aside for each earth terminal to transmit its output queue size. Using this information, the allocating agent at the satellite assigns the remaining $B - N_{ET} \cdot \rho^{fx}$ bandwidth. This technique can be thought of as a hybrid fixed bandwidth allocation and demand assigned multiple access or FBA/DAMA. This scheme seeks to exploit the fact that the bit rate of a sustained stream never drops to zero. FBA/DAMA was shown to outperform DAMA and FBA under certain loading conditions [11]. Two drawbacks to this technique exist. First, continually measuring and sending the output queue size every TDMA frame requires constant processing at both the satellite and the earth terminal. Transmitting this queue size metric each frame in mini-slots wastes uplink bandwidth. Second, and more importantly, the DAMA part of this technique attempts to track the instantaneous queue size, which changes on a rapid time scale. This can be viewed as a control system with the output queue size as the input signal and the feedback loop being the reverse link DAMA bandwidth allocation (see Figure 4.1). In a GEO scenario, the loop delay is 2D, making accurate tracking of the instantaneous queue size inaccurate. Recent research indicates that source coded video, in this case MPEG [25], can be thought of as a slow time process modulating a fast time process. The slow time process (STP) can be thought of as the mean information rate of a particular scene in a video sequence. The authors in [25] indicate that this STP changes on an order slower than 2D. The faster time scale process is the oscillation about the STP and represents the variability of each frame of video.

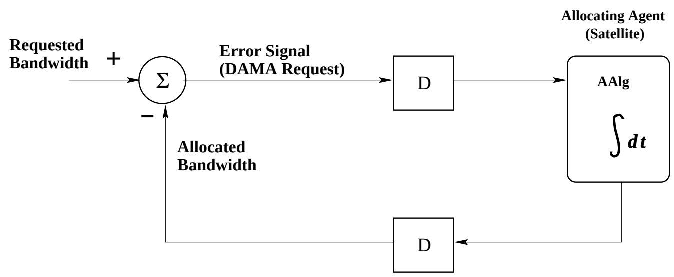

Figure 4.1: Control System Representation of a DAMA System

4.2 Combined Random Access / Demand Assigned Multiple Access

This slow time / fast time behavior is the motivation for our hybrid random access / demand assigned multiple access (RA/DAMA) technique. If the STP can be tracked and its value used for the request metric, then a slow time-changing amount of DAMA bandwidth can be allocated to each terminal that follows the slow time behavior of each source. This makes up the DAMA portion of the RA/DAMA technique. The signaling required for this DAMA bandwidth will occur at the rate that the STP changes, rather than the TMDA frame size. This will result in a reduction in the amount of bandwidth required for DAMA signaling. The means by which this slow time behavior is tracked is described in section 4.3. In order to monitor the STP, an estimate of the instantaneous bit rate must be formed. This will inevitably require some time window. During this time the video source could potentially go through a *scene change*. A scene change simply means that the STP “jumps” to a new value. If this scene change is in the direction of increasing bit rate, packets will begin to accumulate at the terminals output queue. Since a certain amount of time is required to form an estimate of the new STP value and a round trip time is needed to have this acknowledged by the allocating agent (satellite), packets produced during this time may see unacceptable end-to-end delays. To address this, the unused portion of the uplink TDMA frame is made available for random access transmission. Since packets will utilize the random access channel only during scene changes, collisions on the RA channel will only occur if scene changes occur simultaneously in independent uplink video sessions. If this probability is relatively low, then packet loss rates will be low as well. The exact methodology for transmitting packets over either the DAMA or RA channel is described in detail in section 4.4. Figure 4.2 shows a block diagram of an earth terminal employing this process. All packets arriving from the Network Layer are initially queued in the DAMA queue. A delay threshold is constantly being maintained at the terminal. When packets in the DAMA queue exceed this threshold packets are dequeued from the *head* of the DAMA queue and placed into the RA queue. As stated previously, the DAMA channel and the RA channel reside on the same TDMA uplink frame. Their exact starting points vary from frame to frame based upon demand. Allowing terminals to transmit packets in a random access manner introduces the possibility of packet loss. For some applications, i.e. computer data, this is unacceptable, but for some real time applications, light packet loss may be tolerable.

When compared to the video quality currently provided to real-time applications by the Internet MBone, this light packet loss may appear attractive. If so, then employing this hybrid RA/DAMA technique can bring about delay improvements over DAMA and FBA/DAMA and a substantial link utilization increase over FBA. Section 4.5 describes the measured performance benefits.

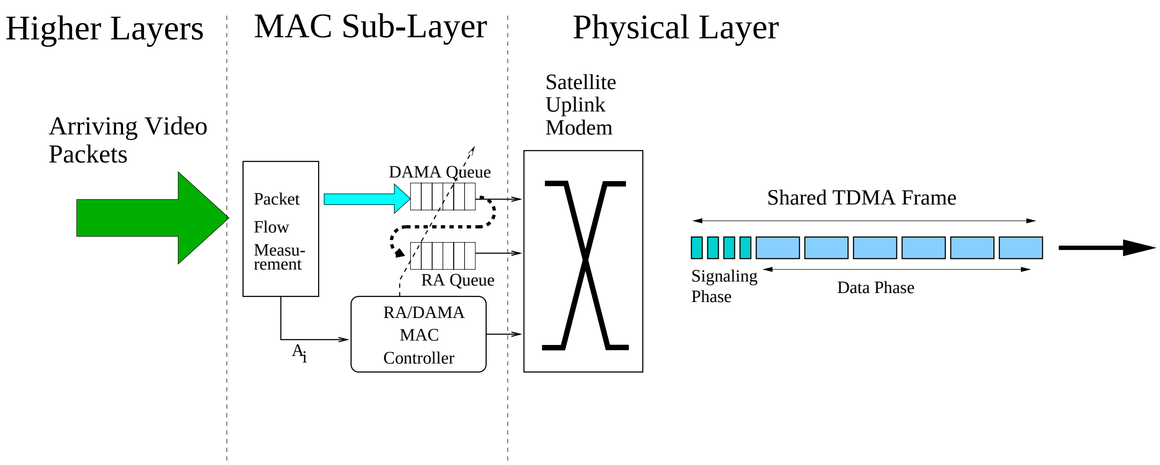

Figure 4.2: Block Diagram of a Communication System employing RA/DAMA

4.3 Bandwidth Request Metric for RA/DAMA

In order that the slow time scale process might be tracked and used for DAMA bandwidth allocation, a suitable metric must be formed that accurately measures it. The idea that the instantaneous bit rate of source coded video could be viewed as a two component process was conceived by Jelenkovic *et. al.* [25] when examining MPEG-1 coded video traffic.

They stated that the video stream at the group of pictures (GOP) level could be thought of as a sequence of scenes. A scene change occurs when a significant change occurs in the GOP size. Our focus is not an in-depth study of source coded video, rather we will use some of the techniques presented in [25] to extract statistics of the slow time process (STP) and the fast time process (FTP) in order to derive an appropriate metric for acquiring bandwidth on the DAMA channel. Jelenkovic *et. al.* states that video scenes change at a rate that is on the order of seconds (greater than 2D). Therefore reconsidering our control system analogy, a DAMA system which allocates bandwidth based upon the STP would track accurately. This also means that the DAMA signaling (requests) will be less frequent therefore leading to a reduction in the amount of bandwidth lost due to control.

Letting A_i be the number of bits received from a source over the period $\{i \cdot T_c, (i+1) \cdot T_c\}$, the authors in [25], used the following normalized second difference equation to determine when a scene change had occurred.

$$\frac{(A_{i+1} - A_i) - (A_i - A_{i-1})}{\frac{1}{25}\sum_{j=i-24}^{i} A_j} < \gamma \tag{4.1}$$

where:

$$A_j = \int_{\tau=j \cdot T_c}^{(j+1) \cdot T_c} arriving\_bits(\tau) d\tau \tag{4.2}$$

T_c is referred to as the *collection window*. It is the interval over which the size (in bits) of each arriving video frame is accumulated to form A_i. Gamma (γ) is an threshold constant, determined empirically to be -0.5. This parameter γ is the value which, in the authors of [25] opinion, determines when a video scene change has occurred. Figures 4.3 and 4.4 show the collected process A_i, for an H.261 encoded video trace and an MPEG-1 trace respectively. The H.261 trace came from CNN's Headline News and was produced using the video conferencing tool *vic* [36]. *vic* has three parameters to control the bandwidth and quality: **b** (bandwidth), **f** (frame rate), and **q** (quality). The trace shown in figure 4.3 was obtained using b=384 kb/s, f=10 frames/s, and q=10, which yielded reasonable video quality. The MPEG-1 trace is from the movie *Terminator* and was obtained form the Institute of Computer Science at the University of Würzburg [46]. The MPEG sequence has a GOP size of twelve frames that follows the pattern IBBPBBPBBPBB. The peak and mean bit rates are 740 kb/s and 270 kb/s respectively. Both of these video traces fit the definition of what we would consider to be medium quality video. It would appear, at least empirically that A_i has a slow time or low frequency process associated with it in both cases. This slow time process varies on the order of hundreds of milliseconds, in excess of a round trip time. Therefore if this process can be tracked, its value can then be communicated to the allocating agent for DAMA bandwidth allocation.

Figure 4.5 shows the *scene process*, $\{S_i, i \geq 0\}$ and *collected process*, A_i, from the H.261 encoded trace of CNN's Headline News (CNN-HN). The process S_i is generated by taking the average value of A_i between two consecutive points in the point process resulting from Equation 4.1.

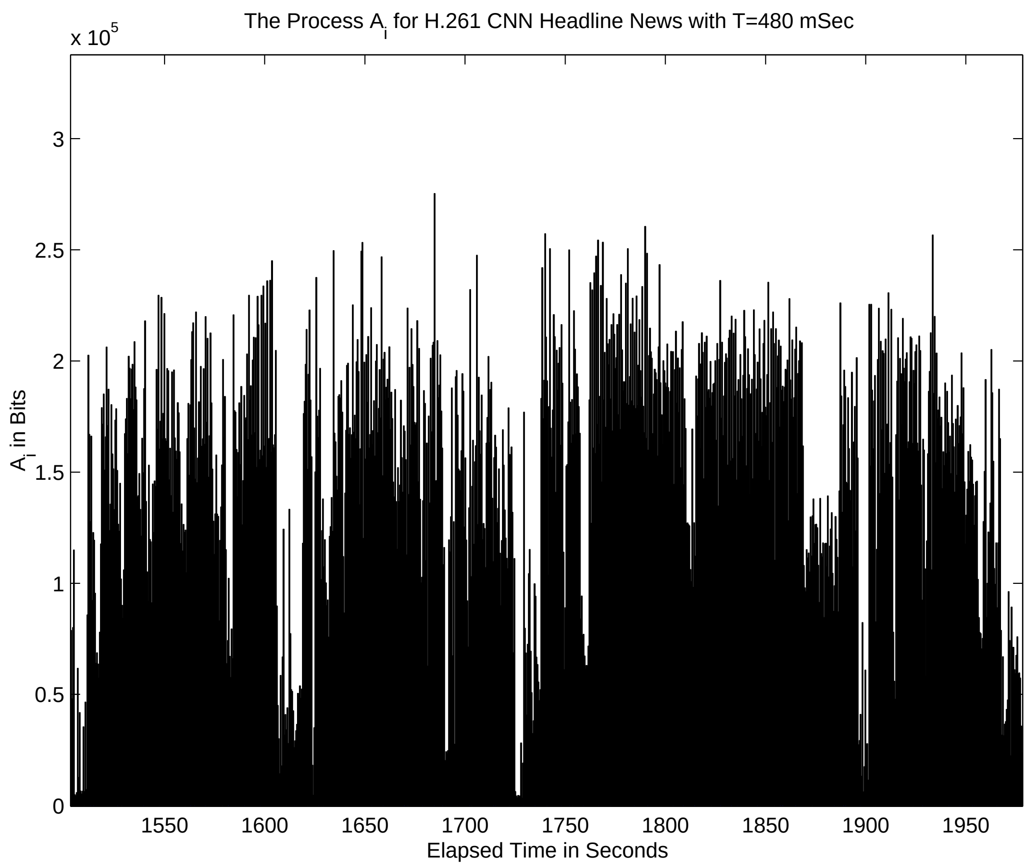

Figure 4.3: The Collected Process A_i for H.261 CNN-HN

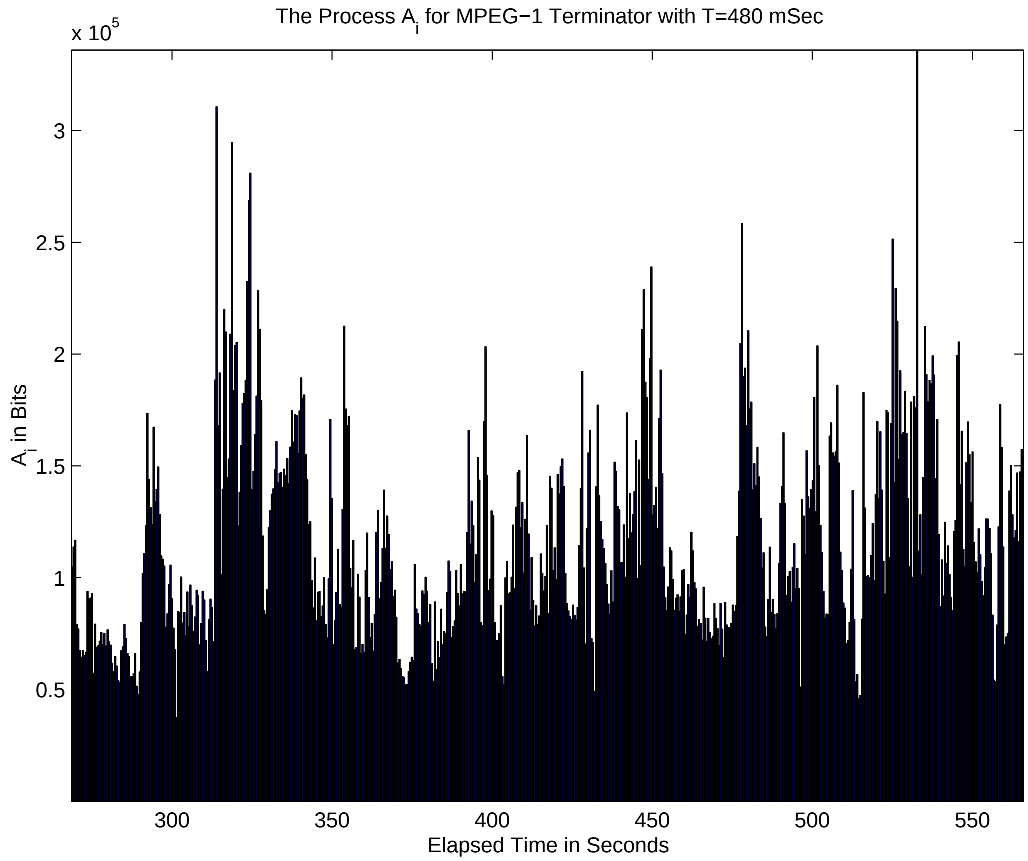

Figure 4.4: The Collected Process A_i for Terminator, MPEG-1

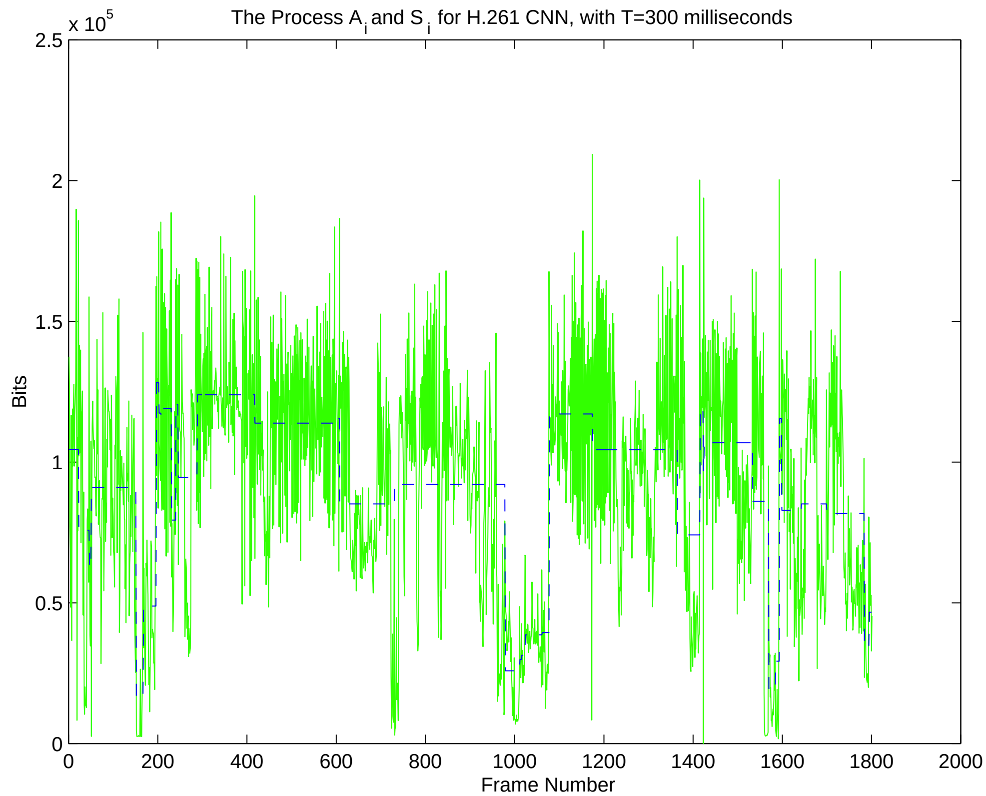

Figure 4.5: Collected (solid) and Scene (dashed) Process

In forming an estimate of the STP, there are two competing interests. We would like the estimate to be accurate, however the more accurate the estimate, the more samples are required. The time to from the estimate is also of great importance. The DAMA bandwidth is apportioned so as to track the slow time average, which changes as the scenes change. If a scene change occurs such that the bit rate increases, an excessively long estimation period will lead to excessive packet delays.

Using this process A_i the RA/DAMA request metric, or *packet flow rate metric* (PFRM) which seeks to track the STP, is the scalar,

$$M(i) = \sum_{j=0}^{w} A_{i-j} h_j \tag{4.3}$$

This metric is simply an inner product of the process A_i with a vector $\vec{h}$ of length w. Clearly $\vec{h}$ must be some form of low-pass or averaging filter. Since this vector is arbitrary, for simplicity we simply chose the unit step function $u(n) - u(n-w)$. Therefore our packet flow metric is a time windowed average of the collected process A_i. What remains is to find the proper values of for the time window w, and the collection interval T_c.

Figure 4.6 shows the standard deviation of the process $\{A_i, i \geq 0\}$ with various collection intervals for both the H.261 *vic* trace and the MPEG Terminator trace. The size of T_c should be chosen such that the process A_i accurately represents the instantaneous bit rate at the scene level. Figure 4.6 shows that when T_c is chosen less than $length_{GOP}$, the variability of A_i is due primarily to the size difference between I and P/B frames. The GOP size for the H.261 trace varies between 2 and 3 (i. e. IP or IPP) making $length_{GOP}$ either 200 or 300 milliseconds. For the MPEG trace the GOP size is always 12, making $length_{GOP}$ 480 milliseconds. Figure 4.6 show that for both traces, once T_c reaches $length_{GOP}$, the standard deviation of A_i has fallen by approximately 87%. This implies that for purposes of tracking the STP, T_c must be chosen in excess of $length_{GOP}$. In Figure 4.5, T_c was set to 300 milliseconds.

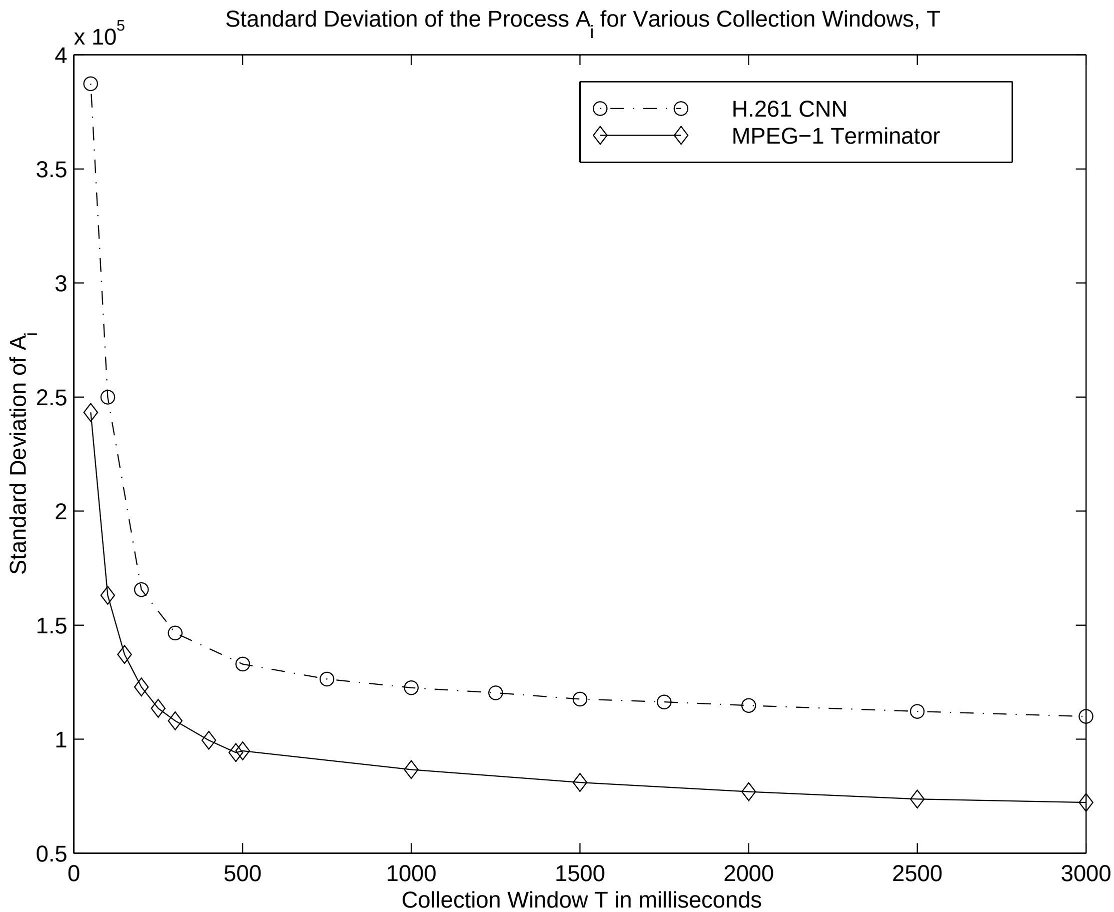

Figure 4.6: The Standard Deviation of the Process A_i

The best value for w is much less obvious. Referring back to Figure 4.5, we can imagine the process $\epsilon_i = S_i - A_i$ as "noise" riding on top of a low frequency signal S_i. Since our goal is to extract the process S_i and use it as our metric for DAMA bandwidth allocation, then our ability to estimate it will depend upon the variance of ϵ_i. As with T_c, w should be made small enough to make the request metric $M(i)$ respond quickly to scene changes, but long enough such that $M(i)$ tracks the STP. Since there is no known probability density function (PDF) associated with the process ϵ_i, there is no closed form expression relating the variance of $M(i)$ to the averaging filter length w. Therefore the variance of $M(i)$ was found using Monte Carlo techniques over the entire lengths of the CNN-HN trace and several MPEG-1 traces [46]. The MPEG sequences all have a GOP size of twelve frames that follows the pattern IBBPBBPBBPBB. The peak and mean bit rates vary from 270 - 2,710 kb/s and 150 - 490 kb/s respectively. Figure 4.7 shows the standard deviations of $M(i)$ as a function of w for all traces. The difference in standard deviation between the two coding techniques is due primarily to the means by which the video sequences were encoded. The MPEG sequences have frames in a fixed pattern arriving at fixed intervals of 40 milliseconds. When *vic* encodes the CNN sequence, the actual pattern (either IP or IPP) and the interframe spacing vary, depending upon the three parameters **b**,**f**, and **q**. This figure shows that for both traces the sharpest drop in standard deviation occurs before $w = 5$.

As stated previously, our simulations will be conducted using an actual trace as input, as opposed to a parametric traffic generator. We chose the CNN *vic* trace over the MPEG

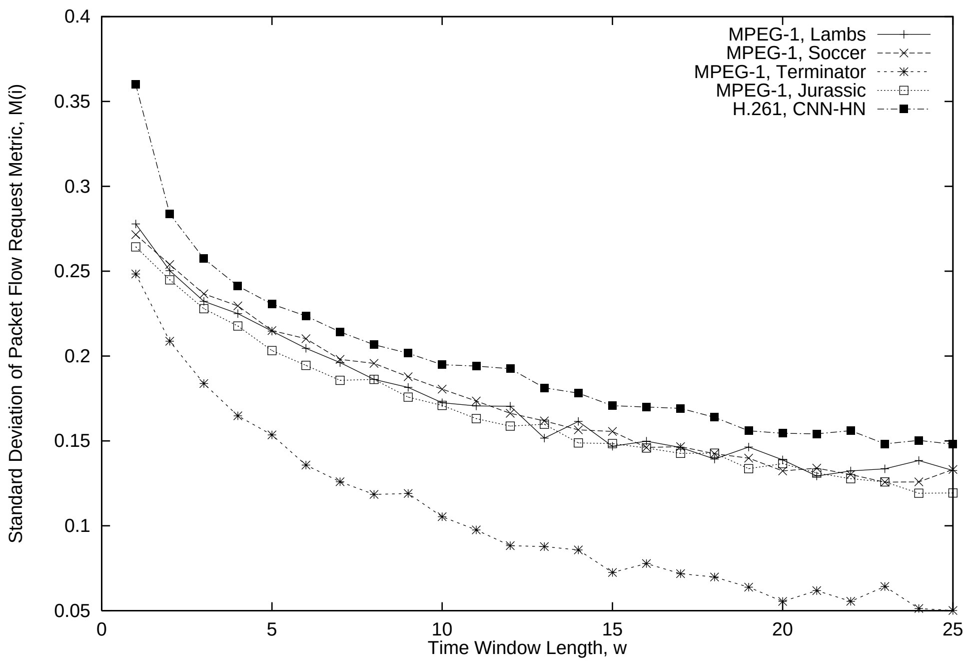

Figure 4.7: Standard Deviations as a Function of *w*

traces to drive our simulations since the video content is a diversity of many scenes of video, from action news footage to a “talking head” news anchor. For this trace we will set w equal to 4 and set T_c to be the maximum GOP length of 300 milliseconds. Also since the variance of $M(i)$ is higher for the H.261 trace, it will be the more challenging sequence to track.

4.4 Packet Transport Methodology for RA/DAMA

As stated previously, RA/DAMA employs time division multiple access (TDMA) at the Physical Layer. All time slots are of equal length and the packet flow rate metric is transmitted in a piggy-backed manner within the data slots. This means there are no “mini-slots” needed. The random access channel and the DAMA channel are partitioned as follows. Let N be the number of slots per TDMA frame. The DAMA channel is the first L slots where:

$$L = \sum_{j=1}^{N_{ET}} B^j(i) \tag{4.4}$$

where

N_{ET} = number of Earth Terminals

$B^j(i)$ = the amount of bandwidth allocated to terminal j during the

i*th* frame

The RA channel is then simply the $N - L$ slots that remain in the current TDMA frame. This logical partition creates a TDMA physical access method where the boundary point between the RA and DAMA region changes. This is sometimes referred to as *dynamic TDMA*. Let T_f be the length of a TDMA *frame*. Since we are using the PFRM $M(i)$, derived in Section 4.3, for DAMA bandwidth acquisition, it must be transmitted to the allocating agent every collection interval, T_c. This imposes the restriction that $T_c \approx m \cdot T_f$, m an integer. This means that the length of a *superframe*, $T_{sf} = m \cdot T_f$, must be close to the collection interval for the metric to work efficiently. The exact location of the boundary between DAMA slots and RA slots is communicated to each earth terminal every superframe by the allocating agent. This is illustrated in Figure 4.8.

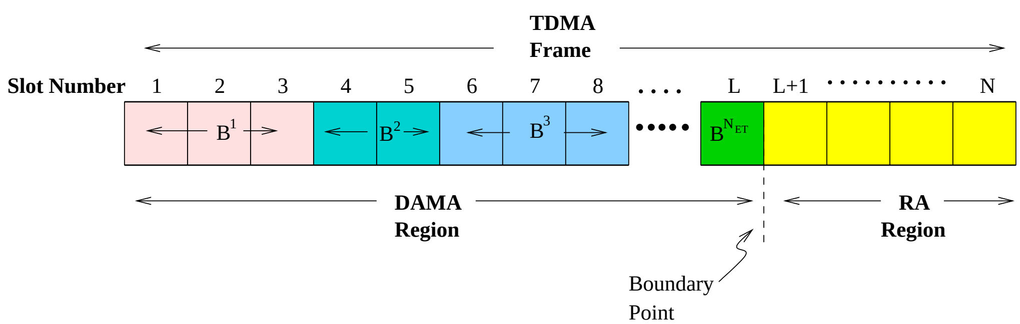

Figure 4.8: Partitioning of Channels for RA/DAMA

4.4.1 The Channel Selection Algorithm

The packet transport methodology consists of an algorithm for selecting the logical channel a given packet must cross. We refer to this as the *channel selection algorithm* (CSAlg). All packets arriving from the upper layers are time stamped and placed in the DAMA queue (DQ). These packets are considered Network Layer (***NL***) packets. When packets are finally transmitted over the TDMA channel, they are considered Physical Layer (***PHY***) cells. If the length of the ***NL*** packets are greater than the length of the ***PHY*** cells, then fragmentation will occur at the Physical Layer. The distinction between ***NL*** and ***PHY*** cells is necessary to understand the CSAlg. Referring to Figure 4.9, the CSAlg uses the PFRM and the size of the DAMA and random access queues (DQ and RAQ respectively) to form delay estimates of the last packet in each queue. If it appears, using these delay estimates, that the packets in the DQ will see delays in excess off a preset threshold, packets are removed from the front of the queue and placed in the random access queue (RAQ). When packets are moved from the DQ to the RAQ, they are dequeued as ***NL*** packets. This is necessary since if ***length(NL)*** $>$ ***length(PHY)***, then fragmentation will occur and we would prefer all fragments of a ***NL*** packet be transmitted over the same logical channel to reduce packet loss rates and to simplify the de-fragmentation process. This technique is described by the following pseudo code.

CSAlg

```
tdma_frame_start()
{
    compute:
        d_max^DAMA
        d_max^RA

    while( d_max^DAMA > D_T && d_max^RA < d_max^DAMA )
    {
        dequeue_nl_packet(queue = DQ, position = head)
        enqueue_nl_packet(queue = RAQ, position = tail)
        compute:
            d_max^DAMA
            d_max^RA

    }

}
```

Where:

d_{max}^{DAMA} is the delay estimate of the last packet in the DAMA queue
d_{max}^{RA} is the delay estimate of the last packet in the RA queue
D_T is an arbitrary delay threshold for the *CSAlg*

This algorithm seeks to limit the end-to-end delays to below a delay threshold D_T. The value D_T is an adjustable simulation parameter. Since there is also a constraint that $d_{max}^{RA} < d_{max}^{DAMA}$, there is no guarantee that packets will never experience a delay in excess of D_T. This algorithm causes packets to be moved from the DQ to the RAQ until either d_{max}^{DAMA} is less than D_T or until d_{max}^{RA} exceeds d_{max}^{DAMA}. This is an attempt to "balance" the delays seen by packets on each channel. It is important to note that the random access channel is *not* ARQ (automatic repeat request) based. All transmissions are done in a randomized manner that will be described in Section 4.4.3 and no attempt at retransmitting colliding packets is made. This makes the packets crossing the RA channel subject to losses, but makes the RA channel stable (i.e. not subject to congestion collapses).

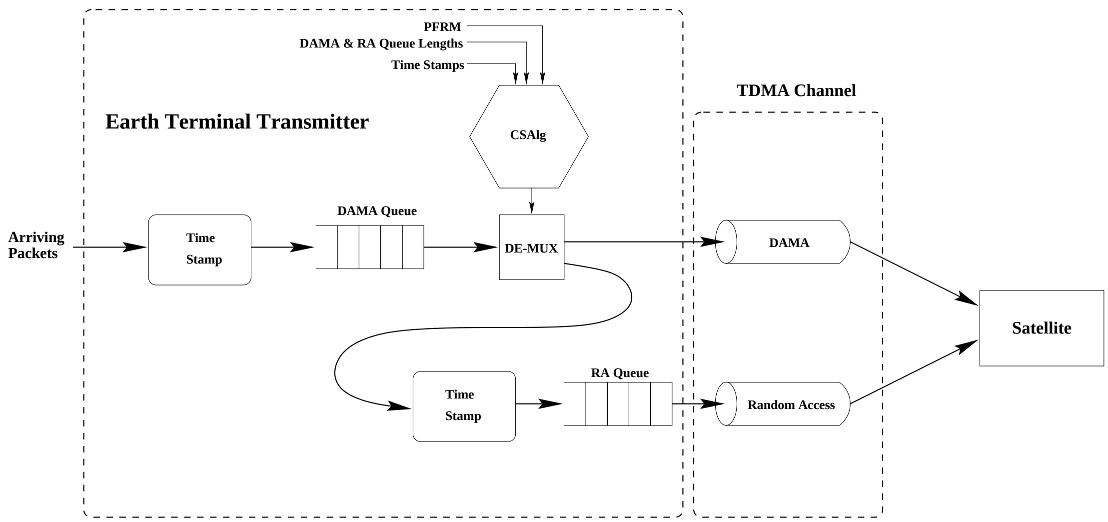

Figure 4.9: Block Diagram for ***CSAlg***

4.4.2 Technique for Generating Packet Delay Estimates

The CSAlg determines what channel packets will cross based upon the estimated delay. Therefore computing an accurate delay estimate is essential for the algorithm to work successfully. To accomplish this, let D represent the one hop delay from an earth terminal to the satellite and let T_f be the length of a single TDMA frame. Let:

$$z = \left\lceil \frac{2D}{T_f} \right\rceil \tag{4.5}$$

where $\lceil a \rceil$ equals the smallest integer greater than or equal to a.

The parameter z is the round trip quantized into TDMA frames. It represents the amount of time between a DAMA request for bandwidth and the actual time before which that requested bandwidth will be available. Let i be the number of the current TDMA frame, then $M(\lfloor \frac{i}{m} \rfloor)$, where m is the number of TDMA frames per superframe, equals the packet flow metric sent for the superframe containing i. M is measured in packets per second, but for the purposes of delay estimation, it is converted to ***PHY*** cells per frame using the following conversion:

$$M_{packets/frame} = \left\lceil \frac{M_{packets/sec}}{T_f} \right\rceil \tag{4.6}$$

Let $B(\lfloor \frac{i}{m} \rfloor)$ equal the DAMA bandwidth, also measured in ***PHY*** cells, apportioned to the earth terminal in the ith TDMA frame. Ideally $B(\lfloor \frac{i}{m} \rfloor)$ should be equal to $M(\lfloor \frac{i-z}{m} \rfloor - 1)$, if the exact amount of bandwidth requested was allocated. Then using these definitions, the estimate of the delay seen by the last packet in the DQ is:

$$d_{max}^{DAMA}(i) = \{\sum_{k=0}^{\infty} \mathcal{I}_{\{\sum_{j=0}^{k} \Theta(j,i) < \mathcal{L}(i)\}}\} \cdot T_f + 2D + W_{last\_packet}^{DAMA} \tag{4.7}$$

where:

$$\Theta(i,j) = \begin{cases} B(\lfloor \frac{i}{m} \rfloor) & if \quad j \leq i\text{mod}(m) \\ M(\lfloor \frac{i+j-z}{m} \rfloor - 1) & if \quad i\text{mod}(m) < j \leq z \\ M(\lfloor \frac{i}{m} \rfloor - 1) & if \quad j > z \end{cases} \tag{4.8}$$

and

$\mathcal{L}(i) =$ the length in ***PHY*** cells of the DQ at frame i

$\mathcal{I}$ is the Indicator Function

$W^{DAMA}_{last\_packet} =$ the waiting time of the last packet in the DQ

Equation (4.7) seeks to estimate the delay seen by the last packet in the queue by using the bandwidth allocated in the current superframe and the requests for bandwidth made in previous superframes. This is illustrated in Figure 4.10. $d^{DAMA}_{max}(i)$ is an estimate because there is no guarantee that $B(\lfloor \frac{i}{m} \rfloor)$ will equal $M(\lfloor \frac{i-z}{m} \rfloor - 1)$. However for light to moderate network loads, $E\{B(\lfloor \frac{i}{m} \rfloor)\} \approx E\{M(\lfloor \frac{i-z}{m} \rfloor - 1)\}$

The delay estimate for the last packet in the RAQ can be computed as:

$$d^{RA}_{max}(i) = \left\lceil \frac{\mathcal{L}^{RA}(i)}{|RA|_{packets}} \right\rceil * T_f + 2D + W^{RA}_{last\_packet} \tag{4.9}$$

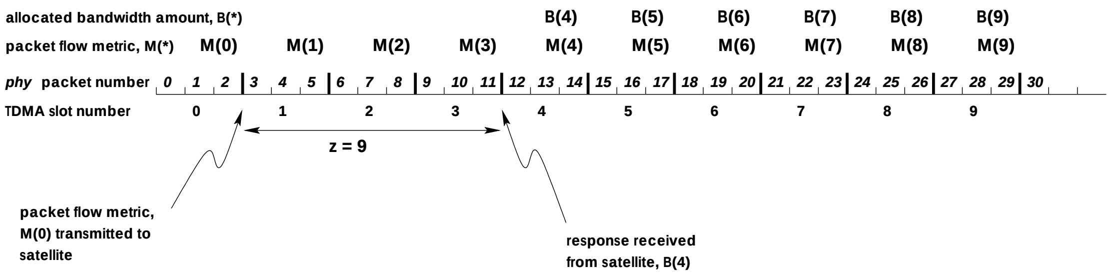

Figure 4.10: A Sample of Uplink Delay Estimation Process

where:

$\mathcal{L}^{RA}(i) =$ the length in ***PHY*** cells of the RAQ at the ith frame

$|RA|_{packets}$ $=$ the size (in ***PHY*** cells) of the RA region in the current TDMA frame

$W^{RA}_{last\_packet}$ $=$ the waiting time of the last packet in the RAQ

4.4.3 Miscellaneous MAC Sub-Layer Issues

Since this technique works on a frame-by-frame basis, the packet interarrival process will have a significant impact on the effectiveness of the CSAlg. The channel selection algorithm seeks to adapt to and track the slow time behavior of the video sequence. Therefore allowing for some smoothing, or shaping, of the input packet sequence may improve performance. This will be accomplished, for the simulations described in Section 4.5, by limiting the maximum flow rate (MFR) of packets arriving to the MAC sub-layer to a preset value.

The packets transmitted across the RA channel are subject to loss due to collisions. Collisions are possible due to the fact that the random access portion of the uplink channel is shared among all the earth terminals in the satellite's footprint. Since there is only coordination among the earth terminals about the allocation in the DAMA region, the only information available to each terminal concerning the RA region is the starting slot. The granularity of a collision is a TDMA slot. Referring to Figure 4.11, if there are N_p ***PHY*** cells per slot and the slot experiences a collision, all of these packets are lost. Compounding this loss is the fact that if only one ***PHY*** cell is lost, the entire ***NL*** packet is discarded and considered lost.

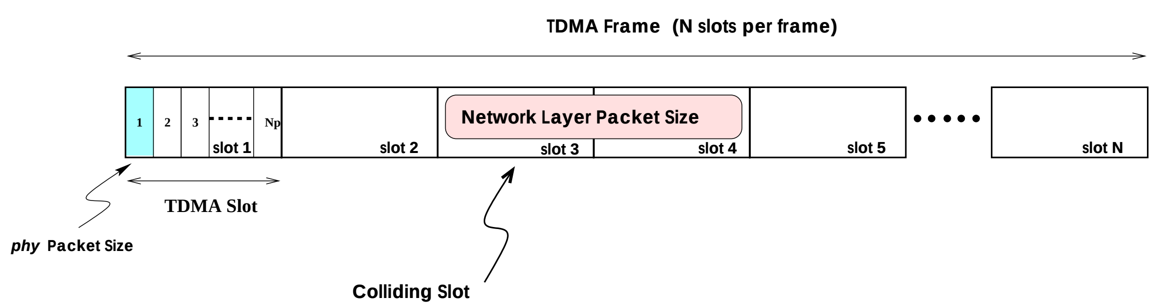

Figure 4.11: ***PHY*** and ***NL*** Packet Framing for TDMA

To mitigate the packet losses seen at the Network Layer, a random slot pattern is chosen per ***NL*** packet. Since packets are moved from the DQ to RAQ on a ***NL*** packet basis, random transmission patterns should remain unchanged until the entire ***NL*** packet has been transmitted. For light network loads, this amounts to a new slot pattern each TDMA frame. At higher loads however, the RA channel could be potentially small, leading to a transmission pattern which would stay constant over multiple frames (until the entire ***NL*** packet has been transmitted).

4.5 Simulation Results and Analysis

Since our source traffic is an actual video packet trace, analytical results for delay, loss, and delay variation are not possible. We have therefore resorted to extensive simulation of our MAC technique using an actual video trace. For the purposes of comparison we have conducted simulations not only on our RA/DAMA protocol, but also on FBA, DAMA, and the FBA/DAMA protocol proposed by Hung *et. al.*[21].

For the purposes of our simulation, we used a two hour long packet trace of CNN's Headline News, generated using *vic*. The mean bit rate of this trace was 292 kb/s and the maximum packet size was 1024 bytes. All packets produced from *vic* are Internet Protocol (IP) packets. The Network Layer packets mentioned earlier are in this case IP packets. This packet format includes a 20 byte IP header. For evaluating the RA/DAMA protocol, two maximum transmit unit (MTU) sizes were chosen; 512 and 53 bytes. Since we are using a *vic* trace, the packets generated by *vic* are on average close to 1024 bytes. However, MTU sizes of 512 are allowable for most Internet applications and we would like to see if a lower MTU leads to lower packet loss. This means that the average bit rate of the 512 byte

MTU trace is slightly higher than the 1024 byte MTU trace due to the replication of the IP header. Also to make our work relevant to ATM applications, we have chosen the standard 53 byte size as our other experimental MTU. The previously mentioned traffic shaping was incorporated into our RA/DAMA technique. To accomplish this, we simply used a "token bucket" [53] with a bucket depth of zero and a token rate equal to the maximum flow rate. Briefly, token bucket shaping is accomplished using a scheme illustrated in Figure 4.12. Tokens "fall" into a bucket of depth B at a rate R, when a packet arrives to the input of the shaping function, it cannot pass through until enough tokens are accumulated in the bucket to match the size of the arriving packet. If the bucket fills, then each new token that falls into it will be discarded. Making the bucket of depth zero and the token rate, R, equal to the desired maximum flow rate, has the effect of smoothing the incoming packet sequence such that the instantaneous bit rate never exceeds a preset maximum value. This value is one of the parameters of simulation.

Figure 4.13 shows a diagram of our simulation. Each earth terminal has a single video source attached to it. The uplink channel is shared among all terminals. The bandwidth of the uplink and downlink channels are the same. All cells transmitted in the uplink to the satellite are re-transmitted in the downlink to the gateway terminal. Once the cells are received at the gateway, they are reconstructed into ***NL*** packets and are split back into separate streams, copied and the replicated packet is passed through a traffic shaper. The purpose of the traffic shaping is to restore the maximum flow rate of each stream to its original value. When cells cross a Physical Layer, they can often experience a "clumping"

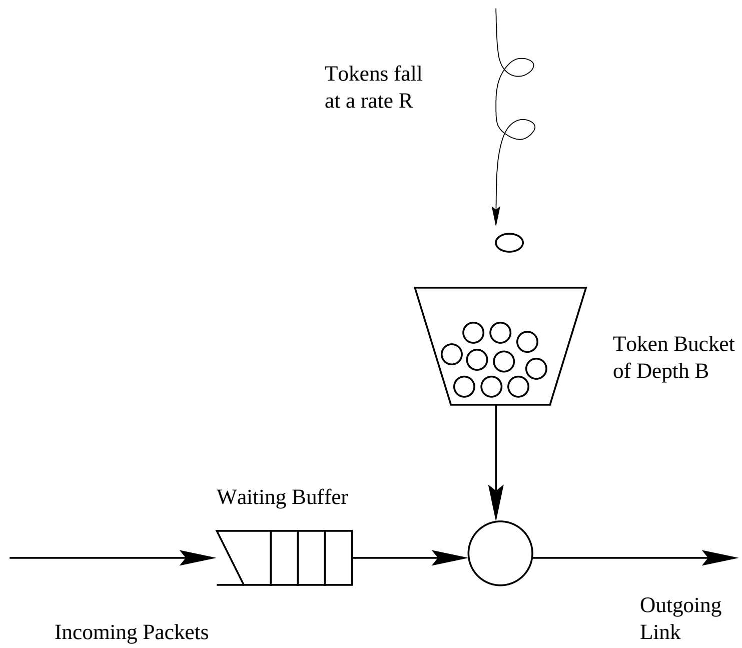

Figure 4.12: Token Bucket Shaping

effect due to the MAC protocol and due to how the physical channel (i.e. TDMA, FDMA, CDMA) transports cells. Source traffic descriptors [20] are instrumental in connection establishment. We want to measure the effect the MAC protocol has on the source traffic description and if simply restoring the packet stream to its original maximum flow rate is sufficient to re-shape the traffic.

The underlying TDMA channel has time slots of 20 milliseconds in length each containing 4 ***PHY*** cells. The number of slots per frame and the total bandwidth is a function of the network load. Load is defined as:

$$\rho = \frac{N_{ET} \cdot S_{AVG}}{B_T} \tag{4.10}$$

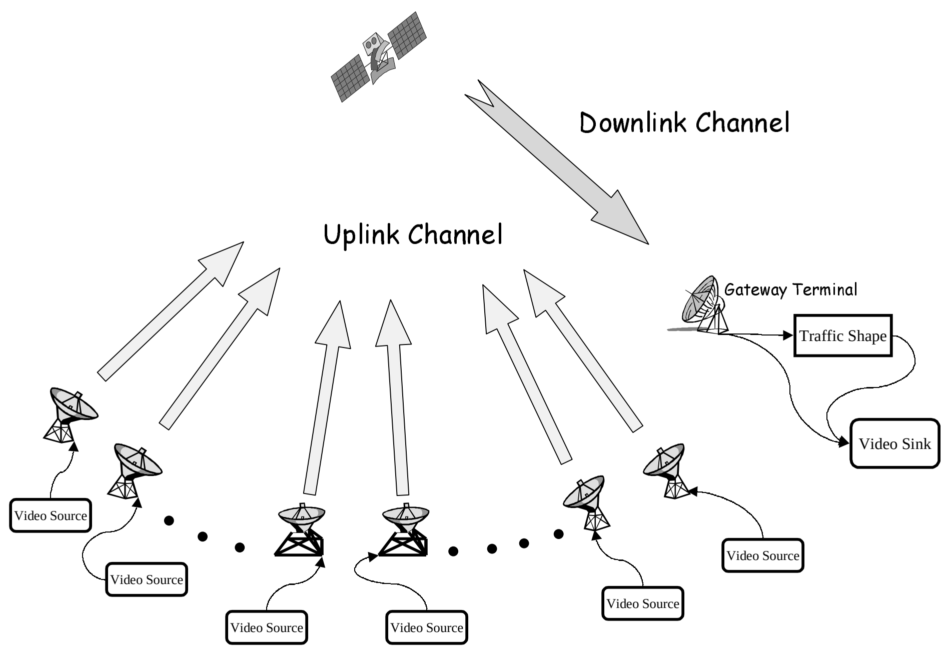

Figure 4.13: Simulation Environment

where:

N_{ET} = the number of earth terminals

S_{AVG} = the average source rate of the video trace

B_T = the total available uplink bandwidth

For network stability, $\rho < 1$. Our simulations were conducted for loading values from 0.1 to 0.9 for completeness, however the interesting loading values are those that exceed 0.5. For our simulations we kept the number of earth terminals at 25 and varied the loading be varying B_T. Since we are primarily concerned with geosynchronous (GEO) satellite systems, our one-hop delay (i. e. earth terminal to satellite) is 135 milliseconds. The statistics measured for all protocols are delay and delay variation. For RA/DAMA Network Layer packet loss was also recorded.

Using our simulation tool, we first implemented FBA, DAMA, and hybrid FBA/DAMA protocols to evaluate their performance for video traffic. In the FBA case, the total bandwidth, B_T, is simply divided equally among all of the earth terminals. For the DAMA protocol, every TDMA frame each earth terminal transmits its output queue size to the allocating agent at the satellite using the control mini-slots available at the start of each frame. The allocation of bandwidth is done exactly according to the transmitted request with a round-robin mechanism in place to ensure fairness in the average sense. The FBA/DAMA technique, described briefly in Section 4.1, was proposed by Hung et. al. and is implemented in exactly this manner with the only difference being the inclusion of a round robin service policy, also to insure fairness. Figures 4.14 and 4.15 show the end-to-end delay and delay standard deviation for FBA and DAMA. Intuitively, under light loading a fixed bandwidth allocation provides the lowest delay. Once the loading exceeds 0.7, the delay increases rapidly and by 0.73, it has surpassed DAMA. DAMA end-to-end delays are always quite high and relatively unaffected by the loading. Since each arriving packet is accounted for in each new request, the bandwidth allocated, for low to moderate loads, exactly matches the bandwidth required by an earth terminal. This leads to high utilization of the uplink channel and thus the delay and delay variation are not affected by the operating load point. Only at very high loads (≈0.9) do the delay statistics show any change.

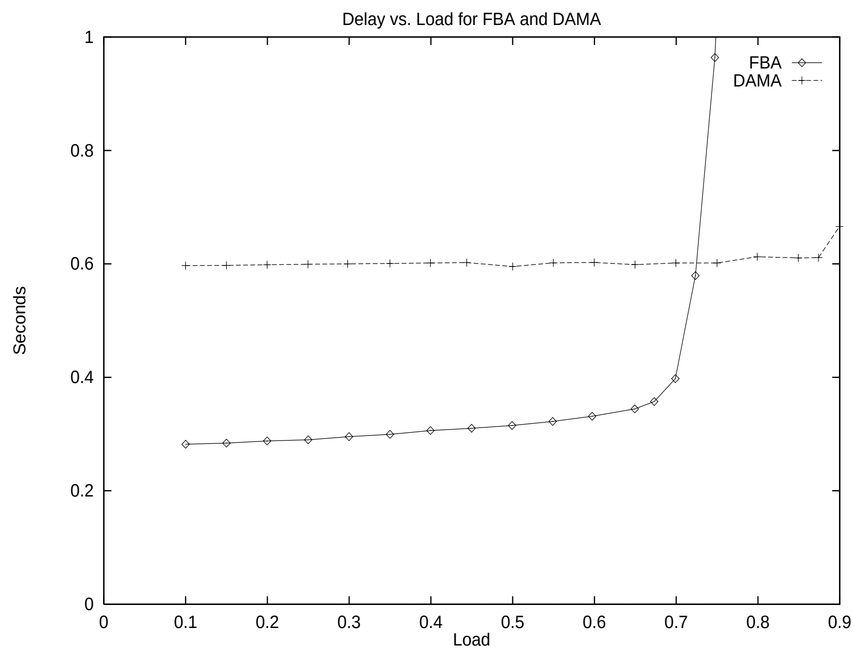

Figure 4.14: Packet Delay for FBA and DAMA

The same statistics are shown in Figures 4.16 and 4.17 for hybrid FBA/DAMA. As stated in Section 4.1, each earth terminal has a fixed bandwidth, ρ^{fx}, allocated to it at all times.

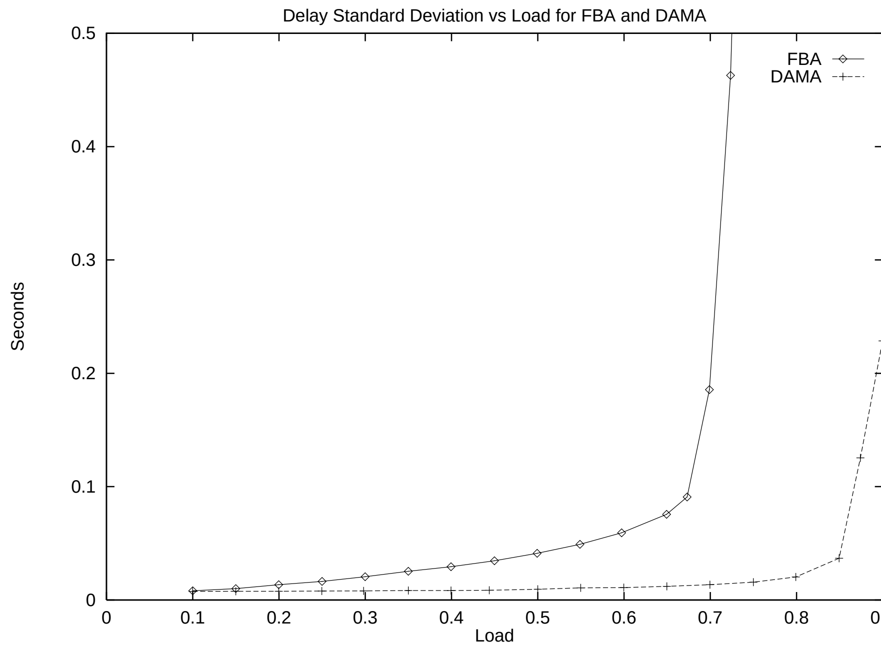

Figure 4.15: Packet Delay Standard Deviation for FBA and DAMA

We wanted to quantify the delay performance as a function of the ratio of fixed bandwidth (ρ^{fx}) to the average source rate (S_{AVG}) mentioned in equation (4.10). We will call this ratio, ρ^{fx}/S_{AVG}, G. For example, when G=0.5, each source has half of its average rate, in this case 146 kb/s, allocated to it at all times. The lower G, the closer the system is to pure DAMA and the higher, the closer it is to FBA. Some load points are not possible for certain values of G. Since we chose to keep N_{ET} constant and vary B_T, there comes a point at which there is not enough bandwidth to satisfy the value required for a given G. This occurs for the higher values of G. The delay variation, unlike the delay, is not monotonically increasing with decreasing G. Since delay variation is lowest for DAMA, FBA/DAMA shows a decrease in delay variation as G approaches DAMA, but not before an apparent jump occurring at G=1.25.

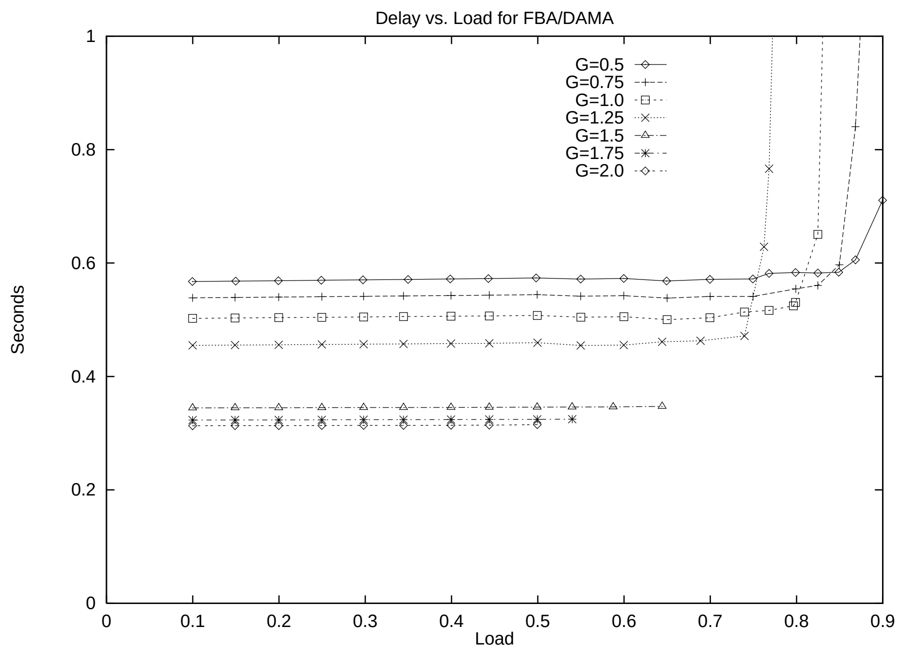

Figure 4.16: Packet Delay for FBA/DAMA

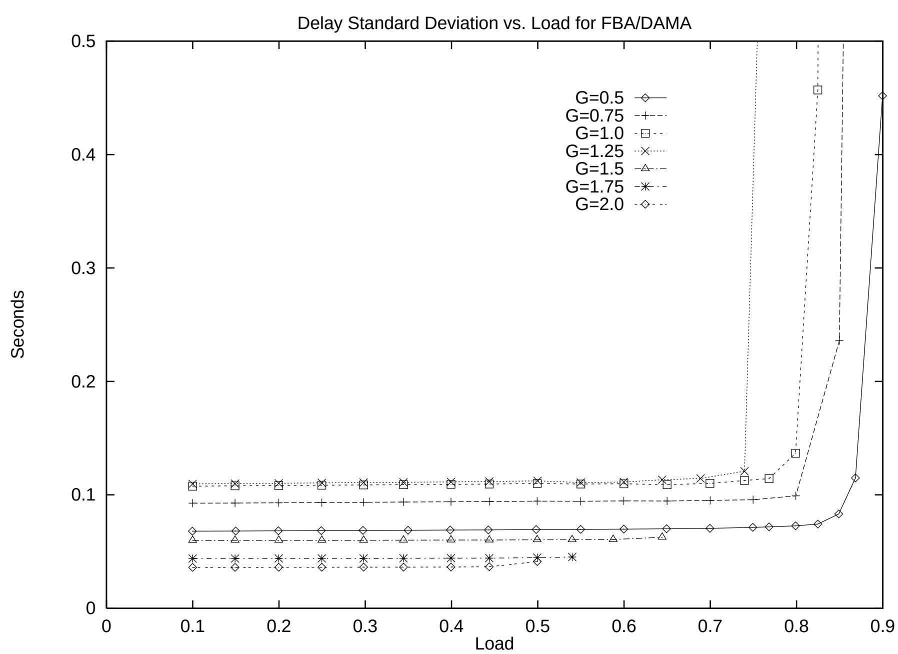

Figure 4.17: Packet Delay Standard Deviation for FBA/DAMA

These results indicate that if the network is lightly loaded (i. e. ρ $\leq$0.6), then a simple FBA allocation will provide end-to-end delays of 300 milliseconds or below. This means that the total delay exceeds the transmit time (270 msec) by only 30 msec. However to achieve significant statistical multiplexing gain, and hence an increase in network capacity, we must be interested in the loading region [0.65,0.85]. In this region, the performance of FBA rapidly deteriorates making FBA/DAMA the more desirable MAC technique. If ρ^{fx} can be varied as the load varies, then delays of less that 600 milliseconds can be achieved even to network loads of 0.9. The penalty paid for abandoning FBA is in delay variation. Only by adopting a pure DAMA approach will delay standard deviations be held below 50 msec.

These results give a background with which to compare our RA/DAMA technique. Since there are many simulation parameters that are potentially arbitrary, we have chosen some a priori and others we have made parameters of simulation. As stated in section 4.3, for generating a DAMA metric that tracks the slow time behavior of the video trace, we have set $T_c = 300$ milliseconds and $w = 4$. This will cause the arriving packets to accumulated over a 300 msec interval, which is the length of the GOP, and to be averaged over the past 4 values. The delay threshold, D_T, was set to 500 milliseconds. This value will cause packets to be moved from the DAMA queue to the RA queue if the estimated delay exceeds 500 msec. This is simply a threshold value for our algorithm and it will not guarantee either maximum or average delays. We have chosen as our parameters of simulation the maximum transmit unit (MTU) size and the maximum flow rate (MFR). The MTU size will take on three values; 1024 bytes, 512 bytes, and 53 bytes. To measure the effects of traffic shaping, we have selected four values for the MFR; ∞ (i. e. no shaping), 750kb/s, 500kb/s, and 400kb/s.

Figures 4.18, 4.19, and 4.20 show the average delay, delay variation, and packet loss rate for the 512 byte MTU case. There appears to be little difference in the delay statistics when the maximum flow rate is greater than or equal to 500 kb/s. Average delays of 390 to 420 milliseconds are achievable with network loads of up to 0.82. Once the MFR is constrained to 400 kb/s, the average delay increases by approximately 200 milliseconds. The increase in delay is due entirely to the token bucket smoothing. The packet loss rates monotonically decrease with decreasing MFR. Packet losses appear to be very low at the final load point of 0.9. This sudden drop off in packet loss at high loads is due to a limitation in the estimation algorithm (4.7). At heavy loading, the assumption that $E\{B(\lfloor\frac{i}{m}\rfloor)\} \approx E\{M(\lfloor\frac{i-z}{m}\rfloor - 1)\}$ is no longer valid. This produces an estimated delay that is too low. This, combined with a average reduction in bandwidth on the RA channel, causes the CSAlg algorithm to move very few packets from the DQ to the RAQ. This leads to a dramatic increase in average delay and decrease in packet loss rates. These results would indicate that the optimal delay / loss operating point would be MFR=500 kb/s. This point has nearly the same delay as the two higher maximum flow rates while providing a loss rate of 1.2% or less at loadings of up to 0.85. If a loss rate of below 1% is desired, then a delay penalty must be paid.

When the MTU size is reduced to 53 bytes, the delay values show similar behavior. Figures 4.21 and 4.22 show that the average delays are slightly higher for the larger maximum flow rates. This can be attributed to the QA algorithm moving a lower percentage of the packets from the DAMA queue to the RA queue, leading to slightly higher delays. Figure 4.23 show that the loss rates for ATM sized cells are less than when the MTU is 512. This is primarily due to the reduction in collision region resulting from the reduction in MTU size. It is also due in a secondary manner to the previously mentioned reduction in use of the RA channel. In comparing the MTU sizes it is clear that the average delays and delay deviation is relatively unaffected by MTU. However the packet loss rates for ATM sized packets are from 2 to 4 times less.

Comparing RA/DAMA to the other MAC schemes reveals that over all load points, RA/DAMA outperforms DAMA by up to 180 milliseconds. In comparison to FBA, when the network loading is light (i.e. $\rho < 0.7$), FBA is superior to RA/DAMA. Once the loading exceeds 0.7 however, FBA becomes unusable and is vastly outperformed by RA/DAMA. Over the load region [0.75 , 0.9], RA/DAMA has a average delay that ranges from 50 to 120 milliseconds less than hybrid FBA/DAMA. The delay variation of FBA and FBA/DAMA always is less than RA/DAMA for light loading, and for the loads of interest, the delay variation is roughly the same, depending on the choice of G. All of these results indicate that regardless of MTU size, if a packet loss rate of 1.5% or less is tolerable, then using RA/DAMA over other MAC techniques will deliver a delay of 50 to 180 milliseconds less during heavy network loading (≈ 0.9). Over this same load region, our results indicate that the delay variation is not greatly affected by the choice of MAC technique, excluding FBA.

Figure 4.24 shows histograms of inter arrival times for the four techniques, each recorded at

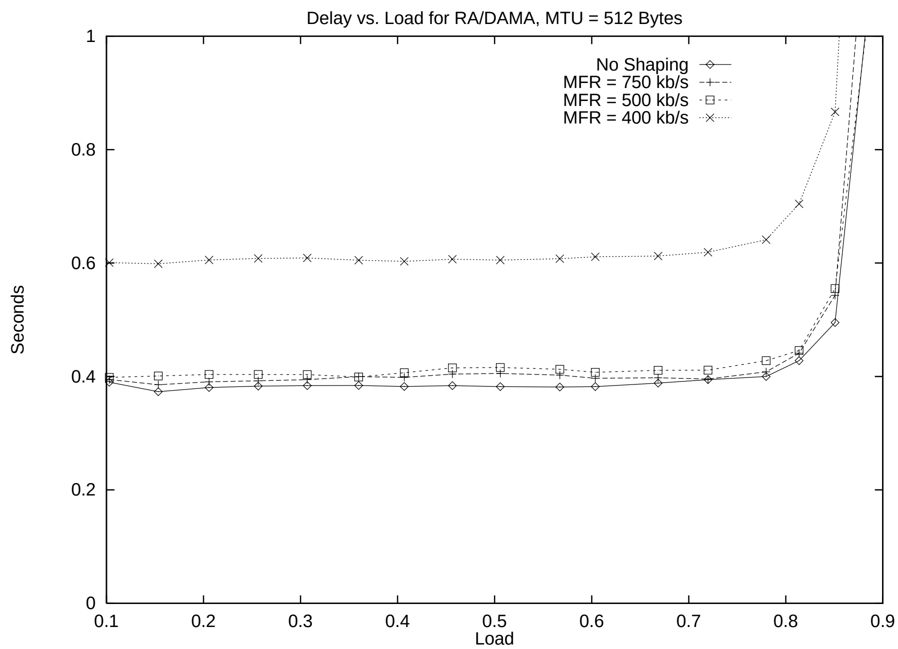

Figure 4.18: Packet Delay for 512 Byte Packets

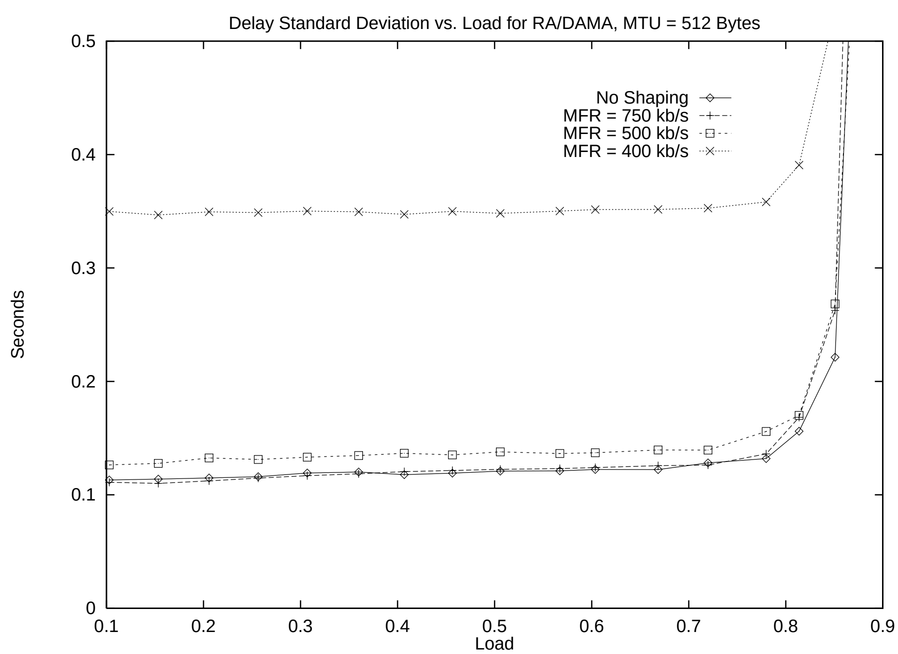

Figure 4.19: Packet Delay Standard Deviation for 512 Byte Packets

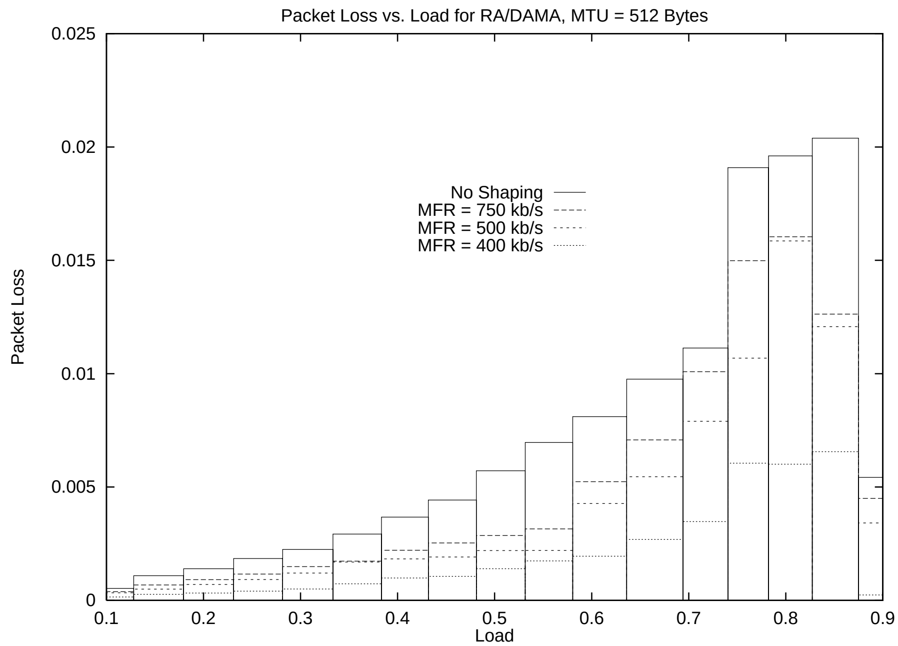

Figure 4.20: Packet Loss Rate for 512 Byte Packets

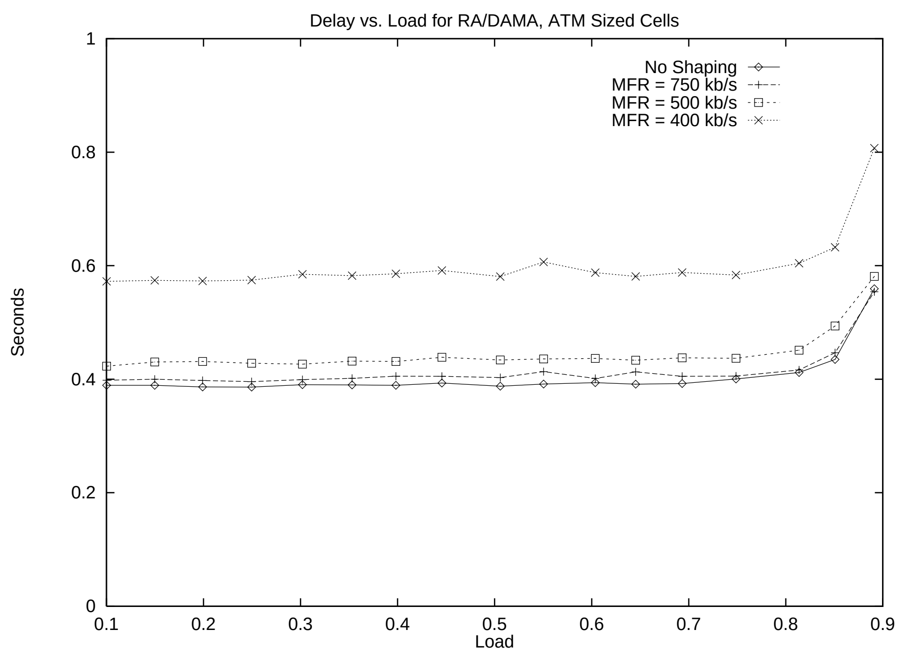

Figure 4.21: Packet Delay for ATM Sized Packets

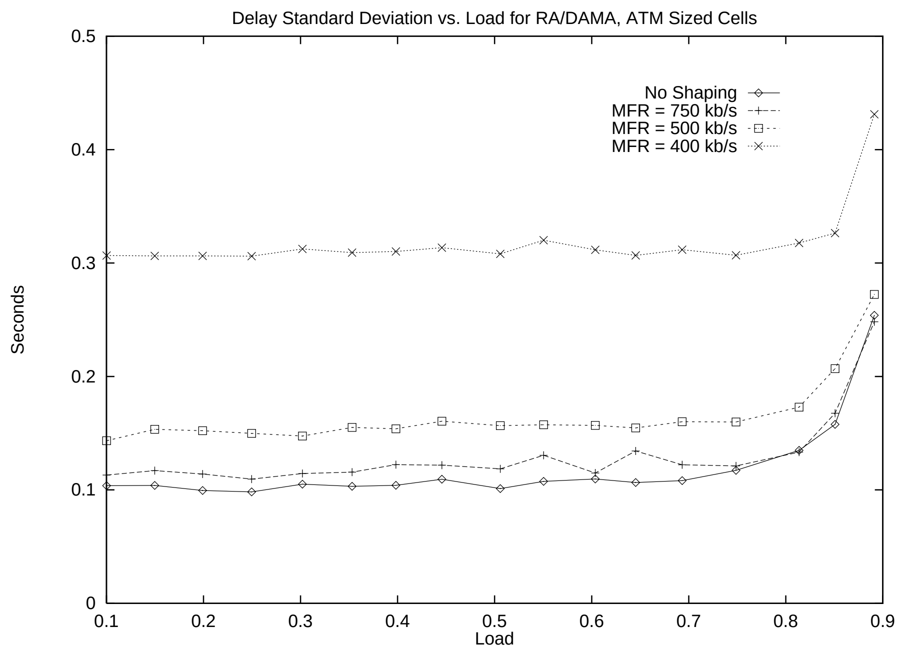

Figure 4.22: Packet Delay Standard Deviation for ATM Sized Packets

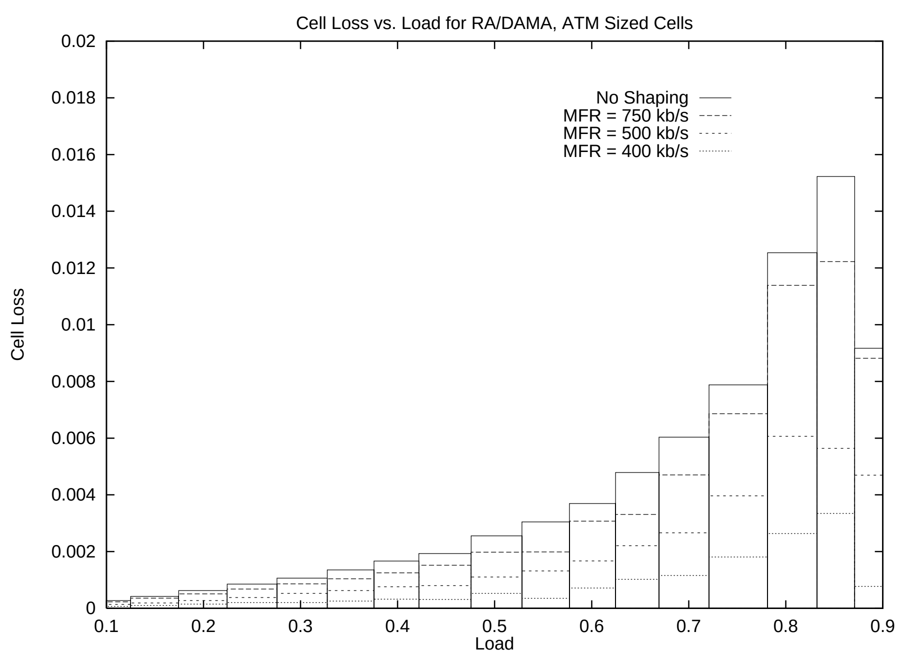

Figure 4.23: Packet Loss Rates for ATM Sized Packets

a network loading of 0.75. DAMA has a symmetric histogram centered at 565 milliseconds, approximately two round-trip times. FBA has a unimodal distribution, with the greatest weight at 270 msec. This corresponds to packets arriving at an empty queue and being transmitted immediately. FBA/DAMA is bimodal, with peaks occurring at one and two round-trip times. For RA/DAMA, there is a very large drop off at 500 msec due to the channel selection algorithm. Since the delay threshold D_T was set to 500 msec, this causes the rather abrupt drop off.

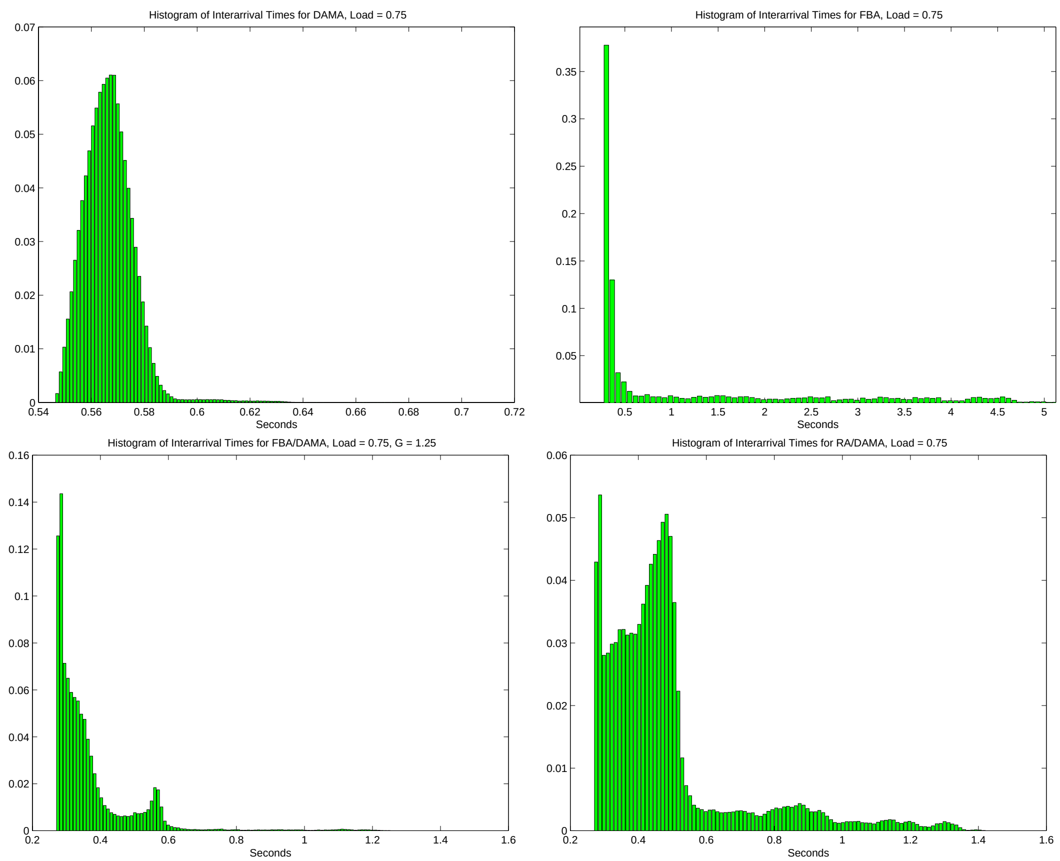

Figure 4.24: Histograms of Packet Inter Arrival Process

The effect of the MAC sub-layer and the Physical Layer on the traffic description of the video stream is shown in Figures 4.25 and 4.26. The values shown are token bucket parameters computed for the original stream and received streams [40]. This figure shows the clumping effect of the PHY and MAC layers. Figure 4.26 shows the token bucket parameters after the packet stream has been restored to its original maximum flow rate. It would appear that the closer the MAC protocol is to FBA the greater the effect on the traffic descriptors. This can be explained by referring back to Figures 4.14, 4.16 and 4.24. At a loading of 0.75, both FBA and FBA/DAMA, G=1.25, have high delay variation. The histogram of FBA has significant weight up to 4.5 seconds and the histogram of FBA/DAMA is strongly bimodal. This large variation in inter arrival times will have a strong effect on the traffic descriptors. DAMA seems to cause little change to the token parameters. If restoring the maximum flow rate is the only traffic shaping possible under strict end-to-end delay constraints, then DAMA and DAMA hybrids will lead to better performance.

4.6 Conclusion

In this chapter we presented a new medium access control protocol suitable for real-time sources communicating over high latency satellite channels. This MAC protocol employs a combined random access / demand assigned multiple access so as to achieve high channel utilization, low delay, low control overhead, and tolerable packet loss. We have conducted computer simulations to verify our technique and for comparison purposes. We have concluded that if packet loss less than 3% is tolerable, then RA/DAMA will achieve

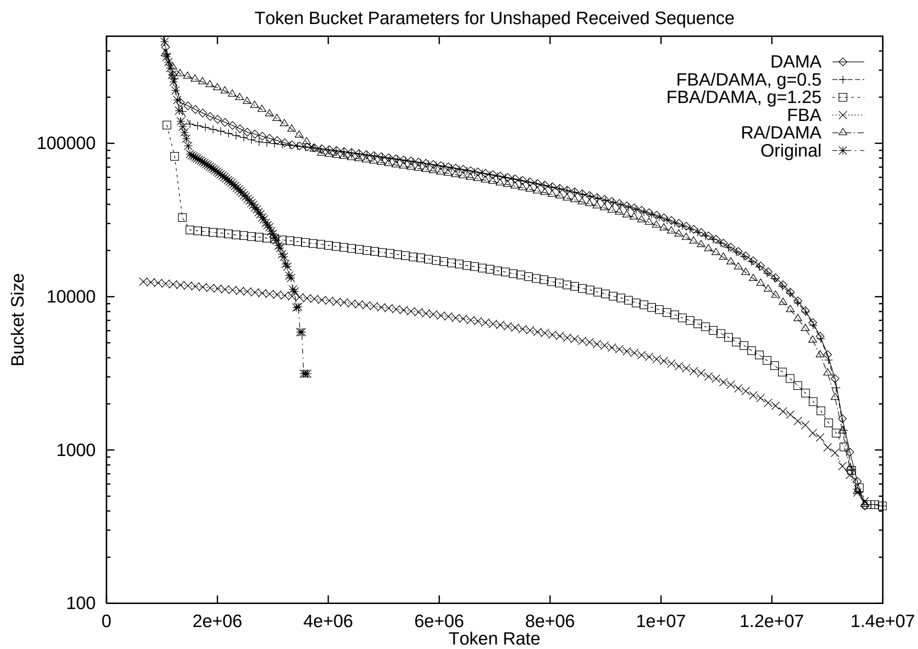

Figure 4.25: Token Bucket Parameters for Unshaped Received Sequence

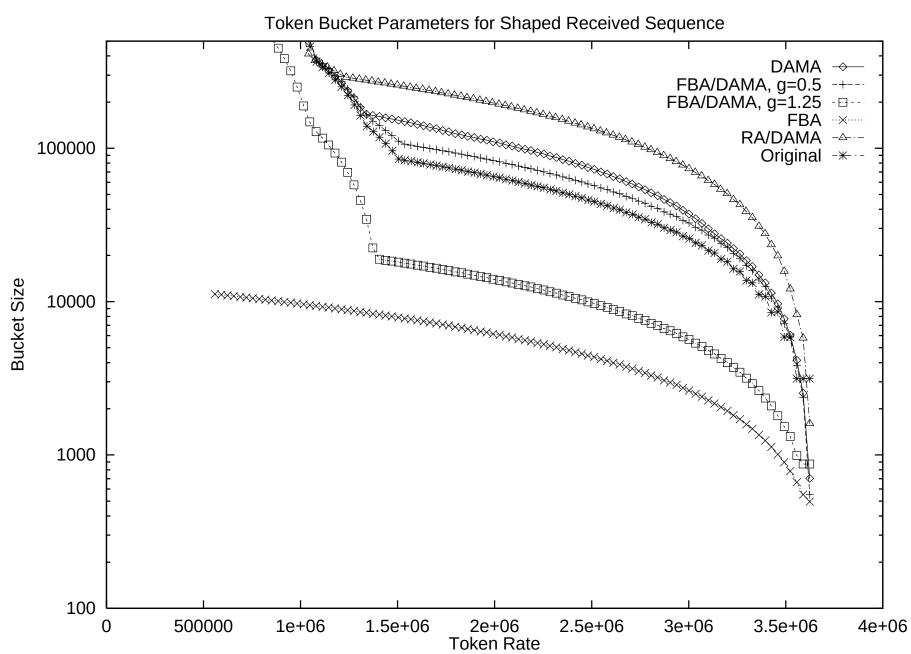

Figure 4.26: Token Bucket Parameters for Shaped Received Sequence

considerably lower delays than other known satellite MAC protocols. There are some limitations of this protocol. First we have only considered a certain traffic type, namely source coded video. In order to provide a complete MAC solution that provides a suitable QoS for other Internet applications, additional MAC techniques must be developed that can work in conjunction with RA/DAMA. In Chapter 5, another TDMA based MAC technique will be presented which is suitable for such applications as interactive computing (i.e. `Telnet, rlogin`), World Wide Web browsing, and file transfers (i.e. `ftp`). This protocol can work in conjunction with RA/DAMA since they are both designed for a dynamic TDMA Physical Layer. Another limitation of RA/DAMA is that it requires some information (i.e. encoding format, bit rate, etc.) about the video source to be known apriori. For non-real time applications, this is not any more of a problem than what is faced in ATM or QoS-enabled Internet networks. However, since RA/DAMA is meant for medium quality video conferencing over satellite uplink channels, it will most likely be delivering real-time audio and video. This means that some form of connection signaling must be passed to the MAC sub-layer prior to the beginning of the streaming session.

Chapter 5

A Medium Access Control Protocol for Computer Data over Satellite Channels

Continuing with the goal stated in Chapter 1 of developing a comprehensive medium access solution for many of the common Internet applications, this chapter introduces a MAC technique for efficiently and effectively transporting computer data over a satellite uplink channel. In recent years, many satellite companies have announced plans to build a network of packet switched capable satellites [51, 22]. These so-called “networks in the sky” are primarily intended to serve as the network of choice for Small Office / Home Office (SOHO), multi-unit dwellings (i.e. large hi-rise apartment buildings), and for areas of the globe where wireline infrastructure is not plentiful. These networks fall under our definition of a “personal earth terminal” wherein it is conceivable that many earth terminals may have only a single user utilizing it at any time. This scenario is illustrated in Figure 5.1. Currently, individuals can purchase a relatively small (75cm) satellite dish capable of duplex communication with a GEO satellite [30]. In future non geo-stationary orbit (non-GSO) systems the dish size will be considerably smaller. The possibility exists for perhaps thousands or millions of single users accessing an uplink channel calls for a medium access scheme that can accommodate as many users as possible, while giving each user a suitable QoS, in terms of throughput. This scenario was the impetus for our creation of *Response Initiated Multiple Access* or RIMA. This uplink MAC protocol follows the assertion made in Chapter 3, Section 3.2.2 in that it is optimized for a certain application / transport layer behavior, while taking into account the limitations imposed by the physical layer (i.e. delay). This chapter will follow the following organization: First, Section 5.1 will provide a more complete explanation of some of the more detailed aspects of the TCP/IP protocol suite. This amount of additional detail is necessary to understand the finer points of RIMA. Then Section 5.2 will present a road-map for designing a MAC protocol for various computer data applications and explain some of the limitations of previously developed MAC techniques. This will be followed by Section 5.3, which lays out Response Initiated Multiple Access in detail, including a state transition diagram. Section 5.4 will describe our simulation methodology for testing RIMA and other protocols in a diverse set of computer data transfer scenarios. The simulation results will be given in Section 5.5, for a variety of scenarios including World Wide Web browsing, bulk file uploads, and interactive computing. We will conclude in Section 5.6.

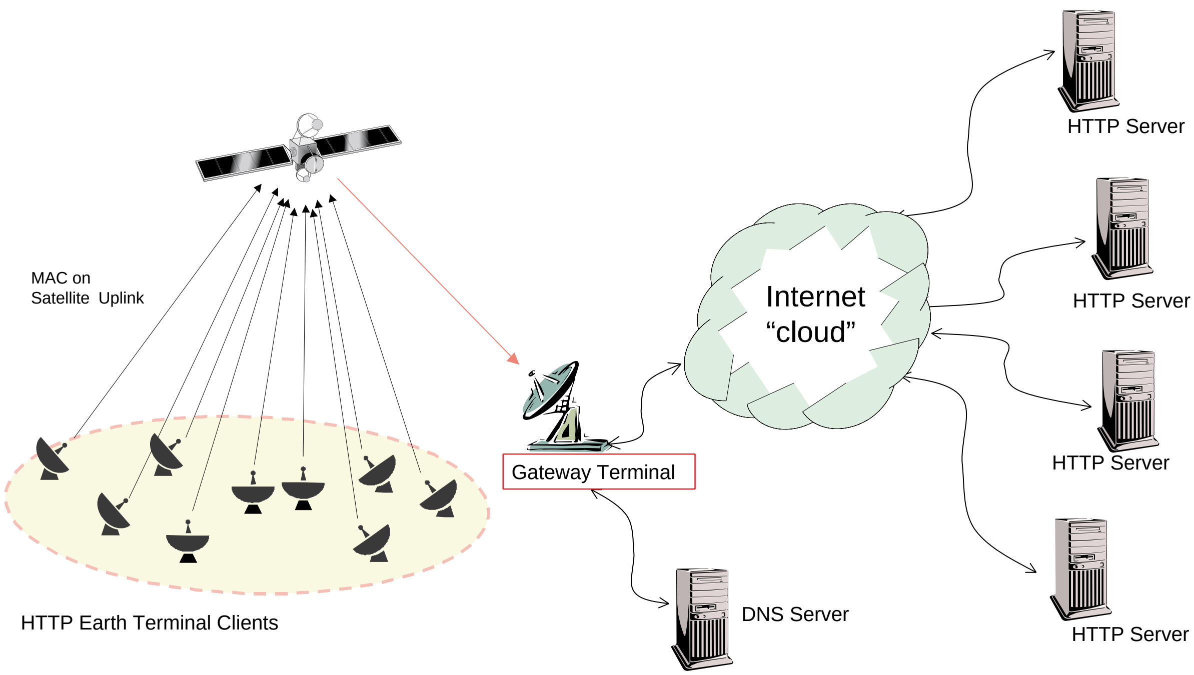

Figure 5.1: A Satellite Network connecting Clients and Servers through the Internet

5.1 Some Additional Background on the TCP/IP Protocol Suite

In this section, some additional background information on the TCP/IP protocol suite will be provided that goes beyond the overview presented in Chapter 2. This is required in order to understand the packet level dynamics of computer data traffic that has been generated by TCP/IP applications and subsequently shaped by the mechanics of TCP. First, a more detailed description of TCP will be provided, followed by a description of HTTP, the File Transfer Protocol (`ftp`), and the interactive console application `Telnet`.

5.1.1 Transmission Control Protocol (TCP) Mechanics

In Chapter 2, Section 2.5.1 we provided a brief overview of the Transmission Control Protocol. In this section we will add some more detail, as it is needed to understand the design methodology behind RIMA. TCP is a *connection oriented* Transport Layer protocol. A connection oriented protocol (as opposed to a connectionless protocol like UDP) not only establishes who the sender and receiver are, but also establishes a "connection" between the sender and receiver. Since TCP is a duplex connection, the notion of sender and receiver is blurred, so we will call one end the client and the other end the server. The TCP connection is established by a three way "handshake" initiated by the client. The client does this by first sending a 40 byte TCP **SYN** packet. The client considers this an *active open*. Upon reception of this SYN packet the server replies with a 40 byte SYN+ACK packet. The server considers this a *passive open*. The client, upon receiving the server's SYN+ACK packet, replies with an ACK packet of its own and considers the connection now to be open. When the server receives the client's ACK, it considers the connection now to be open. This process is displayed in Figure 5.2. This initial connection set up procedure is important to understand because if the amount of information to be transmitted across the TCP connection is relatively small and, as we shall see shortly, flow control mechanisms inherent to TCP slow the initial rate that a particular TCP connection can inject packets into the network, then reliable delivery of connection establishment packets are crucial to overall performance. If one is considering Web page transfers, in general the size in bytes is small [32] making this an issue. Also having a simulation model which models all of this behavior in an accurate manner is necessary. **ns** is just such a simulation tool.

Once the TCP connection is established, the packets can begin to flow in either direction, subject to TCP's flow control mechanisms. TCP is an end-to-end acknowledgement based transport protocol. For each half duplex connection, the receiver (i.e. the server if the client is sending data, or conversely the client if the server is sending data) "advertises" a window. This window represents the number of unacknowledged packets that the sender can inject into the network. As transmitted packets are acknowledged, the sender advances the window (this is why it is sometimes referred to as a *sliding window*). The reason why the receiver advertises the window is so that the sender does not send packets at a rate that is greater than the rate at which the receiver can process. A key characteristic of this flow con-

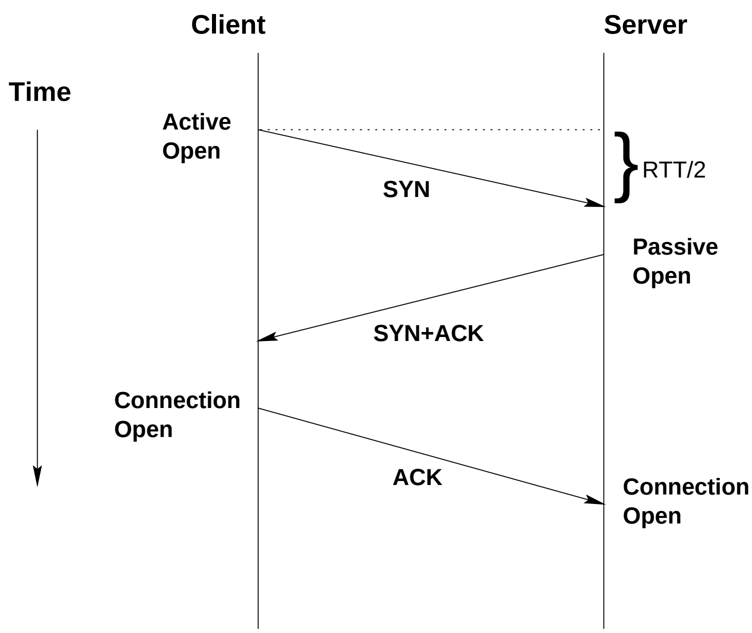

Figure 5.2: Time Line of a TCP Connection Establishment

trol scheme is that it ignores the capacity of the network. The sender and receiver may be able to handle a high rate of packet flow, but the multi-hopped network that interconnects them may not be capable of routing at this rate. This reality lead to a significant modification of the basic TCP algorithm [24]. This modification added *slow start*, *congestion avoidance*, and the concept of *timeout*. Slow start occurs either at the beginning of a connection, or when the connection experiences network congestion. Slow start can be briefly explained in the following manner. Along with receiver's advertised window, another window called a *congestion window* is added to the flow control process. Therefore the sliding window of unacknowledged packets is now the minimum of the congestion window and the receivers window. At connection establishment, the congestion window is set to one. Therefore the sender can send only one packet into the network. Upon receiving the ACK, the sender injects two packets. These two packets will produce two ACK's from the receiver and for each of these two ACK's, two more packets (four in total) will be injected by the sender into the network. This process of two packets for every one ACK allows the congestion window to open at an exponential rate. *Congestion avoidance* is described as follows. Upon detection of network congestion, the current window (i.e. **min(congestion window , receiver advertised window)**) is halved and stored in a variable *ssthresh* and the congestion window is reduced to one. The connection again repeats the slow start procedure (i.e. exponentially increasing the congestion window) until the current window reaches *ssthresh*. Once this occurs the congestion window increases linearly with each acknowledged packet at a rate of $1/ssthresh$. The reasoning behind this is that once *ssthresh* is reached, the connection is approaching the capacity of the lowest bandwidth link on the connection, and therefore should increase its throughput in a more network friendly manner. The means for detecting congestion is when a packet experiences a *timeout*. A timeout occurs when a sent packet goes unacknowledged for a set period of time. This time period is called the retransmission timeout value, or RTO. RTO will be different for each connection, since the round trip time (RTT, the time from transmitting a packet to receiving an acknowledgement) varies with each different client-server connection. Therefore the estimated round trip time, R, is determined using the following recursion equation.

$$R_i = \alpha \cdot R_{i-1} + (1 - \alpha) \cdot RTT_i \tag{5.1}$$

where α is 0.9

RTT_i is measured by subtracting the timestamp of the received ACK packet i from the computer's current clock time. The recursion equation is simply a low pass averaging filter. Equation 5.1 serves to generate a retransmission timeout value that matches the round trip time of the concatenated link. RTO is then calculated to be $2 \cdot R_i$. When a packet is injected into the network, a count-down timer of value RTO is set. When this timer expires, if an ACK has not been received, then connection is said to have experienced network congestion as mentioned earlier. When this happens the connection returns to the slow start mode and the packet is re-transmitted. There are many more details of TCP, since it is a complex and ever evolving protocol. The main characteristics were outlined here since RIMA seeks to capitalize on some of the deterministic behavior of TCP to achieve better performance. For a more in-depth description of the Transmission Control Protocol, please refer to [50].

5.1.2 Web Browsing and the Mechanics of HTTP

Clients retrieve web documents from remote servers using the Hyper Text Transfer Protocol [19]. The mechanics of this retrieval are as follows. The client passes a universal resource locator (URL) message to a domain name server (DNS). In a satellite network this DNS will usually reside at a gateway node. The DNS replies with an Internet Protocol (IP) address of the server which contains the Web page the client desires. This transaction is usually done using the user datagram protocol (UDP) [42]. The client then issues an HTTP GET message to the server, with the name of the desired file. The server responds with the desired file. This transaction occurs using TCP. Often there are many inline objects in the requested file (i.e. attached gif images). Once the client detects the presence of these objects, it sends HTTP GET messages for each object. The mechanics of these "secondary" retrievals are what separate the various HTTP implementations. In the oldest versions, the secondary retrievals occurred in serial only after the initially requested (primary) file was completely received. This meant that there was only one active TCP connection at any given time during the transaction. Netscape version 3.0 and before performs Web transfers using a different technique called parallel connections. In this case up to (usually) four TCP connections were used to retrieve additional inline objects. In this case, if there were more that four objects to retrieve, the HTTP GET messages would be queued until one of the existing TCP connections was closed. This method was replaced by persistent connections (post Netscape 3.0 and others), in which when one of the parallel TCP connections became idle, instead of closing the connection and restarting a new connection, the existing connection was re-used. This technique reduced the time to transfer files since the TCP connection establishment "handshake" would not need to be repeated and more importantly, the TCP slow start mechanism could be avoided. Since individual Web transfers tend to be small, avoiding slow start would aid in reducing the transfer time. Recently HTTP 1.1 has been proposed [19]. HTTP 1.1 uses only a single TCP connection and pipelines HTTP GET messages. Using this method, when the client discovers an inline object, it simply issues the HTTP GET on the existing ongoing TCP connection. Barring network losses (from congestion, corruption, or packet drops), a client using HTTP 1.1 will only experience slow start once. For purposes of our simulation, we have only considered HTTP 1.1 since it is anticipated to be the dominant form of HTTP in the coming years.

5.1.3 Bulk Transfers and Interactive Computing

Bulk transfers are usually needed to transfer files from one computer to another. This is accomplished using the File Transfer Protocol (`ftp`). `ftp` is an application level protocol which establishes two TCP connections, a control connection and a data connection. The control connection is created once and is used for the client to interactively log onto to the server and determine what there is to transfer and where it is located. Once the file has been identified, a data TCP connection is established and the server delivers the entire file to the

TCP module in the operating system's kernel. This file, now considered a TCP message, is segmented into packets and delivered according to the mechanisms of TCP outlined in Section 5.1.1. Once the file is transfered, the data connection is terminated. The control connection is kept alive until the user exits `ftp`. Interactive computing is accomplished using either `Telnet`, `rlogin` or a secure version of either. These protocols are simply interactive application layer protocol which encapsulate keystrokes into messages and then deliver them to TCP. For each protocol, a TCP connection must be established between the client and the server. This is always done at the instantiation of the protocol (i.e. `rlogin hostname.domainname.edu`). Seldom do the flow control mechanisms of TCP effect the performance of either `rlogin` or `Telnet`. The biggest effect on interactive computing performance using either of these protocols is in the latency through the network from client to server.

5.2 Design of Medium Access Control Protocols for Computer Data

We will now lay out a road-map for designing an uplink MAC protocol for TCP/IP computer data by indicating the desired features. This will make clear our approach in designing RIMA. Following the methodology presented in Chapter 3, Section 3.2.2, we have targeted a dominant Internet application, that being HTTP World Wide Web browsing, and have set this as a reference point. Identifying a particular application allows us to compare and contrast existing schemes as to their ability to perform given this application and it allows us to determine what characteristics an effective and efficient medium access scheme must posses. Section 3.2 gave an overview of medium access control. In that overview, random access (RA), reservation random access (RRA), and demand assigned multiple access (DAMA) were presented. In order to "lay the ground work" for the ideas behind RIMA, some additional intuition about medium access must be stated. One of the main drawbacks to DAMA techniques is the delay incurred by having to perform a request "handshake" with the *Allocating Agent* (AA). For most satellite channels, this delay is non-negligible. Since we are simulating HTTP Web transfers, the source traffic emerging from an earth terminal will be quite bursty, so clearly using a fixed bandwidth allocation (such as TDMA) will yield very poor network capacity. In considering DAMA, one needs to determine the distribution of packet sizes that will be transmitted in the uplink channel. Demand assigned multiple access works well for bursty sources only if the sizes of the data packets far exceed the size of the DAMA request packets. If this is not the case, then DAMA performs no better than random access. Studies have shown a strongly bi-modal distribution of TCP packet sizes [9] due mainly to the fact that most TCP packets are either data carrying packets, which are close to the maximum transmit unit (MTU) size, or acknowledgement packets, which tend to be less that 100 bytes. Since the uplink channel will be used by the HTTP client (this is our dominant application that we are designing the MAC to support), and since the bulk of the data will be flowing in the downlink channel (i.e. from server to client), the majority of packets that will be transmitted in the uplink channel will be acknowledgement packets. For this reason and the fact that satellite channels have high latencies we have eliminated traditional DAMA techniques from consideration and focused on random access techniques when designing RIMA. When any form of random access is used, one must make the *collision space* as large as possible to reduce the collision probability. For instance, the collision space in a TDMA system is the number of time slots available for random access transmission. The collision space is widened when either more slots are available for RA transmissions, or when the number of users contending for these slots is reduced. The collision space can also be logically widened by synchronizing contending users, as was shown by the improvement of Slotted ALOHA [26] over the original ALOHA protocol.

The two most common satellite random access or reservation random access protocols are Slotted ALOHA and Reservation ALOHA [13]. Slotted ALOHA is a synchronous time version of the original ALOHA protocol proposed first by Abramson in 1970. The original ALOHA protocol had users transmitting whenever they received a packet in their output queue. The receiver, in this case the satellite, would acknowledge the packet. If the packet was not acknowledged, then the transmitter would assume the packet lost and backoff for an exponentially distributed period of time. Slotted ALOHA simply uses the exact same mechanism as ALOHA, except when a packet arrives to a user's output queue, it is transmitted at the next synchronous instant in time (i.e. the beginning of the next TDMA slot). The backoff mechanism for Slotted ALOHA can take on many forms. The development of a suitable backoff scheme will be presented in Section 5.5. The main drawback to both

ALOHA techniques is that they end up with very poor throughput (i.e. the percentage of the channel actually used for transmitting data). With packets that have interarrival times that are exponentially distributed, ALOHA and Slotted ALOHA achieve throughputs of 0.18 and 0.36 respectively. The benefit of both ALOHA techniques is the simplicity of implementation. Reservation ALOHA was first proposed in [13]. It sought to achieve the simplicity of use that ALOHA offers while attempting to implicitly reserve bandwidth to improve on throughput. When a packet is successfully received at the satellite, bandwidth is implicitly reserved in that slot in the next frame (in this case a TDMA frame is the round-trip time to the satellite). This was done to accommodate bursts of data that took the form of many packets. A shortcoming in Reservation ALOHA occurs when the arrival statistics revert to single packets with exponential interarrival times. Under this scenario, Reservation ALOHA achieves less than half the throughput of Slotted ALOHA.

In designing an uplink MAC technique, three things should be taken into consideration : the terrestrial user's needs, the terrestrial user's traffic profile, and the satellite system's needs from the perspective of the network operator.

Terrestrial User's Needs

1. Instant access to the satellite uplink channel without any bandwidth allocation set up time
2. High throughput
3. No collisions on the uplink channel

The first item would indicate that a random access scheme is needed since RA needs no channel set up. The third item seems to preclude random access since collisions are unavoidable with random access. Proceeding now to the terrestrial user's traffic profile:

Terrestrial User's Traffic Profile

- For **HTTP WWW** users the following behaviors / conditions exist:

  1. The user will "think" and be inactive for some period of time. This corresponds to the user reading the current Web page. During this think time, no uplink bandwidth is needed.

  2. When the user emerges from thinking, it will need to transmit a relatively small packet (either a DNS lookup packet, which is on the order of 50 to 100 bytes , or a TCP *SYN* packet which is exactly 40 bytes).

  3. The user will then commence a heavily asymmetric transaction with a remote HTTP server. The asymmetry will be in the reverse link (i.e. server to client), however there will be points in the transaction when the client will have a sizeable amount of data to send on the forward link. This will occur for the initial HTTP *GET* message and any subsequent GET messages needed to request inline objects. Figure 5.3 shows an example of the bandwidth utilization of a HTTP transfer on both the forward and reverse links.

- For other protocols, the following conditions may exist:

1. A terminal may produce and transmit packets in a sporadic bursty manner (i.e. `Telnet`).

2. A terminal may require a bulk data transfer in the forward link (i.e. `ftp` or `smtp`).

3. A terminal may produce and transmit packets in a constant bit rate or *sustained stream* manner.

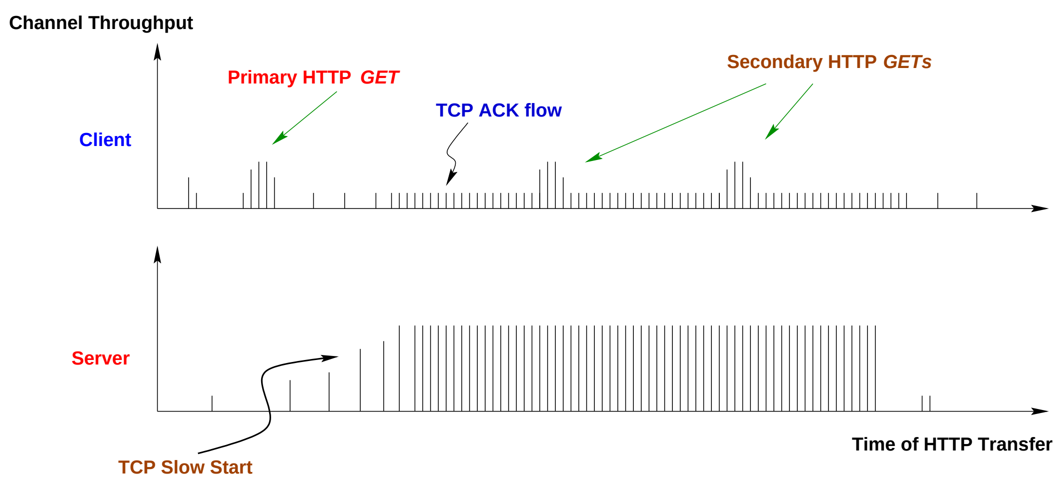

Figure 5.3: Link Utilization For a Client-Server HTTP Transfer

Satellite System's Needs

1. To support as many terrestrial users as possible. This maximizes the network operator's revenues.

2. To provide a suitable QoS for those users accessing the network

5.3 Response Initiated Multiple Access

The key idea behind the development of RIMA was that TCP (upon which HTTP runs), has predictable behavior and that TCP packet headers contain a port identification field where what are called “well known port numbers” reside. These well known port numbers indicate the type of application that the particular TCP connection is currently carrying [50]. For HTTP, this well known port number is 80. Let us now define a *TCPAgent* as one end of a duplex TCP connection. When a large packet arrives at a TCP Agent, it will (most likely) transmit an acknowledgement (ACK), since the large packet it just received (most likely) contained data needing to be acknowledged. If a TCP Agent receives a small packet (i.e. an ACK) , then it may transmit a large packet, if it has more data needing to be sent. This behavior is due to the window based flow control mechanism on which TCP operates. For a star topology, with high latency, the ramifications of this behavior are as follows. When the allocating agent (AA) at the satellite receives a packet from the gateway node, a packet whose destination is an earth terminal (ET) within its spot beam, it can know that once the ET receives the packet, if it was a large packet, then with high probability the ET will need uplink bandwidth in which to transmit the ACK. If the packet was small, with reasonable probability it will need uplink bandwidth in which to transmit more data. Regardless of how bursty and unpredictable the arrival of traffic from the gateway to the satellite is, this behavior will still exist and be predictable. Thus if collision free (i.e. reserved) uplink bandwidth is allocated in the uplink channel for a particular user upon receiving a packet destined for that user from the gateway, this will reduce the need for random access transmissions, since the uplink bandwidth needs of users can be accurately predicted. This has the effect of widening the *collision space*. Since random access collisions are very costly in high latency systems, if random access is used sparingly, then a potentially huge throughput increase can be realized. This gives rise to the name *Response Initiated Multiple Access*, in that a response from the server initiates collision free multiple access in the uplink. The only time random access must be employed is when a HTTP user emerges from the “thinking” period, since this cannot be predicted. When a user emerges from the thinking period, it will either issue a DNS lookup packet or a TCP SYN packet, both of which are small. Small packets have lower collision probability since they will occupy a smaller region of the collision space. Since RIMA will guarantee collision free ACKs to flow in the uplink channel, the TCP window at the server will open very quickly, resulting in excellent throughput. With this methodology understood, a framework for implementing RIMA can be established.

Figure 5.4 shows how RIMA is implemented in a satellite based AA. As with the ALOHA protocols, RIMA is TDM based and assumes time slots making up a TDMA frame. At the beginning of each downlink TDM frame, the RIMA AA issues a *frame descriptor packet* (FDP). This packet indicates which collision free slots are allocated to which ETs. The remaining slots are available for RA transmissions. After the FDP is sent, the remaining bandwidth in the downlink is used for data transmissions. The FDP is generated by a combination of two inputs : the RIMA Allocation Algorithm operating on server packets arriving via the gateway (see the following pseudo code), and additional bandwidth requests received from the ETs within the satellite's spot beam.

The following pseudo code illustrates the RIMA Allocation Algorithm (RAA).

```
if( PORT_NUM == 21 ) { /* ftp is port number 21 */

    if( PACKET_SIZE <= ACK_SIZE )

        BYTES_ALLOC = MTU_SIZE

    else BYTES_ALLOC = ACK_SIZE

} else if( PORT_NUM == 23 ) { /* telnet is 23 */

    if( PACKET_SIZE <= ACK_SIZE )

        BYTES_ALLOC = 0

    else BYTES_ALLOC = ACK_SIZE

} else if( PORT_NUM == 25 ) { /* smtp is 25 */

    if( PACKET_SIZE <= ACK_SIZE )

        BYTES_ALLOC = MTU_SIZE

    else BYTES_ALLOC = ACK_SIZE

} else if( PORT_NUM == 80 ) { /* http is 80 */

    if(PACKET_SIZE <= ACK_SIZE)

        BYTES_ALLOC = 500

    else BYTES_ALLOC = ACK_SIZE

} else BYTES_ALLOC = 0
```

Where:

PORT_NUM is is the TCP well known port number
PACKET_SIZE is the size in bytes of the received server packet
ACK_SIZE is the size of a TCP ACK packet (in IPv4 this is 40 bytes)
BYTES_ALLOC is the amount of bandwidth in bytes that is apportioned by the RAA
MTU_SIZE is the size of the maximum transmit unit

What this algorithm accomplishes is that if a packet arrives at the satellite from the server, the TCP and or UDP port number can be read and the proper amount of uplink bandwidth can be pre-allocated. We are most interested in HTTP, therefore when a small HTTP packet arrives (i.e. a TCP ACK), 500 bytes are allocated in the uplink channel. Apportioning 500 bytes if the packet is small is a subjective value chosen based upon the measurement of HTTP GET sizes presented in [32]. Likewise, when a large HTTP packet is received, just enough bandwidth is allocated to accommodate an ACK packet (40 bytes). Since we are running RIMA using a TMDA multiple access physical layer, these byte quantities will be rounded up to the appropriate amount of TDMA slots. Other TCP/UDP port numbers are included in order to make RIMA robust across the entire suite of dominant TCP/IP applications. Since there is always the possibility that an individual ET needs more bandwidth than RIMA Allocation Algorithm gives, then additional bandwidth requests can be "piggy backed" on arriving data packets. These additional bandwidth requests will be accommodated, subject to B_{max}^{frame}, the maximum amount of bandwidth an individual user may have on a per frame basis. Having a upper bound on bandwidth will preclude the real possibility of one user "grabbing" a disproportionate amount of uplink bandwidth when performing a bulk transfer. Another notable benefit of RIMA is that very little *state* is maintained at the satellite. Many previously proposed MAC techniques had the AA at the satellite maintaining state information about every active user in the spot beam. In a GEO satellite environment where there are potentially tens of thousands of users, a scheme that requires the AA to maintain user state information is not scalable.

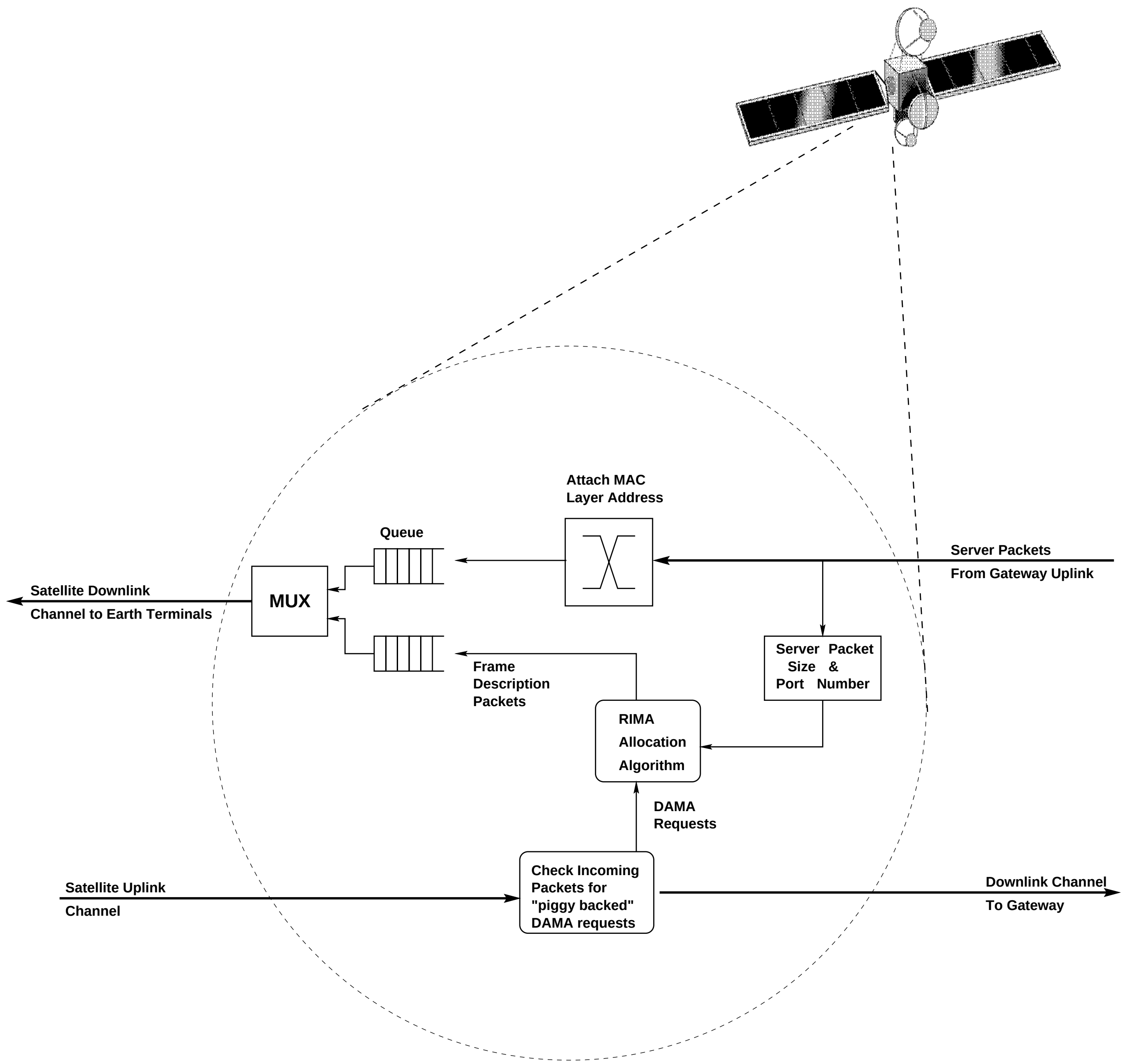

Figure 5.4: A Functional Block Diagram of how RIMA is implemented in a Satellite

Since RIMA has piggy backed DAMA functionality, it can accommodate bulk transfers (`ftp`, `smtp`) with the same efficiency as DAMA. If a constant bit rate (CBR) source is attached to RIMA, it would also work since DAMA techniques work on CBR sources as well. Sporadic bursty users (such as `Telnet`, `rlogin`) can also be accommodated through random access (since their byte allocation in zero). All of this means that although RIMA is optimized for a particular, currently dominant Internet traffic pattern (i.e. HTTP WWW transfers), it will also work as well as other MAC techniques for the rest of the applications common in the Internet. Figure 5.5 shows the state machine that makes up the RIMA protocol.

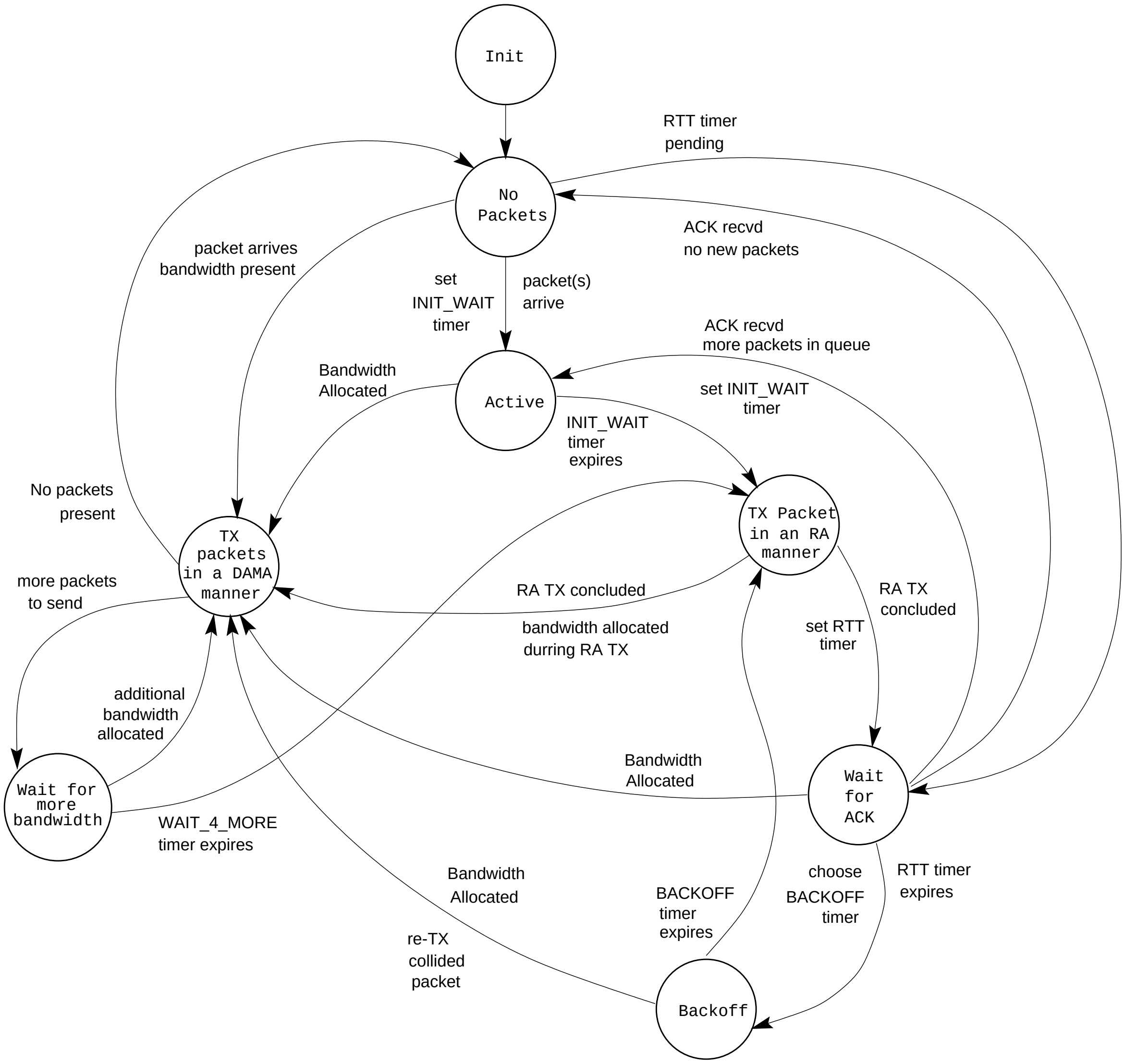

Figure 5.5: A State Transition Diagram for Response Initiated Multiple Access

5.4 Simulation Methodology

We now describe our satellite simulation test-bed. Figure 5.1 shows how the earth terminals are connected with remote Web servers. The HTTP clients are connected to the individual earth terminals, one client per terminal. This emulates the "personal earth terminal" model that has been assumed throughout this dissertation. All earth terminals share a common uplink channel. The MAC protocol in place on the uplink channel is either S-ALHOA, R-ALOHA, or RIMA. The DNS server is located at the terrestrial gateway node, one satellite hop away from each earth terminal. This architecture reinforces the notion that the satellite is operating as an Internet "node" and not in a bent pipe manner. Consistent with the simulation done in Chapter 4, the only channel impairment that we are concerned with is multiple access interference (i.e. collisions), and not losses due to various other impairments such as link outages, burst errors, and rain fades. Packets that are successfully received at the satellite are then transmitted in the downlink channel to the gateway node. The gateway node then forwards the packets across the Internet "cloud" to the appropriate web server. Packets will experience random delays as they make their way to and from the server. The method for emulating end-to-end Internet delays will be described in detail in Section 5.4.8. For the case where we are simulating Web browsing, the client measures the time from the start of the HTTP transaction until its completion. This measurement will serve to quantify performance of the three MAC schemes. Similar transfer time measurements will be taken to asses each MAC protocols performance for bulk transfers and interactive computing. This entire simulated network was again implemented using the discrete event network simulator **ns**. We had to add the MAC layer functionality by modifying the **TDM Controller** mentioned in Chapter 3, Section 3.4. We also added the necessary code to accurately model HTTP 1.1 mechanics (see Section 5.4.5). With this simulation tool enhanced, we experimented with Web transfers, bulk uplink transfers, and interactive computing using both Slotted and Reservation ALOHA, and RIMA.

5.4.1 Implementation of Slotted ALOHA

Slotted ALOHA is a time synchronized version of the ALOHA protocol first proposed in [26]. In our simulation, we implemented it in the following manner. When a packet arrives at the MAC layer of an earth terminal, it is transmitted immediately. After transmission, a timer is set that will expire one satellite round-trip time in the future. A satellite round-trip time is twice the delay between the earth terminals and the satellite. If the packet arrives successfully at the satellite, the satellite sends an acknowledgement packet in the downlink channel. When the user receives the acknowledgement, it cancels the timer and considers the packet successfully received. If a collision occurs on the channel, no packet is received at the satellite. When the round-trip time timer expires, the terminal enters the *backoff* state. It chooses a random backoff time according to either an **linear** or **exponential** model. In the linear case, the backoff time is chosen according to $\mathcal{E}(i\lambda)$, where $\mathcal{E}(\xi)$ is an exponential random variable with mean ξ and i is the number of successive collisions. For the exponential case, the backoff time is chosen according to $\mathcal{E}(2^{i-1}\lambda)$. These two techniques mean that for each sequential collision, the variable i is incremented and then a random backoff time is chosen according to either the linear exponential manner. When the packet is successfully received at the satellite and an acknowledgement is received, i is reset to one. The difference between these two backoff strategies is that for the linear method, the mean backoff time grows linearly with each successive collision, and in the exponential case, it grows exponentially. Figure 5.6 shows a state diagram of how S-ALOHA was implemented for the purposes of our simulation.

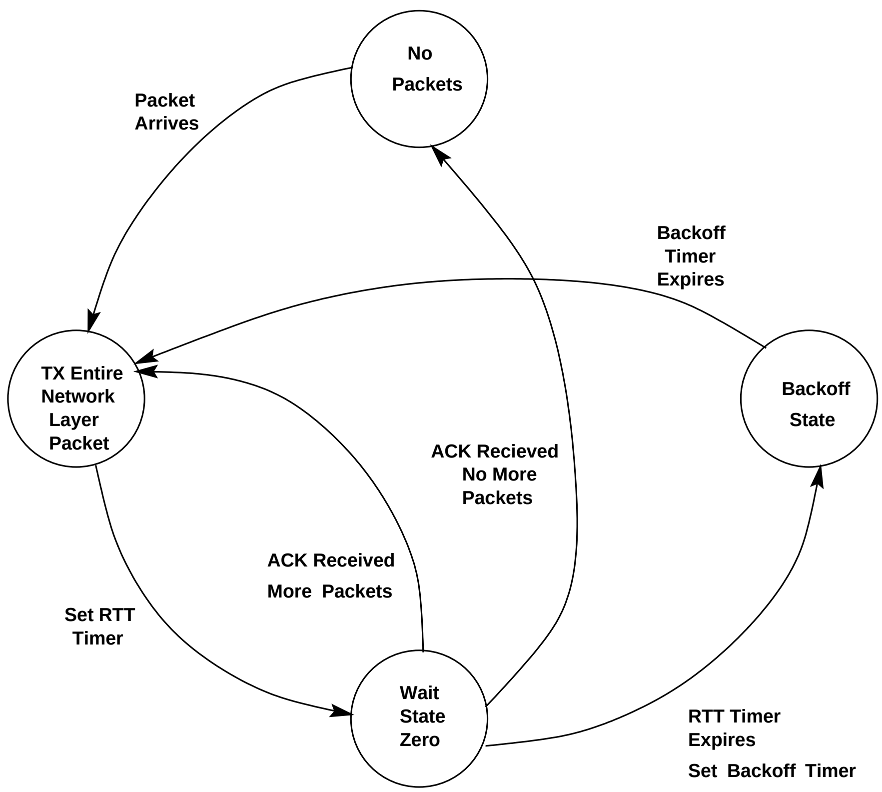

Figure 5.6: State Transition Diagram for S-ALOHA

5.4.2 Implementation of Reservation ALOHA

Our implementation of R-ALOHA works in the following manner. The satellite is constantly broadcasting who "owns" slots in the uplink channel. If a earth terminal successfully transmits a packet in an uplink slot, that terminal owns that slot in the subsequent time division multiplexed (TDM) frame. A TDM frame therefore is equal to one satellite round-trip time. Packets arriving at the earth terminal are placed in the *outgoing* queue. When the next slot interval arrives, the slot can be classified as one of three states; ***EMPTY***, ***MINE***, or ***OTHER***. ***EMPTY*** means that in the previous frame, either no earth terminal attempted transmission or more than one did resulting in a collision. ***MINE*** means that the slot belongs to this particular earth terminal due to a successful transmission in the previous TDM frame. ***OTHER*** means that the slot is reserved by another terminal. Given that there are packets to transmit, the earth terminal chooses to transmit with the following contention probability P_c.

$$P_c = \begin{cases} 1 & if \quad \text{slot} = \text{MINE} \\ 0 & if \quad \text{slot} = \text{OTHER} \\ 1 - e^{\alpha \cdot \text{QL}} & if \quad \text{slot} = \text{EMPTY} \end{cases} \tag{5.2}$$

where:

QL = the size (in bytes) of both the *outgoing* and *retransmission* queues

α = the aggressiveness parameter

This method insures that when a terminal has packets to send, it will always send if the slot belongs to him, never if it belongs to another terminal, and if a slot is empty, it will contend with a probability that grows with the number of packets queued at the terminal. The slot ownership broadcast from the satellite serves as an implicit acknowledgement. If a terminal transmits a packet on an empty slot, and one round-trip time later that slot is described by the satellite as ***EMPTY***, the terminal knows that the packet collided and it places the packet in the *retransmission* queue. The state diagram for our implementation of R-ALOHA is shown in Figure 5.7.

5.4.3 The Time-Bandwidth Product Lower Bound

Since TCP uses slow start when injecting packets into the network, the capacity of the concatenated links that make up the end-to-end path is not fully utilized until the congestion window fully opens. The congestion window cannot fully open until several round trip times after the beginning of slow start. If the end-to-end path has a large time-bandwidth product (TBP, i.e. the number of *packets in flight*), then it will take TCP a substantial amount of time before it is able to "fill the pipe" with packets. The TBP is simply: $bandwidth \times$ *round-trip time*. Satellite networks have high TBPs because of the latency from the earth terminal to the satellite is large. Because of this, TCP performance tends to be worse than terrestrial links [3]. When measuring the effect that the MAC protocol

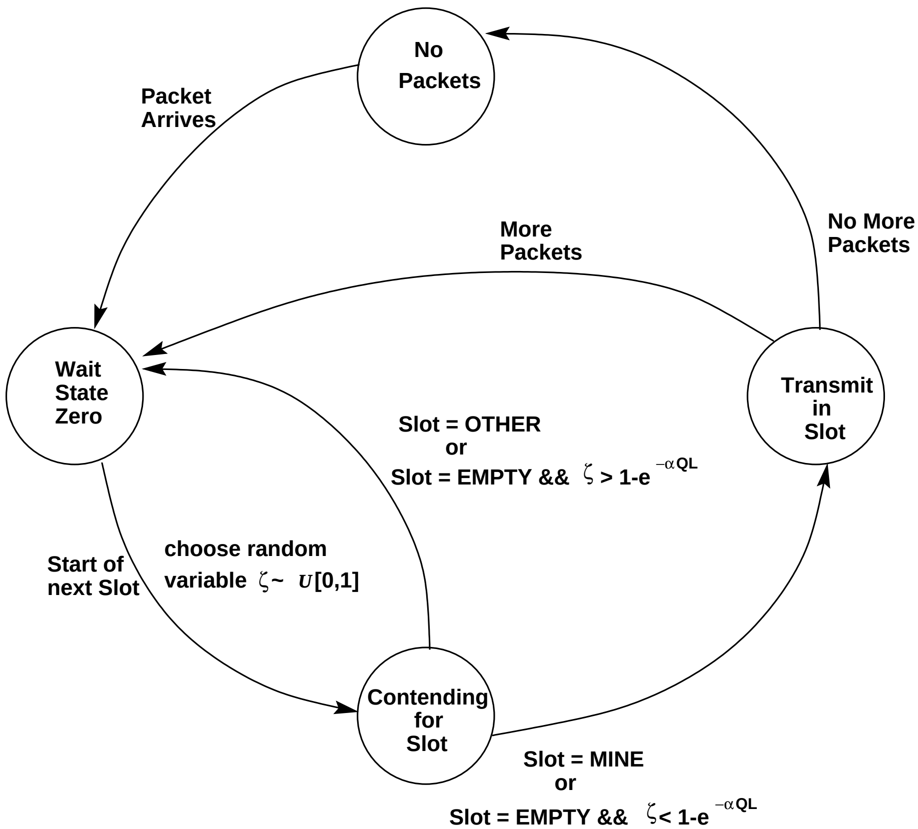

Figure 5.7: State Transition Diagram for R-ALOHA

has upon the performance of any application, one must "subtract out" the effect of TCP. Since the flow control mechanisms of TCP only affect file transfers, we need only concern ourselves with the effect that TCP has upon bulk transfers into the network (i.e from client to server) and on Web page downloads (i.e. from server to client). In order to cancel out the effect of TCP, we need to perform the identical simulations, with the multiple access protocol removed. We can accomplish this by simply multiplexing all of the clients on a single uplink channel. This is illustrated in Figure 5.8. In doing this we retain the delays induced by multiple connections sharing the same channel and the delays caused by the time-bandwidth product of the link at the same eliminating the effects of multiple access collisions. When we performed these simulations, we were able to measure the *Time-Bandwidth Product (TBP) Lower Bound*. This lower bound tells the minimum time it takes TCP to transfer data across the network. When presenting the MAC protocol performance for Web transfers and bulk file transfers in Sections 5.5.3 and 5.5.4 respectively, the TBP Lower Bound will also be given so as to see the effect the medium access technique has on performance.

5.4.4 Empirical HTTP Measurements as simulation Input

In order to drive a simulated HTTP session, the statistics of the Web transfer process must be known with a reasonable degree of accuracy. For the purposes of our simulation, we have relied on the measurements taken in [32]. In this work, the HTTP packets arriving to and leaving from a Web server at the University of California at Berkeley were collected.

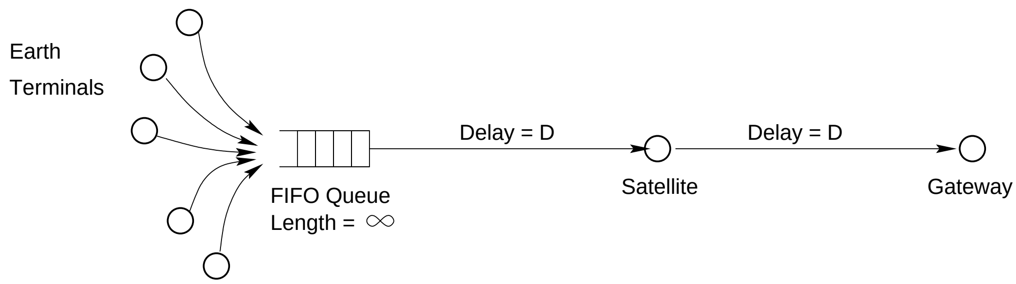

Figure 5.8: Measuring the Time-Bandwidth Product Lower Bound

This data was then processed to produce a set of cumulative distribution functions (CDFs). The statistics relevant to our simulation work were: client think time, HTTP GET length, HTTP REPLY length, inline object count, and consecutive file retrieval count. The client "think" time measures the time between successive initiations of a Web transfer. This represents the time the user needs to examine the Web page and decide to retrieve another. The HTTP GET length measures the size of the individual GET messages. The HTTP REPLY length measures the size of the individual requested pages or inline objects. The inline object count records the number of inline objects requested per Web transfer and the consecutive file retrieval count records the number of successive "hits" to the same Web server. The average of all of these statistics is shown in Table 5.1. For a more complete description of these measurements, please refer to [32].

These CDF's have been integrated into the network simulator ***ns***. Using them allows us to simulate HTTP sessions over satellite channels using the empirical measurements. We have chosen to use all of the above statistics except for one, that being the user think time.

Table 5.1: Summary of HTTP Statistics

Parameter	ThinkTime	GET	REPLY	Objects	Consecutive Hits
Avg.	1313	398	17932	2.9	4.1
Units	seconds	bytes	bytes	none	none

The CDF of the user think time in [32] shows a heavy tail and an average value of 1313 seconds, or approximately 22 minutes between successive Web transfers. These numbers intuitively make sense, however since we are interested is system capacity assessments, we would be interested in a "busy" Web client. We therefore have chosen the user think time to be exponentially distributed with a mean of 60 seconds.

5.4.5 Mechanics of a Simulated Web Transfer

The procedure for a HTTP transaction follows exactly the means outlined in Section 5.1.2. When a HTTP client emerges from thinking, it issues a DNS message to the DNS server located at the gateway node. The gateway node responds with the desired IP address. The client then initiates a TCP connection with the server using the TCP connection establishment procedure (i.e. SYN, SYN+ACK, ACK) described in Section 5.1.1. Once the connection is established the client sends the initial HTTP GET message. The size of this GET message is distributed according to [32]. The server responds with a REPLY, also distributed according to the measurements in [32]. This procedure continues until the initial

Web page and any inline images are completely transferred, according to the procedure of HTTP 1.1. In order to mimic sequential "hits" from a client to a server, when a server is first accessed an integer random variable, h, is chosen according to the CDF of consecutive file retrievals presented in [32]. Every time the client emerges from the thinking state, h is decremented. While h is greater than zero, no DNS lookup is performed prior to TCP connection establishment. Once h reaches zero, a DNS lookup is performed for the next Web transfer and h is randomly re-chosen.

5.4.6 Mechanics of a Bulk Transfer into the Network

The procedure for generating a bulk transfer from an earth terminal to a remote server was done by simply using the `ftp` agent in **ns**. This agent is attached to a *TCPAgent* at an earth terminal and produces a message equal in size to the bulk transfer. This message is passed to the *TCPAgent* to be transmitted to the remote server via the satellite uplink channel. Once the message is completely transferred to the remote server, the `ftp` agent becomes dormant for a exponentially distributed period of time, so as to let the satellite network reach steady state. One earth terminal was designated as the "terminal under test" and all other terminals were generating exponential background traffic. We would have liked all other terminals to be producing HTTP traffic in the background, but this proved to be too simulation intensive.

5.4.7 Method for "Pinging" the Network

Attempting to model an end user using an application such as `Telnet or rlogin` across a satellite uplink channel is difficult. Since the end user is typing and interacting with the server, modeling this behavior accurately is a challenge. Adding to this challenge is the fact that high latencies across the client-server connection effect the end users behavior. Most people when they type on a computer expect to see the characters that they are typing echoed on the screen. Most interactive console applications like `Telnet` and `rlogin` will not echo the character until the packet carrying the character is received at the server and the server acknowledges receiving the packet. If it takes a long time to see the character echoed, then the user will type and interact with the server in a different manner. Because of this, we simply wanted to see how open the collision space (mentioned first in Section 5.2) was on the uplink channel, since the size of the collision space is dependent upon the performance of the MAC protocol and is a good indicator of the overall performance of an interactive application. We therefore made an addition to **ns** in the form of a module that mimics the UNIX command `ping` [49]. This module acts as an agent, attached to a earth terminal, that sends a fixed sized packet into the uplink channel at fixed intervals of time. If the `ping` packet is received at the satellite, it is forwarded down the satellite-to-gateway link of Figure 5.1 to the gateway. Attached to the gateway node is another `ping` agent that replies with a packet of equal size. When the reply packet is received at the sending earth terminal, the round-trip time is measured. This serves to probe the uplink channel to see how large the collision space is for a particular MAC protocol under a given loading condition.

5.4.8 End-to-End Internet Path Emulation

The primary aim of this simulation is to measure the effect of the satellite medium access control protocol on HTTP, `Telnet`, and `ftp` performance, but since packets crossing and end-to-end connection will experience delays due to the switches and routers that make up the Internet, we needed to model, in a somewhat abstract manner, this interaction. Following the claim made in [38], we chose to model the round trip times of the packet crossing the Internet "cloud" of Figure 5.1 as a shifted gamma distribution. A gamma(α,β) random variable has the following distribution.

$$f(x) = \begin{cases} \frac{\beta^{-\alpha} x^{\alpha-1} e^{-\frac{x}{\beta}}}{\Gamma(\alpha)} & if \quad x > 0 \\ 0 & otherwise \end{cases} \tag{5.3}$$

where $\Gamma(\alpha)$ is the gamma function, defined by $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$.

In order to parameterize this distribution, we conducted a series of `ping` measurements on hosts at various distances. Table 5.2 shows the statistics of each host and the two parameters, α and β, that made the round trip time statistics match a gamma distribution. Using these results we implemented and Internet Path Emulator (IPE) that worked in the following manner. Each packet sent from a client has a flow identification, f_{id}^c. When the consecutive hit variable h described in the previous section reaches zero, the client increments f^c_{id}. This signals the IPE that the client is accessing a different server at a different distance (in terms of hops) from the gateway node. The IPE then chooses a new minimum round trip time m_{id}, and new gamma parameters α_{id} and β_{id}. Thus when packet p_i with flow id f^c_{id} enters the IPE, it sees a random delay d_i, where d_i is chosen as follows.

$$d_i = m_{id} + \nu_{(id,i)} \tag{5.4}$$

where $\nu_{(id,i)} \sim$ gamma(α_{id}, β_{id})

Consistent with Table 5.2, for each new flow id we chose m_{id} uniformly between 25ms and 150ms, α_{id} uniformly between 1.1 and 2.0, and β_{id} uniformly between 1.5 and 17.

Table 5.2: Summary of `ping` Host Statistics

Host Name	RTT Min (msec)	RTT Max (msec)	RTT Avg (msec)	α	β
ees2cy.engr.ccny.cuny.edu	90	415	124	2	16.67
utds05.utmem.edu	143	582	162	2	2.5
www2.ece.uiuc.edu	97	132	102	2	2.5
web.cs.ndsu.NoDak.edu	77	178	95	1.5	14.3
info.tamu.edu	75	236	81	2	2
patch.Colorado.EDU	43	245	45	2	1.4
charlotte.it.wsu.edu	51	167	63	2	3.33
info1.ucdavis.edu	26	281	30	2	2.5
www.mit.edu	93	266	103	2	2.5
www.ee.ucla.edu	29	251	31	2	1.43
vole.eng.fsu.edu	90	207	101	1.5	6.66
rouge.engr.wisc.edu	97	192	102	1.1	3.33
www.umf.maine.edu	95	331	103	1.1	2
ece00ws.ece.ncsu.edu	88	369	90	1.5	3.33

5.5 Simulation Results

In this section, we will present the extensive simulation work conducted using our modified version of **ns**. Since we are interested in the performance of the uplink portion of the satellite link, we have chosen a heavily asymmetric (in terms of bandwidth) duplex link. We have set the downlink bandwidth to be 10 Mb/s while only allowing for 100 Kb/s in the uplink. This will allow our simulation results to predict the number of terrestrial users that can be supported per 100 Kb/s of uplink bandwidth. We would liked to have chosen greater bandwidths for both the up and down links, but the limitations of computer simulation prohibited this. The delay across the satellite channel was again chosen to be 135ms. Our primary aim with this simulation work is to determine the performance of RIMA, R-ALOHA, and S-ALOHA with respect to the delivery of HTTP traffic. The goal was to see the number of earth terminals that could be supported in a given amount of uplink bandwidth. With this in mind, we defined a *heavy load* to be 300 earth terminals. To determine what constituted a heavy load, we applied the following calculus. According to [32], the average bytes transferred from a server to a client during a typical Web download was $2.9 \cdot 398 = 1,154$ bytes. Assuming that a transaction occurs on average every 60 seconds, this means a client has an average bit rate of $\frac{1154 \cdot 8bits}{60sec} = 154bps$. This is not taking into account the TCP acknowledgements that will be flowing in the uplink (which are non-negligible). It is well known that the Slotted ALOHA channel has a throughput of 0.36, then taking a 100Kb/s channel at 36% throughput gives us 36 kb/s. Dividing 36 kb/s by

154 b/s, gives us 234 supported users. Thus to simulate a heavy load, the active user count was set to 300 HTTP clients per 100 Kb/s of uplink bandwidth. To see how performance changes with load, we have also conducted simulations with the earth terminal count set to 100 and 200. We have also varied the physical layer *granularity* to see what if any effect that has on the performance of each protocol. By granularity, we mean the number of PHY layer cells per TDMA slot. Ideally one would like to have high granularity in the TDMA framing structure as it reduces the collision space (i.e. a Network Layer packet could be mapped to the minimal time slot profile). In reality though, systems are designed with much lower granularity so as to reduce the bandwidth lost to PHY Layer overhead factors, such as preambles for synchronization. We set the PHY cell size to be 50 bytes (close to a 53 byte ATM cell) and ran simulations with 1, 2, 4, and 8 cells per slot. All of the relevant physical layer statistics are summarized in Table 5.3.

This section detailing our simulation results is ordered as follows. First we conducted a series of tests under heavy loading conditions to determine the best backoff strategy and the best backoff seed for Slotted ALOHA. Next, we simulated Reservation ALOHA under heavy loading to determine what is an optimal aggressiveness parameter. Once the simulation parameter space was established, we then experimented with Web browsing, uplink file transfer, and uplink `ping` measurements.

Table 5.3: Summary of Physical Layer Statistics

Parameter	Value
Uplink Bandwidth	100 kb/s
Downlink Bandwidth	10 Mb/s
TDMA Frame Length	20,40,80,160 mSec
Cells per TDMA Slot Uplink	1,2,4,8
TDMA Slot Length	4,8,16,32 mSec
PHY Layer Cell Size	50 bytes
Slots per TDMA Frame	5
GEO RTT Delay	270 mSec

5.5.1 Backoff Strategies for Slotted ALOHA

Referring back to Section 5.4.1, the first thing to determine was for S-ALOHA, which backoff strategy, either **linear** or **exponential** was superior. Figure 5.9 shows the performance of both techniques in a GEO (i.e. 270 ms round trip time) channel, under heavy HTTP loading. The x-axis shows the web page size (i.e. the size of the entire HTTP transfer) and the y-axis shows the average time to completely transfer the file. Figure 5.9 clearly shows that for a GEO channel, an exponential backoff strategy vastly outperforms a linear strategy. For this reason, we have eliminated the linear backoff strategy, and focused only on exponential.

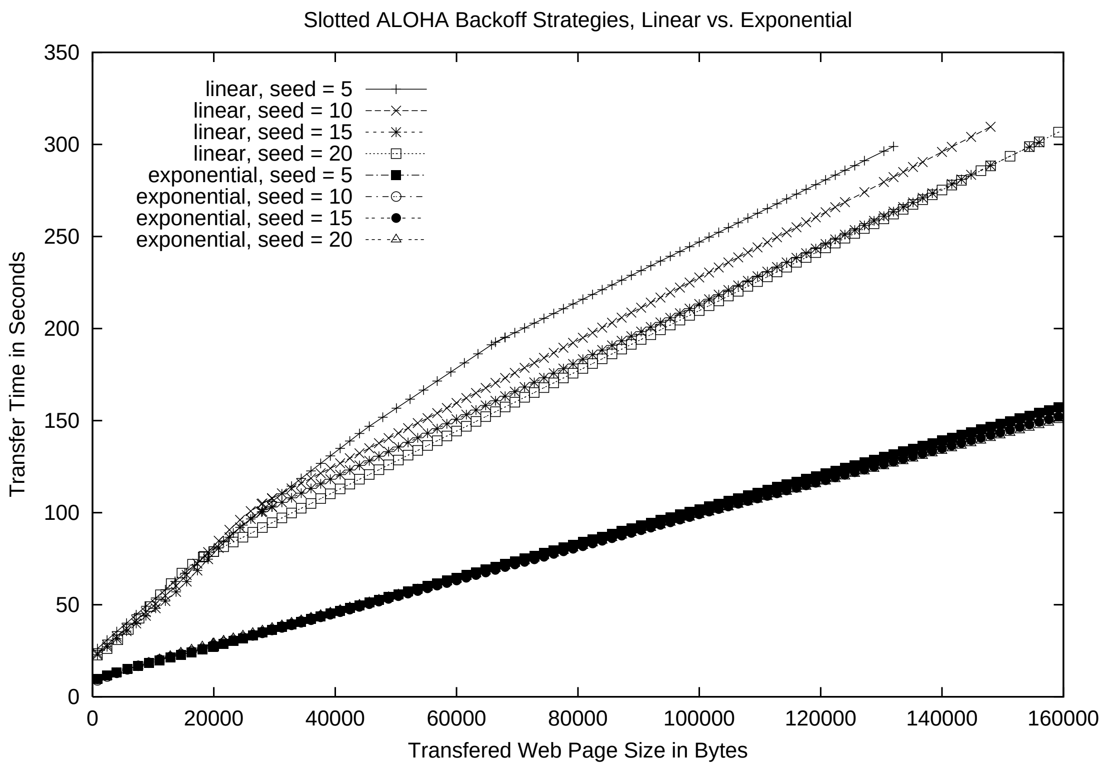

Figure 5.9: Comparing Backoff Strategies for Slotted ALOHA

The next issue for S-ALOHA, is that given an exponential backoff technique is to be employed, what is the optimal initial backoff "seed" λ, mentioned in Section 5.4.1. It would be reasonable to think that this λ should increase with increasing round trip latencies, since collisions are more costly as the time to acknowledge a collision increases. We therefore conducted a series of experiments, again under the heavy loading condition of 300 active users, where we varied the satellite one-hop delays. We chose delays across the channel of 7.5, 30, and 135 msec, with initial backoff seeds, λ, of [2,4,6,8,10], [6,8,10,12,14], and [10,12,14,16,18] slots respectively (7.5 msec would represent a low earth orbit delay and 30 msec would be typical of a medium earth orbit delay [33]). The Figure 5.10 shows the results. It would appear that in at least an average sense, the initial backoff seed is not as important as backoff strategy, since there appears little difference between the performances for all three channels. Therefore we have chosen a minimum seed of 10 slots for all subsequent simulations. The reasoning behind this is that choosing a smaller initial backoff seed will reduce backoff time and increase throughput. If this aggressive approach is shown to not suffer at high loads, then it should intuitively work better at lighter loads.

5.5.2 Determining the Aggressiveness Parameter for R-ALOHA

Now we turn our attention to R-ALOHA. The unknown simulation parameter is the aggressiveness parameter α. This parameter determines, along with the output queue length, how likely a terminal is to contend for an empty slot. After some "back of the envelope" calculations, we determined to simulate R-ALOHA, with an α that varied over an order of

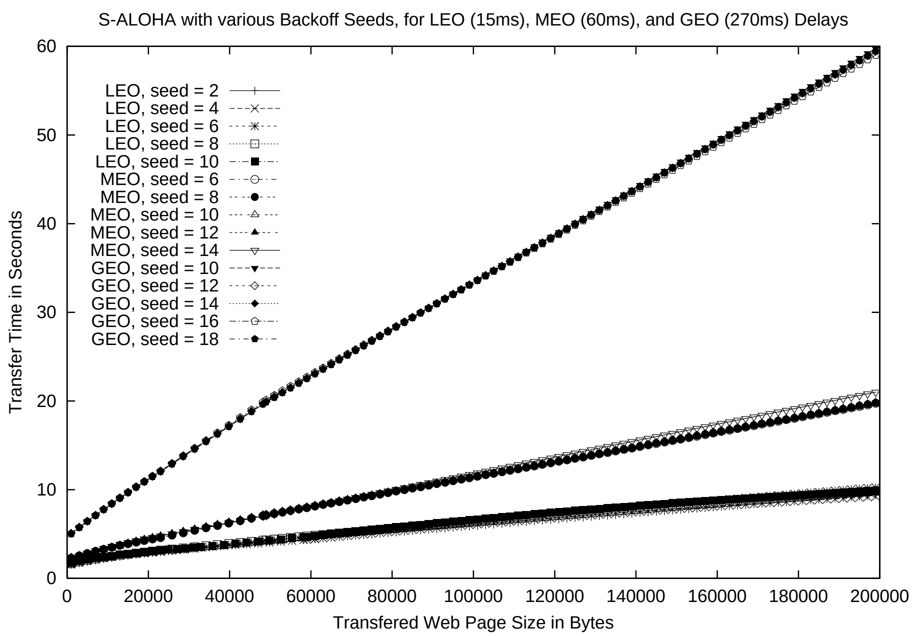

Figure 5.10: Comparing Initial Backoff Seeds

magnitude, 0.01 - 0.5. Again, all of these simulations were conducted under heavy HTTP network loading. Figure 5.11 shows the result. In a 270 msec GEO channel, a conservative α of 0.01 and a more aggressive value of 0.1 appear to achieve the same performance. This shows that both a greedy and a conservative approach achieve the same performance, in the average sense, under heavy loading. We will again side with the aggressive approach and choose α to be 0.1 under the intuition that with lighter network loads, an aggressive strategy will produce better throughput.

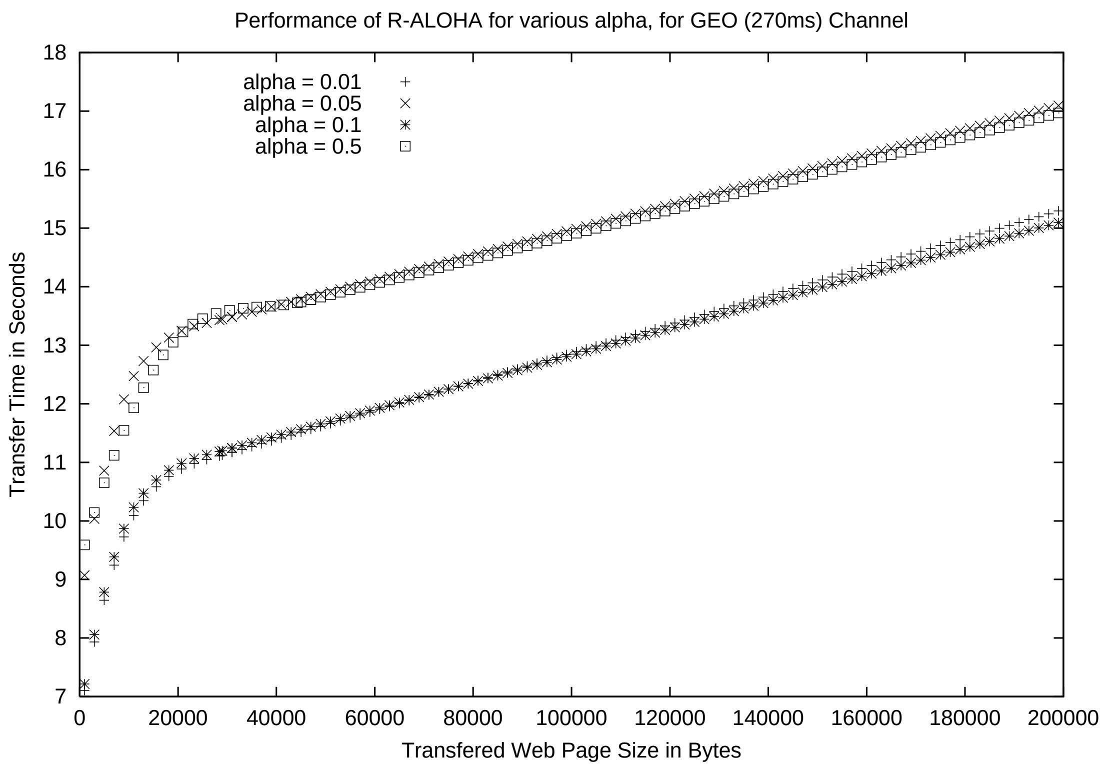

Figure 5.11: GEO Channel performance with respect to the Aggressiveness Parameter, α

5.5.3 Simulation Results for Web Page Downloads

Extensive simulations to determine the performance of the three MAC protocols under HTTP traffic were then conducted. We set the number of active earth terminals to be 100, 200, and 300. As indicated earlier, we consider 300 active earth terminals to be a ”heavy load”. The choice of 100 and 200 terminals was made to show the relative performance as a function of network load. After conducting simulations at these points for Slotted ALOHA it became clear that the uplink had become saturated and that there was no perceptible difference between 100, 200, or 300 active terminals. We therefore repeated the simulations with 20, 50, 100, and 150 active terminals, focusing on performance under load and not taking into account the PHY Layer granularity (i.e. we just looked at the four cell per slot case). Figure 5.12 shows the results with the x-axis again showing the retrieved web page size in bytes and the y-axis showing the average download time in seconds. Even when the active terminal count was as low as 20, the network was still saturated. Fifty, 100, and 150 active terminals show almost identical performance to one another. The reason that the channel becomes saturated, instead of ”blowing up” and becoming unstable is that we set an *Abnormal Timeout* timer to just over a minute (75 seconds). This represents an end user “giving up” on the download if it does not complete in 75 seconds. The average download times for Slotted ALOHA appear to be unbearably long (compared to the performance of R-ALOHA and RIMA, as will be shown shortly), even for those transactions fortunate enough to complete in under 75 seconds. Since our primary application that we would like the MAC to support is Web browsing, we eliminated Slotted ALOHA from further consideration. The results in all remaining sections will only consider Reservation ALOHA and RIMA.

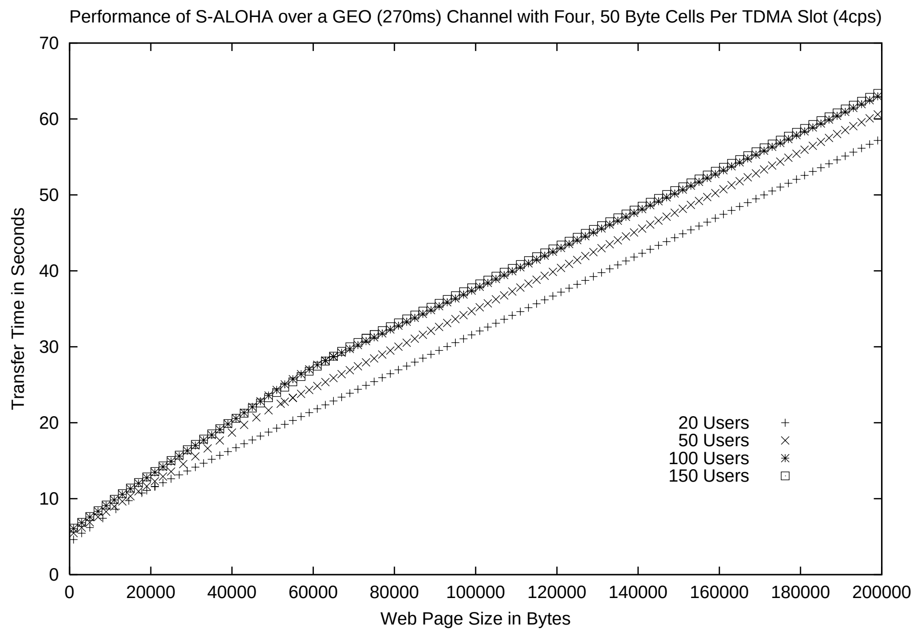

Figure 5.12: The Performance of S-ALOHA with multiple number of Active Users

Figures 5.13 through 5.18 show the HTTP transfer performance for Reservation ALOHA and RIMA. Both of these techniques produce much more tolerable transfer times. Included in each figure is the Time-Bandwidth Product (TBP) Lower Bound. This was generated using the technique outlined in Section 5.4.3 and includes the Internet Path Emulator (which implies that several transactions were conducted and the results were averaged to produce the lower bound curve). For all of the figures, there is a curved region and a linear region. The curved region is due to TCP slow start, mentioned in Section 5.1.1, and is perceived only when the Web page size is small. This is because slow start induces significant delay for small transfers, but becomes negligible for larger transfers. For both protocols, as the cell-slot granularity decreases (i.e. the number of cells per slot increases), the transfer time increases. The is understood by considering the notion of a *collision space* mentioned in Section 5.2. The collision space is widened, without increasing the bandwidth, by increasing the TDMA frame's granularity. With high granularity, and thus a wider collision space, PHY Layer cells transmitted in a random access manner will experience fewer collisions. In a high latency GEO satellite system, these collisions increase significantly the transfer time. Considering Figures 5.13 and 5.14, it would appear that under heavy loading (300 active terminals), RIMA comes to within 2 seconds of the TBP Lower Bound with the highest granularity while R-ALOHA exceeds the bound by upwards of 7 seconds. This means that the MAC induced delay of R-ALOHA is three times greater than that of RIMA. When a more realistic, lower granularity scenario is considered (i.e. 8 cells per slot), RIMA is between 3 to 5 seconds above the bound while R-ALOHA is upwards of 15 seconds above the bound. In this scenario, the MAC induced delay of R-ALOHA is also in the neighborhood of three times that of RIMA. The response initiated nature of RIMA clearly shows its benefits over R-ALOHA's implicit reservation scheme. With RIMA, the only cells that are transmitted in a random access manner are those that initiate the connection. This reduces the collision rate and thus reduces the overall transfer time. With R-ALOHA, whenever the output queue is emptied, the terminal loses its reserved slot. Thus when new packets arrive, they must be sent in a random access manner, increasing the likelihood of collision and thus increasing the transfer time.

Figures 5.15 through 5.18 show the same simulations conducted with 200 and 100 active

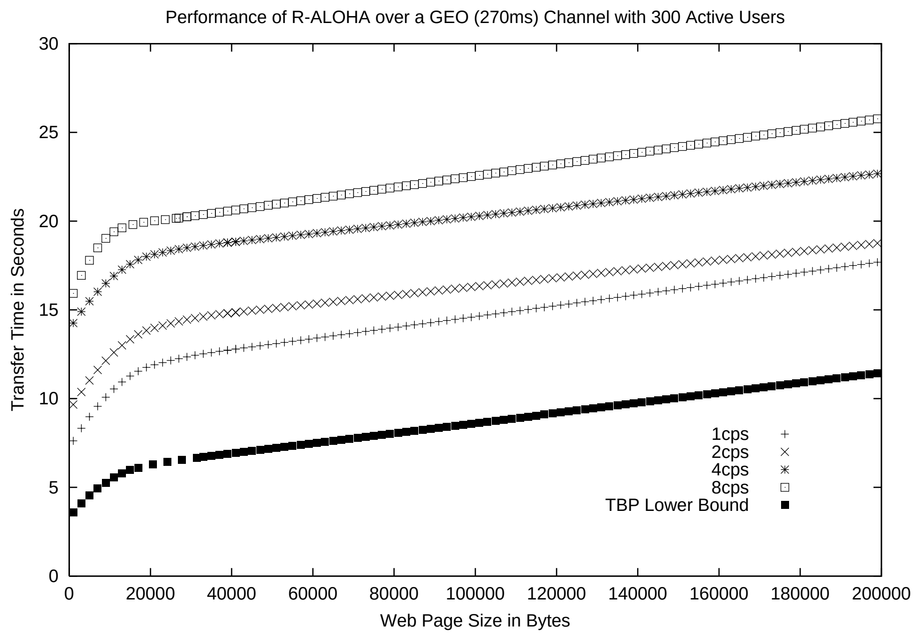

Figure 5.13: The Performance of R-ALOHA with 300 Active Users

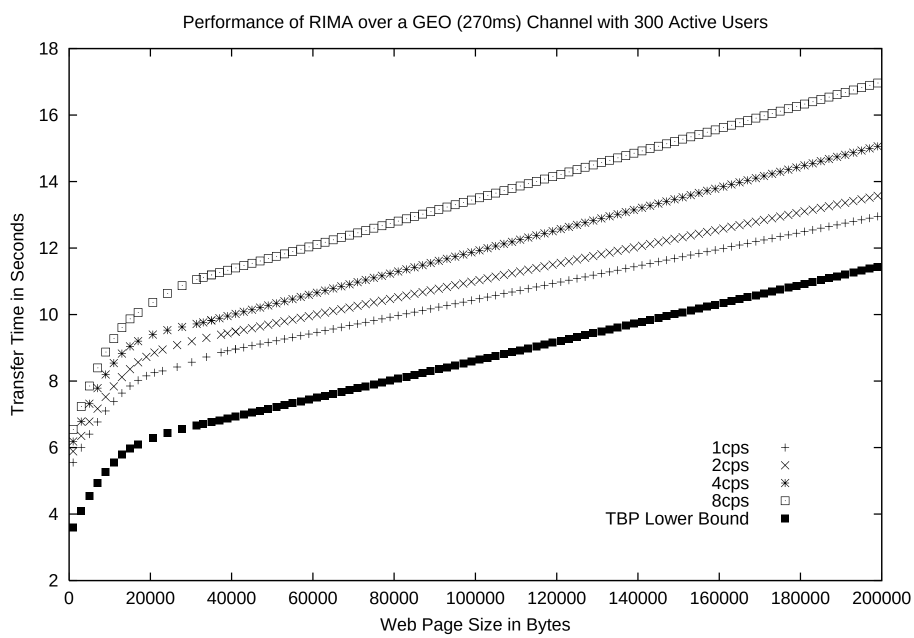

Figure 5.14: The Performance of RIMA with 300 Active Users

earth terminals. Even though the network loading is reduced, RIMA still constantly outperforms Reservation ALOHA. A final thing to consider is that RIMA is immune to high variations in client-to-server round-trip times. We developed our Internet Path Emulator by conducting a series of `ping` measurements in the continental United States. `ping` packets tend to experience better round-trip times due to the fact that they are not ”dropped” as frequently by routers because of their small size (less than 50 bytes). Therefore we would assert that in a more realistic scenario, RIMA would perform the same (with respect to the TBP Lower Bound), while R-ALOHA would deteriorate as the round-trip time variance increased.

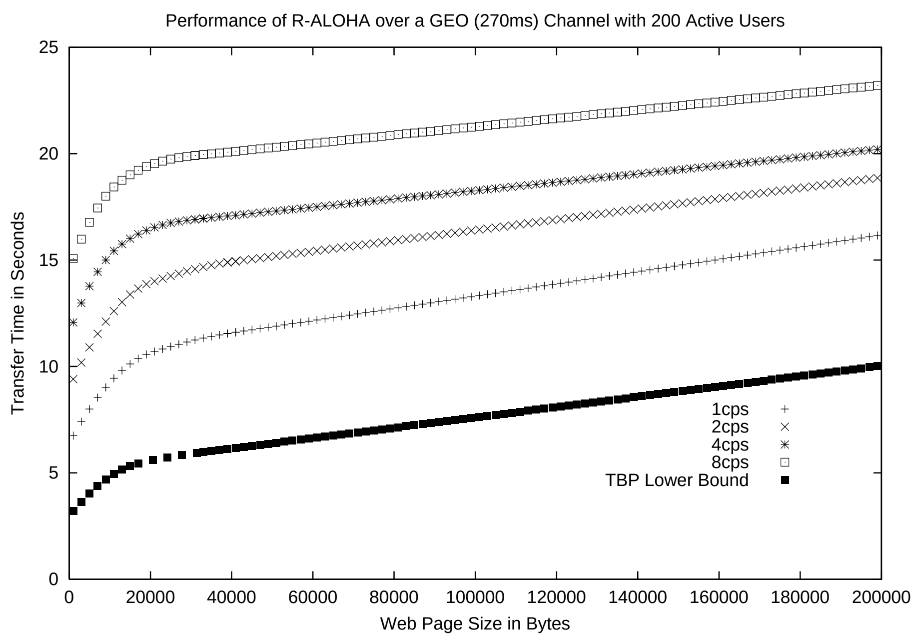

Figure 5.15: The Performance of R-ALOHA with 200 Active Users

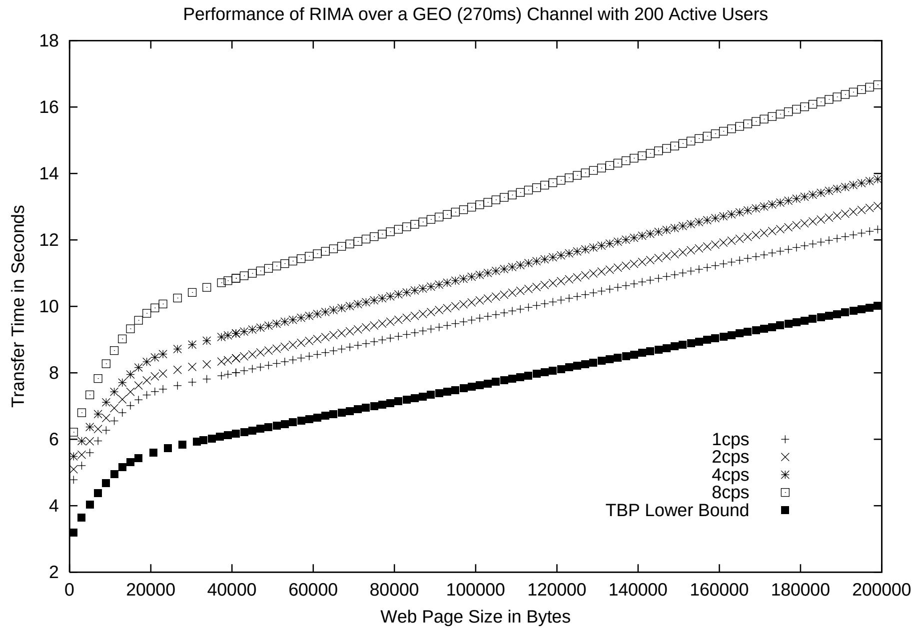

Figure 5.16: The Performance of RIMA with 200 Active Users

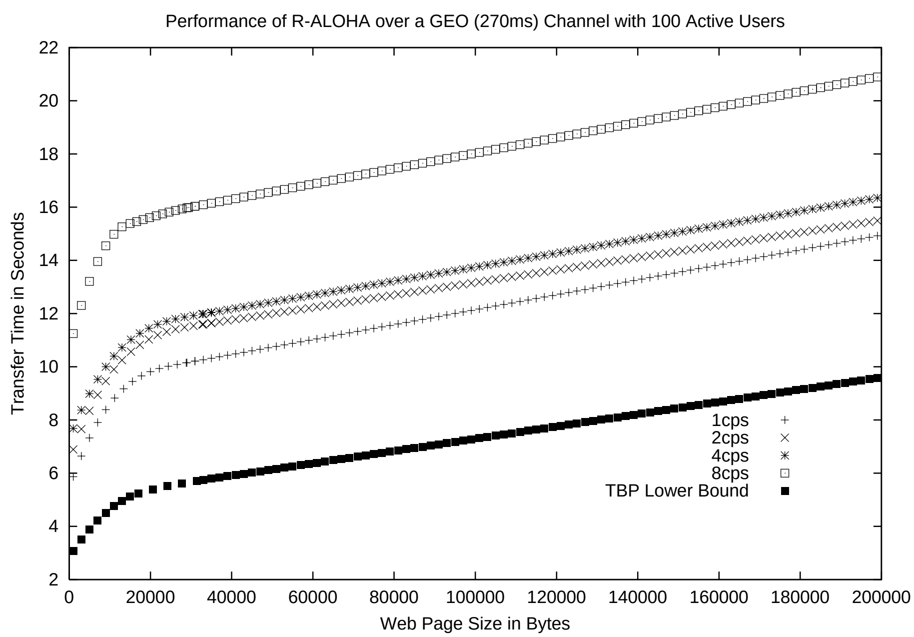

Figure 5.17: The Performance of R-ALOHA with 100 Active Users

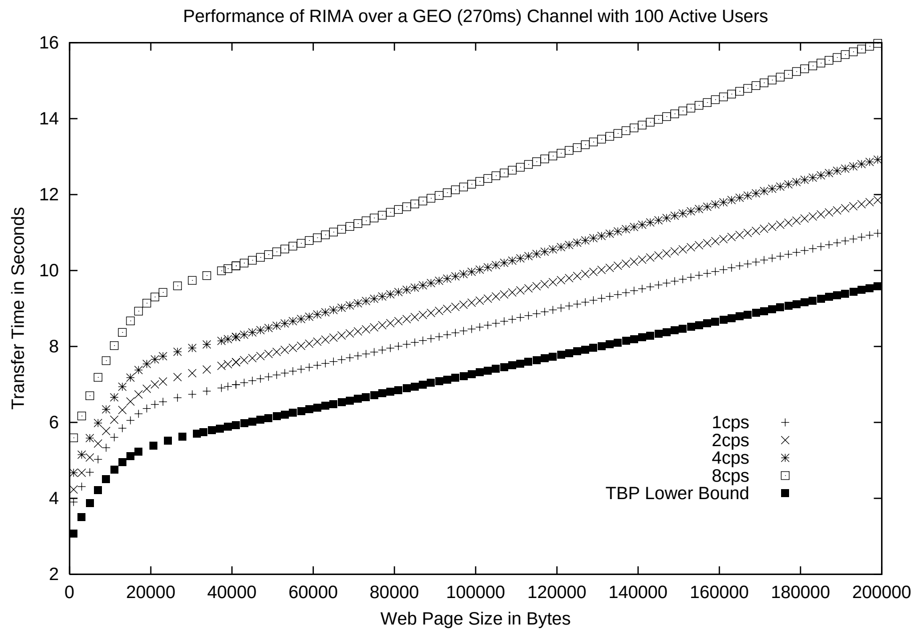

Figure 5.18: The Performance of RIMA with 100 Active Users

5.5.4 Simulation Results for Bulk Transfers into the Network

To assess RIMA and Reservation ALOHA's ability to deliver data into the network (i.e. from the earth terminal to satellite via the uplink channel) we conducted a series of computer simulations with the following conditions. Choosing 101 earth terminals in the satellite's spot beam we selected one terminal to be the source of the bulk transfer and the remaining 100 terminals generated exponentially distributed background traffic of a specified loading. Ideally this background traffic would be HTTP traffic, but in attempting this we discovered that the capacity of the computing power we possessed became exhausted. Since we are primarily concerned with the bulk transfer process, the nature of the background traffic was a secondary concern. The uplink loading due to the exponential sources,

$\mathcal{L}$, is computed according to Equation 5.5.

$$\mathcal{L} = \frac{\lambda \cdot N_{terminals} \cdot |\text{PACKET}_{bytes}| \cdot 8}{BW_{uplink}} \tag{5.5}$$

Where:

λ is the exponential arrival rate

$N_{terminals}$ is the number of terminals in the spot beam

$|\text{PACKET}_{bytes}|$ is the size of the background traffic packet in bytes

BW_{uplink} is the uplink bandwidth

In our case the packet sizes were 50 bytes, the uplink bandwidth was 100 kb/s, the number of terminals was 100, and λ was varied so as to vary the background load. We chose to transfer 10k, 50k, and 100k bytes into the network using `ftp` with a TCP maximum transmit unit (MTU) size of 1500 bytes. The number of actual TCP packets that were needed for the three transfers were therefore 7, 34, and 67. With RIMA, the response initiated and demand assigned multiple access provision (refer back to Figure 5.5) allows for collision free access once the initial packet is successfully received at the satellite. If the initial packet collides, then the MAC state machine will go into an exponential back-off mode just as in Slotted ALOHA. This means of rapidly expanding the backoff time is considered "network friendly" as it allows for the channel to become uncongested when collisions occur. Thus, once the initial packet is received at the satellite, then the remaining transfer time will match the TBP Lower Bound. This is illustrated in Figure 5.19. In the case of Reservation ALOHA, there is no concept of back-off on collision. The probability that a packet enters the uplink channel is simply a function of the output queue size as described in Section 5.4.2. When a packet collides, it is placed back in the output queue and is retransmitted. This means that when the network becomes congested, each terminal aggressively injects packets into the network until it has secured an uplink slot or slots. It will keep these until it has emptied its queue. This means that not only will the original packet be transmitted in a random access manner, but also all packets that lead each TCP slow start interval. The packet numbers will be 1,2,4,8,$\cdots$ as illustrated in Figure 5.20. Because of this, R-ALOHA has more opportunities for collisions than RIMA. This is no longer true as the bulk transfer size increases beyond a certain point. As the size of the transfer grows, TCP acknowledgements will arrive back at the client and new packets will be passed to the MAC sub-layer before it has emptied its output queue. However, the bulk transfer sizes under our consideration are not large enough for this to be the case.

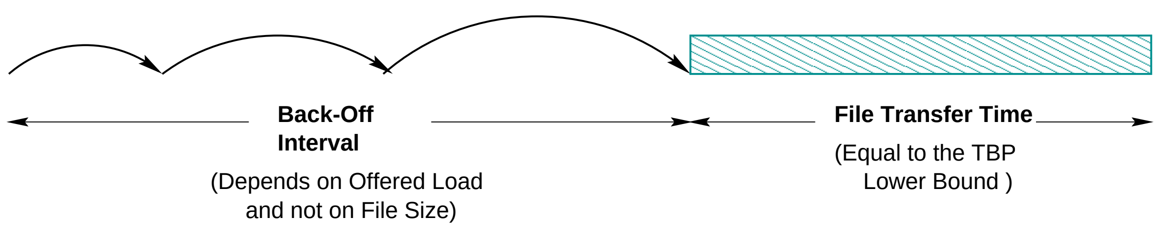

Figure 5.19: An Illustration of the Uplink Bulk Transfer Process for RIMA

Figures 5.21, 5.22, and 5.23 show the simulation results. Each figure has the TBP Lower Bound shown for comparison purposes. It is clear that even for a moderate range of back-

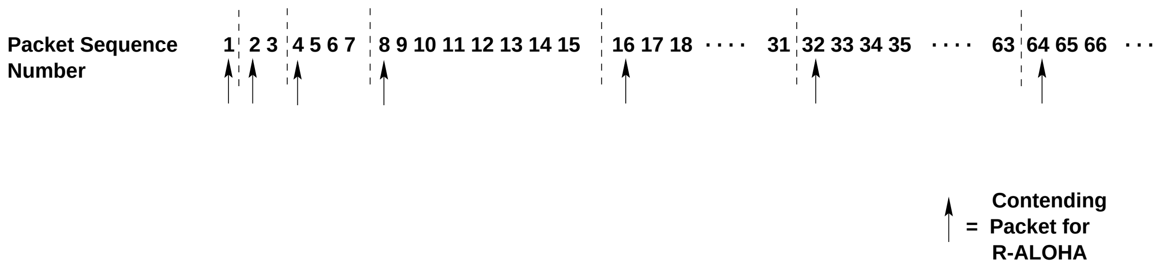

Figure 5.20: TCP Slow Start Packet Sequence Numbers and R-ALOHA

ground loading values, the transfer times rapidly increase. To give a more illustrative representation, we plotted the transfer times in logarithmic form. What can be interpreted from these results is that at light loads, the transfer times do not greatly exceed the TBP Lower Bound. RIMA outperforms R-ALOHA throughout this region due to the fact that it only has one packet subject to potential collisions. As the load increases, R-ALOHA has superior performance. This can be explained by again returning to the exponential back-off mechanism in RIMA. When the background loading becomes higher, collisions become more likely and RIMA will backoff in a network friendly manner. If the chance for subsequent collisions is sufficiently high, RIMA will continue to back off, each time choosing a random back-off value whose mean grows exponentially. This leads to an ever increasing transfer time. In the case of R-ALOHA, the earth terminal continues to retransmit collided packets until a slot is reserved. This aggressive strategy prevails under heavier loads. It is not clear what effect this technique has on the other traffic in the network. Since our background traffic was simply artificially generated, we have no measurement of how the other earth terminals in the network suffer as a result of the R-ALOHA retransmission strategy. One point worth mentioning in passing is that the point at which the transfer times of R-ALOHA become less than RIMA are so large that they are unacceptable. This means that although there is a region where RIMA is outperformed by R-ALOHA, this region is not of much interest as it is outside the QoS targets that any system must deliver.

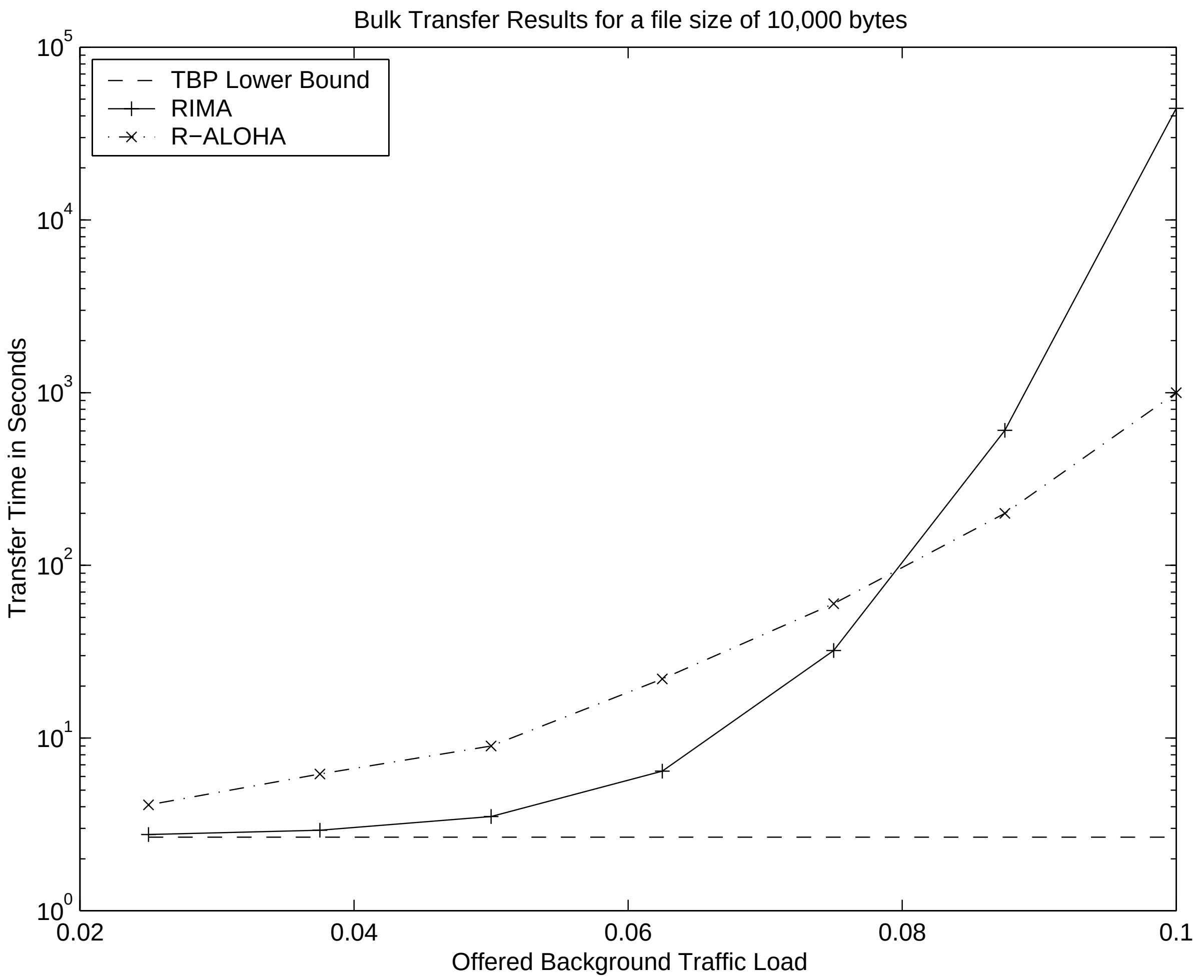

Figure 5.21: Bulk Transfer of 10,000 bytes into the Network

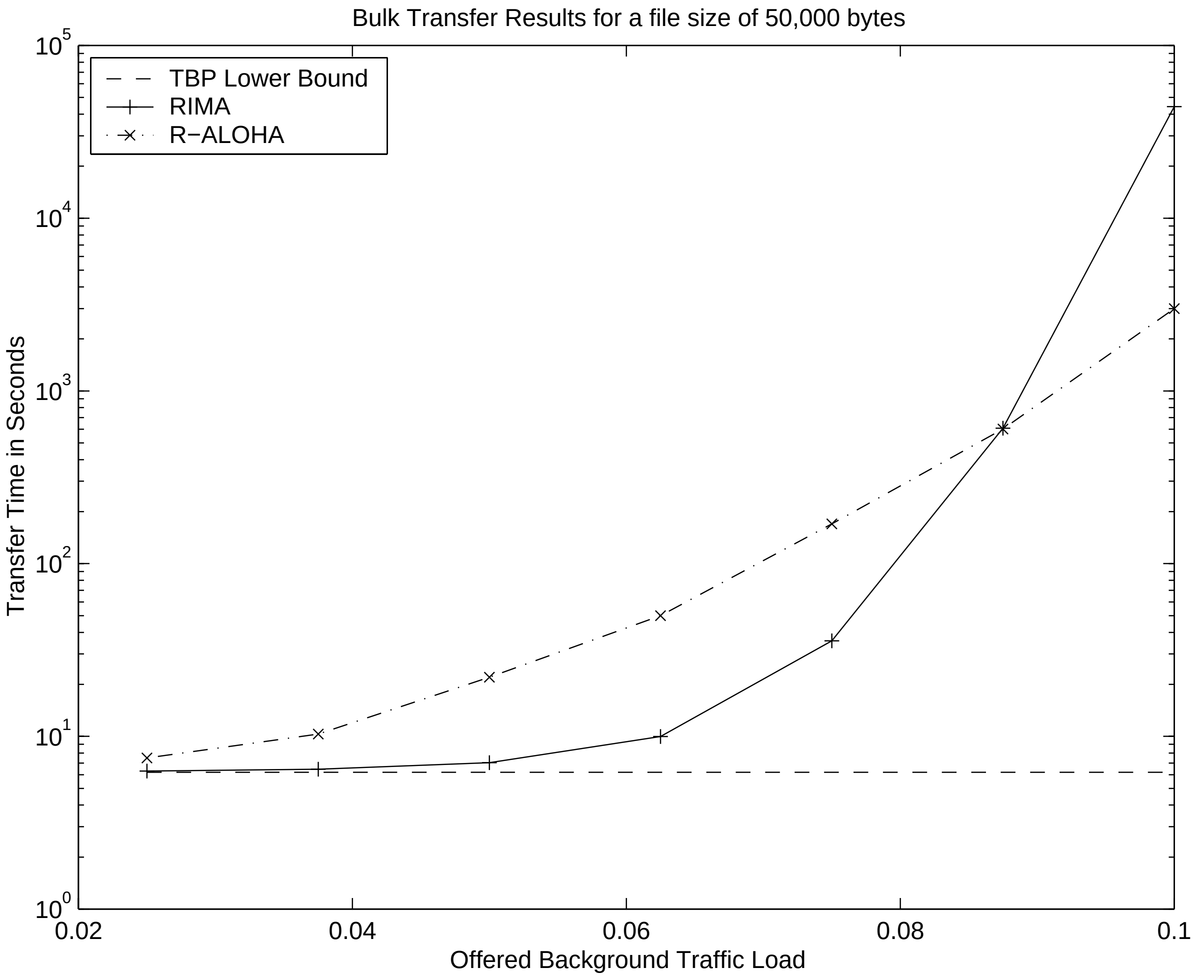

Figure 5.22: Bulk Transfer of 50,000 bytes into the Network

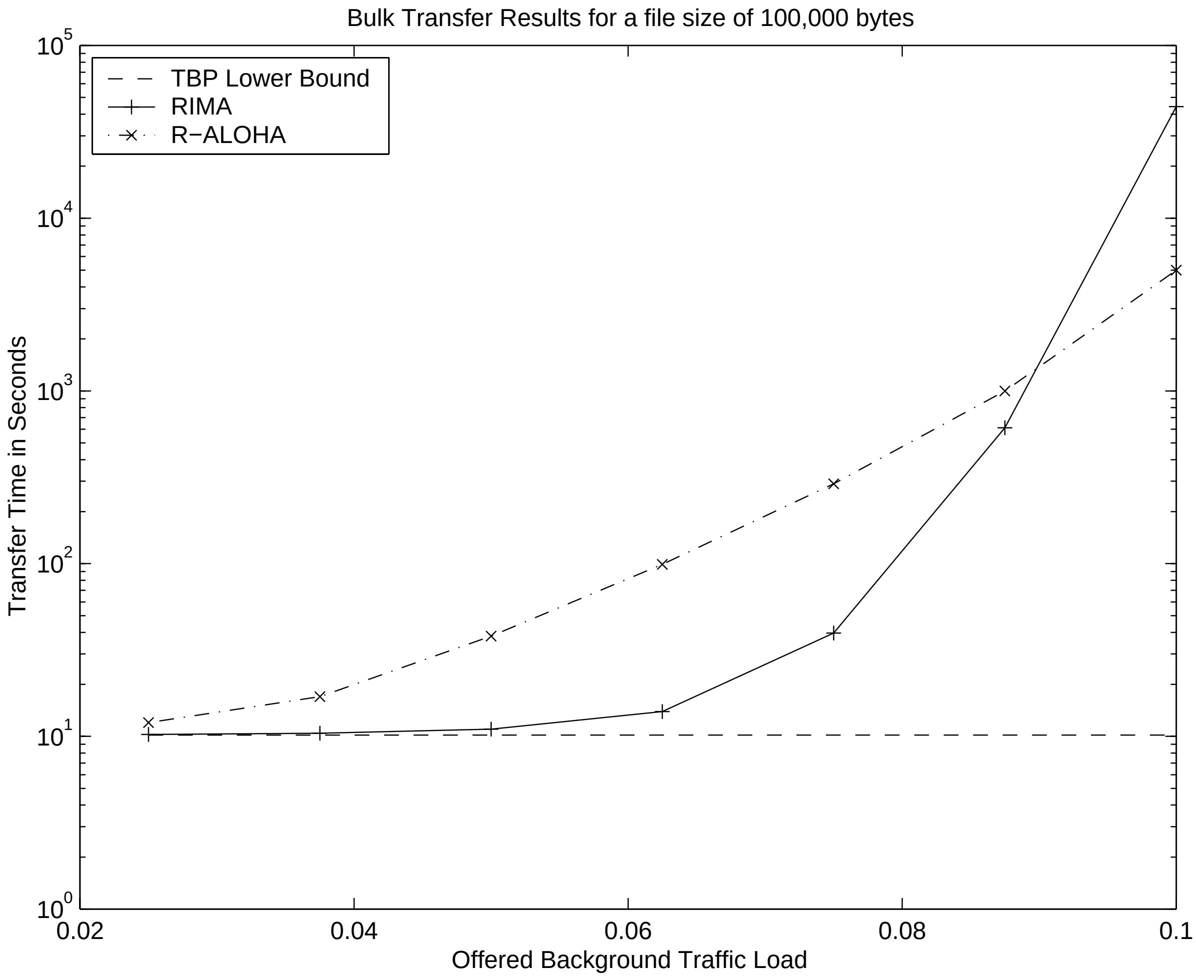

Figure 5.23: Bulk Transfer of 100,000 bytes into the Network

5.5.5 Simulation Results for Interactive Computing

In Section 5.4.7 we described our methodology for determining the feasibility of running an interactive computer data application over a satellite channel with a particular MAC protocol in place on the uplink channel. We implemented this `ping` agent in the following manner. During all the simulations described in Section 5.5.3, there were actually 301, 201, and 101 active terminals. One of these terminals was a `ping` agent, while the remaining were HTTP clients. The `ping` agent sent 50 bytes packets into the uplink channel at ten second intervals. This allowed us to measure the size of the *collision space* with uplink HTTP traffic, in a manner that did not affect the results of the ongoing simulation. In effect we were probing the uplink channel. Once the `ping` agent sent a packet, it started a ten second timer. If that timer expired before a reply was received, it considered that packet lost and transmitted another. If the packet was received within this ten second interval, the round-trip time was measured. If the packet was received after the timer had expired, it was discarded and its statistics ignored. This had the effect of hard limiting all our measured statistics to less than ten seconds, but since we are attempting to measure the performance with respect to interactive applications, then any round-trip time exceeding ten seconds is beyond our region of tolerance. Figures 5.24 through 5.27 show scatter plots of the measured round-trip times for both RIMA and R-ALOHA for link granularities of 1, 2, 4, and 8 PHY Layer cells per slot (cps). What becomes immediately clear is that as the terminal count increases and the granularity decreases, the round-trip time increases. It appears that RIMA has less weight in the region around 10 seconds in the scatter plots.

RALOHA appears to consistently have a significant number of measured values near the 10 second cut-off point.

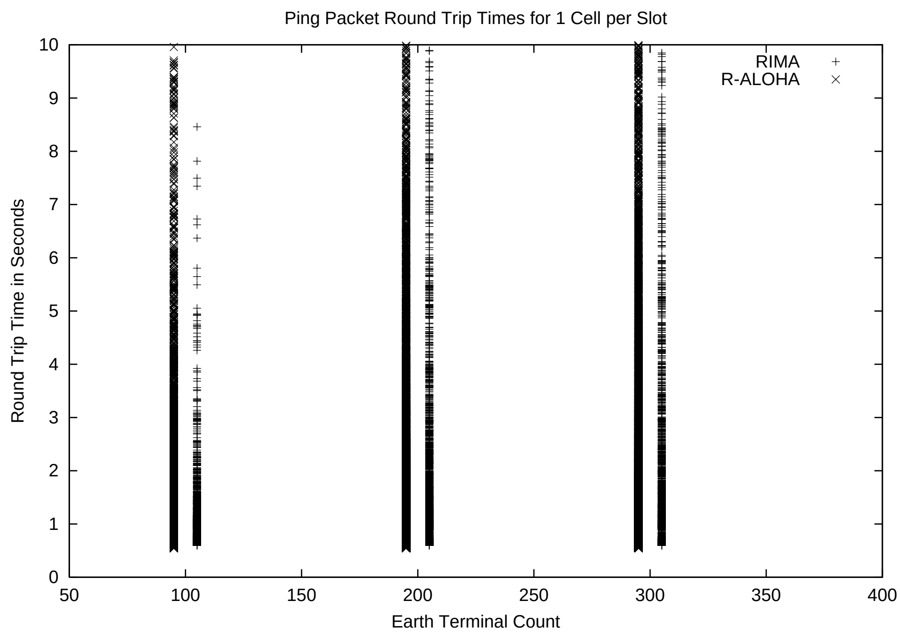

Figure 5.24: A Scatter Plot for Ping Round Trip Times, Granularity of 1 Cell Per Slot

Figure 5.28 shows a histogram of the round-trip times for 1 and 8 cells per slot with 300 active HTTP terminals. It is apparent that RIMA has a single dominant round-trip time value and the weight outside this value appears to be negligible. Reservation ALOHA on the other hand has a strong peak, but has a region of support exceeding this peak. Table 5.4 shows the measured average of the recorded round-trip times. Despite the fact that R-ALOHA's first order statistics are improved by having a hard limit of ten seconds, RIMA still outperforms R-ALOHA by as much as 44%. The values contained in Table 5.4 are all in the 1 to 2 second range. Since the `ping` packet is originating from the earth terminal and returning to the earth terminal, the minimum round-trip time is 0.54 seconds. Therefore

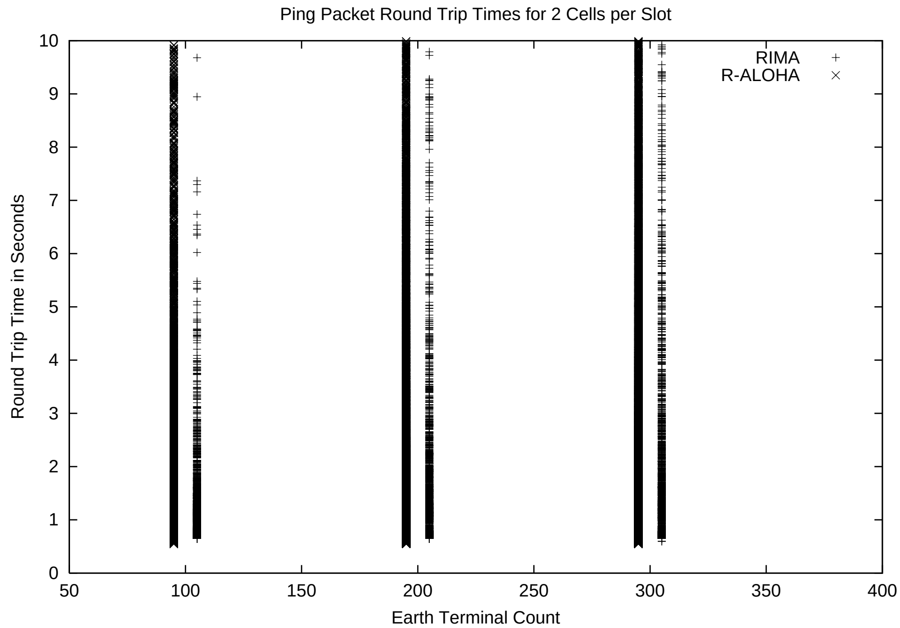

Figure 5.25: A Scatter Plot for Ping Round Trip Times, Granularity of 2 Cells Per Slot

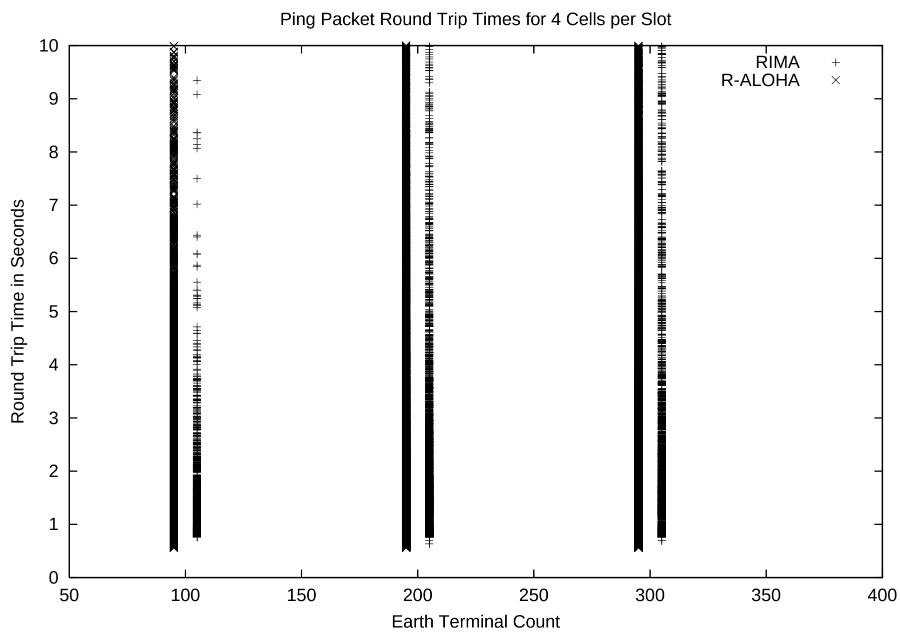

Figure 5.26: A Scatter Plot for Ping Round Trip Times, Granularity of 4 Cells Per Slot

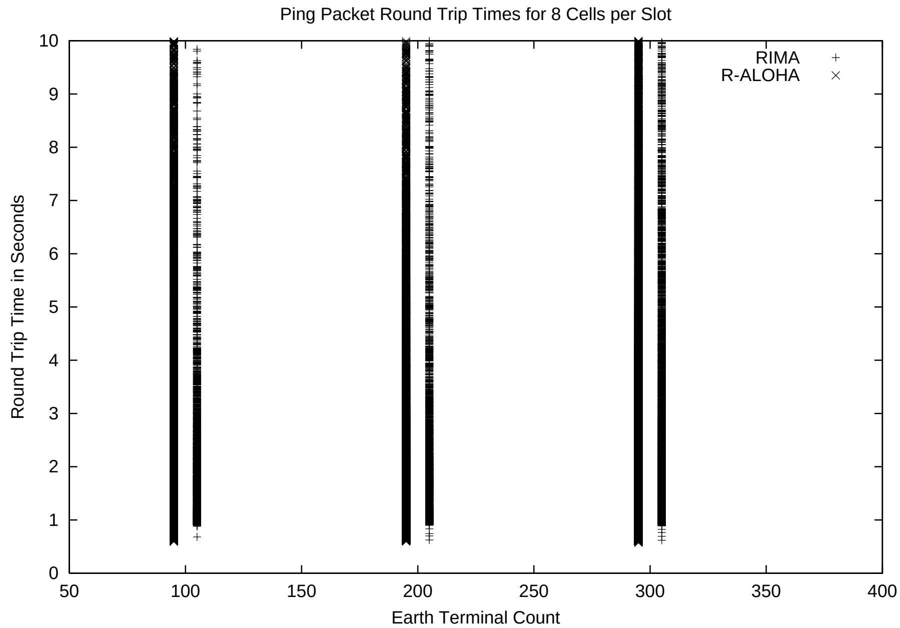

Figure 5.27: A Scatter Plot for Ping Round Trip Times, Granularity of 8 Cells Per Slot

between 25 to 50 percent of the measured round-trip time is due to propagation delay, which is irreducible. In practice these times are not desirable for interactive computing, thus perhaps a GEO satellite is not the preferred last mile technology for this application. In reality though, many regions of the globe have only satellite technology to rely on and until non-geostationary satellites can be reliably produced and deployed, a GEO satellite is the only solution.

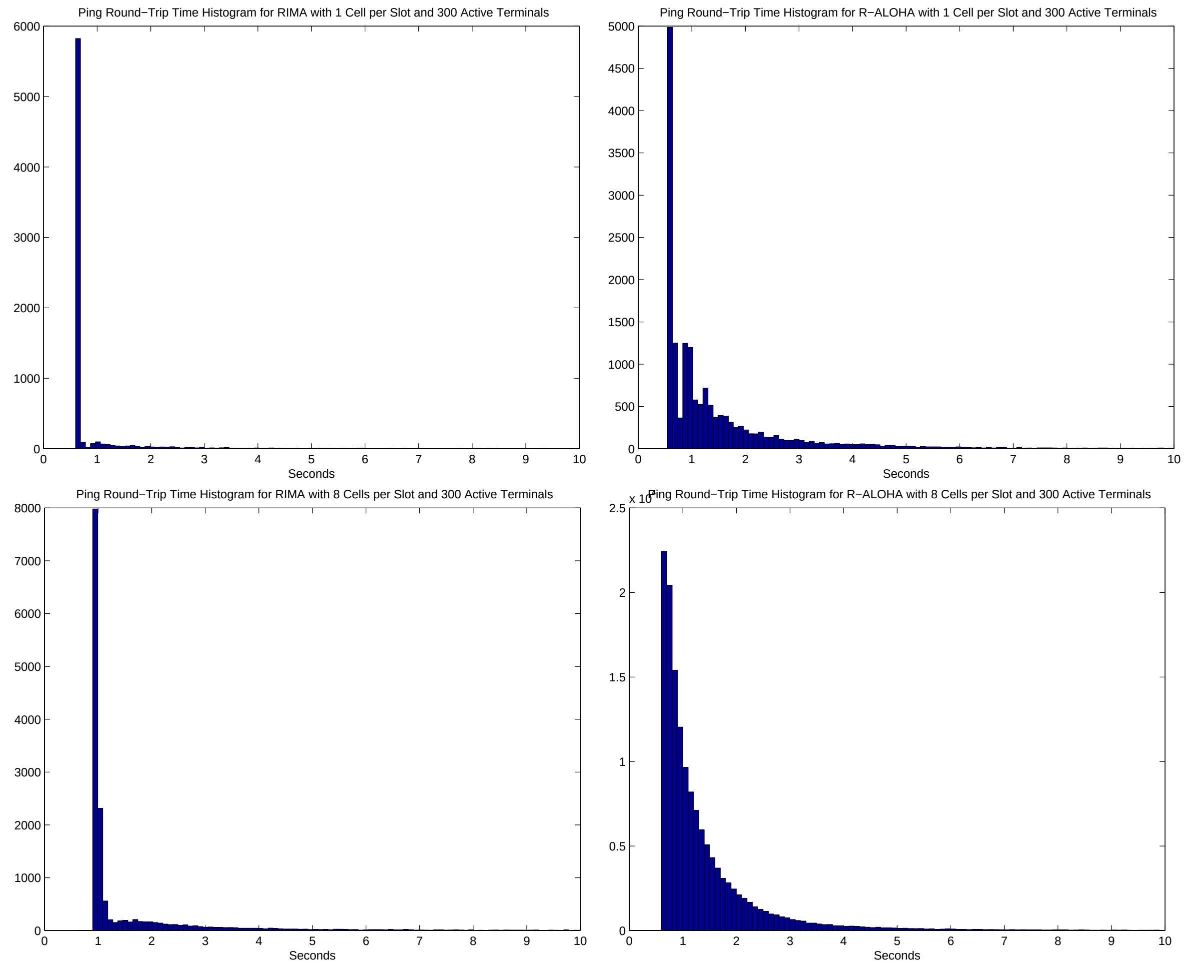

Figure 5.28: Histograms of Ping Round Trip Times

Table 5.4: Comparing Ping's Mean Round Trip Times for RIMA and R-ALOHA

Active Terminals	Slot Granularity	RIMA	R-ALOHA
100	1	0.7289	1.1467
200	1	0.9422	1.3772
300	1	1.2796	1.4915
100	2	0.8034	1.4207
200	2	1.0773	1.9160
300	2	1.6150	2.4357
100	4	1.0128	1.5411
200	4	1.4039	1.9385
300	4	1.8188	2.5923
100	8	1.3821	1.9767
200	8	1.6163	2.1633
300	8	2.0991	2.6108

5.6 Conclusions

In this chapter we presented Response Initiated Multiple Access, a new medium access control protocol designed to deliver computer data over satellite uplink channels. It is fundamentally different from previously designed MAC protocols in that it requires a connection-oriented protocol to be in operation at the Transport Layer. Collision free packet transmission is achieved by pre-allocating uplink bandwidth based upon a response from a remote server. It has been demonstrated via computer simulation to outperform Slotted and Reservation ALOHA for a targeted application, that being downloading World Wide Web pages via the Hyper Text Transfer Protocol. For transferring bulk amounts of data into the network it is roughly comparable to Reservation ALOHA, and when in comes to interactive computing, it outperforms Reservation ALOHA.

Chapter 6

Conclusions

Satellite networks have many appealing features with respect to content distribution to end users. The maturity of geostationary satellite technology, the broadcast nature of the downlink channel, and capability for its use anywhere around the globe make it a compelling technology for certain customers in the broadband market place. Most existing, reliable satellite systems are in place in geosynchronous earth orbit. Although these systems are the most understood, they suffer in performance and capacity due to the long delays across the ground-to-satellite channel.

In this work, we considered means of Quality of Service provisioning for Internet content over geosynchronous earth orbit (GEO) satellite channels by way of robust design of medium access control protocols. We restricted our focus to the Data Link Control Layer, abstracting Physical Layer realities and representing them as multiple access interference, also referred to as collisions. We defined a *personal earth terminal* scenario, which we believe captures an emerging model for broadband access to Internet media. In this model, end users have sole access to a satellite earth terminal and utilize it as their network entry point. We implemented this network, for the purposes of simulation, by creating extensions to the discrete event simulation tool, **ns**. These extensions were generated in a systematic, object oriented manner which allowed us to simulate a diverse set of environments and protocol techniques.

We first outlined various scenarios in which satellite technology is being used for the delivery of Internet packets. In each of these scenarios, we described the most appropriate medium access solution. We then expounded upon one scenario in particular in which satellite networks were being used as a "last mile" technology to bring Internet media directly to end users and labeled this scenario as a personal earth terminal model. Following this we explained our methodology for medium access control protocol design, which takes into account both the nature of the traffic arriving to the Data Link Control Layer and the constraints imposed by the Physical Layer. We then gave examples of how this design method could be applied. Upon doing this we presented in detail two medium access techniques, Random Access / Demand Assigned Multiple Access (RA/DAMA) and Response Initiated Multiple Access (RIMA). RA/DAMA is a MAC protocol intended to deliver real-time, medium quality video content from individual earth terminals into the internetwork, via the satellite uplink channel. We conducted extensive computer simulations of RA/DAMA using **ns** to show its effectiveness. In order to develop a comprehensive MAC solution for all Internet content, we then shifted our focus from real-time media to computer data applications. From this investigation, we designed RIMA, which seeks to be a complementary medium access technique for efficient transport of client-server computer data. As with RA/DAMA, we applied our MAC design methodology to systematically develop RIMA. We also conducted extensive computer simulations using our extensions to **ns** in order to prove RIMA's capability. We did these simulations on three types of client-server applications; World Wide Web transfers, bulk transfers into the network, and interactive computing.

Our final comments will be on the subject of integration and implementation. Both RA/DAMA and RIMA were designed to reside on a TDMA modem. Because of this it is possible to have both MAC protocols co-exist in a modem with a TDMA PHY Layer. This is shown in block diagram form in Figure 6.1. To accomplish this, additional *middleware* is needed on the modem. A packet filter will split the arriving packets into client-server and real-time streams. This is accomplished by reading the IP headers and sending the UDP traffic to a real-time API and the TCP traffic to a client-server API. Client-server traffic can be sent directly into a protocol sub-layer implementing RIMA. The real-time sub-layer will need additional middleware to derive from the incoming stream its traffic description. A protocol, standardized as IEC-61883, aimed at deriving traffic descriptors from audio / video transport streams already exists in the home multimedia environment [23]. Middleware very similar to this could be developed for a TDMA satellite PHY Layer. This would be used by RA/DAMA in generating an optimal DAMA metric. The packet flows from both the RIMA and RA/DAMA sub-layers could then be re-combined at the PHY Layer for transport over the satellite channel.

Approaches like this will be emerging in the very near future in many wireless systems, not just satellite. The primary reason for this is that wireless systems represent both the first and last hops of a communication network. Often times this characteristic brings with it the burden of QoS provisioning for video, audio, and computer data. One of the best ways of accomplishing this task is by applying this design methodology.

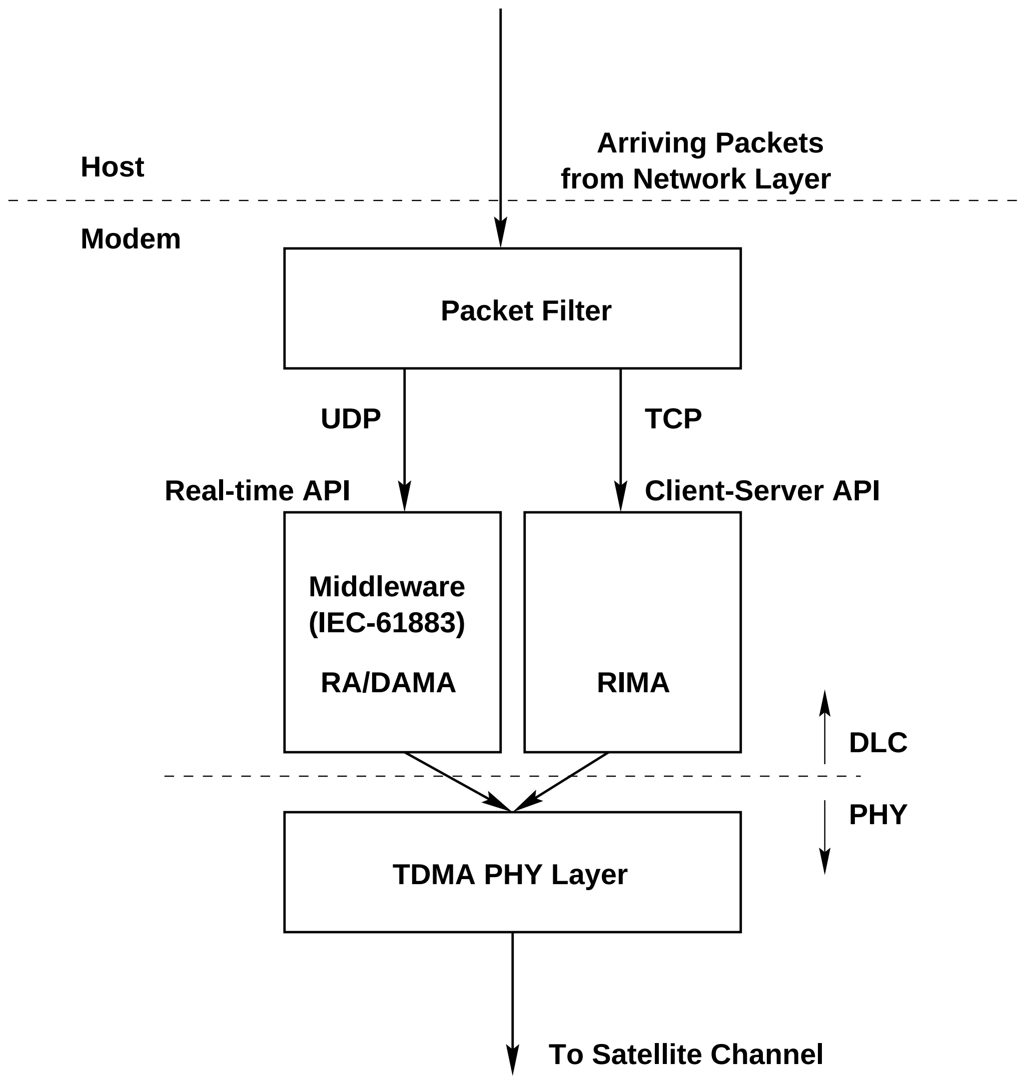

Figure 6.1: A Block Diagram showing RA/DAMA and RIMA's Integration on a Common TDMA PHY Layer

Bibliography

[1] N. Abramson. The ALOHA System - Another Alternative for Computer Communications. In *Proc. of the Fall Joint Computer Conference*, pages 281–285, 1970.

[2] M. Allman et al. Ongoing TCP Research Related to Satellites. Technical Report draft-ietf-tcpsat-res-issues-03.txt, Internet Engineering Task Force, May 1997. **URL**: http://www.ietf.org/internet-drafts/draft-ietf-tcpsat-res-issues-03.txt.

[3] M. Allman, Dan Glover, and L. Sanchez. Enhancing TCP Over Satellite Channels using Standard Mechanisms. Technical Report RFC 2488, Internet Engineering Task Force, January 1999. **URL**: ftp://ftp.isi.edu/in-notes/rfc2488.txt.

[4] A. Baiocchi et al. Modeling of a distributed access protocol for an ATM satellite system: An algorithmic approach. *IEEE JSAC*, 9:65–75, 1991.

[5] H. Balakrishnan, S. Seshan, E. Amir, and R. H. Katz. Improving tcp/ip performance over wireless network. In *Proc. 1st ACM Conf. on Mobile Computing and Networking*, November 1995.

[6] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger. Long-range dependence in variable-bit-rate video traffic. *IEEE Trans. Comm.*, 43:1566–1579, 1995.

[7] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, 1992.

[8] Cable Television Laboratories. *Data-Over-Cable Service Interface Specification (DOCSIS)*, Mar 1998.

[9] R. Caceres, P. B. Danzig, S. Jamin, and D. J. Mitzel. Characteristics of wide-area TCP/IP conversations. In *Proc. ACM SIGCOMM*, Zurich, Switzerland, 1991.

[10] K. Chang, I. Rubin, and G. Forcina. Performance analysis of integrated-services load-adaptive/tdma satellite networks. *Int'l J. Sat. Comm.*, 13:427–436, 1995.

[11] D. P. Connors and B. Ryu. Performance Evaluation of Satellite MAC protocols with QoS Guarantees. In *Proc. 2nd Workshop on Satellite Based Information Services, (WOSBIS'97)*, 1997.

[12] D. P. Connors, B. Ryu, and S. K. Dao. Modeling and Simulation of Broadband Satellite Networks, Part I: Medium Access Control for QOS Provisioning. *IEEE Communications Magazine*, Mar. 1999.

[13] W. Crowther et al. A System for Broadcast Communication: Reservation ALOHA. In *Proc. of the 6th Hawaii International Conference on Systems Sciences*, Honolulu, Hawaii, 1973.

[14] S. Dao, E. Shek, A. Vellaikal, R. Muntz, L. Zhang, M. Potkonjak, and O. Wolfson. Semantic multicast: Intelligently sharing collaborative sessions. *ACM Computing Surveys*, 1999. To appear.

[15] S. Deering and D. R. Cheriton. Multicast routing in datagram internetworks and extended lans. *ACM Transactions on Computer Systems*, 1990.

[16] D. E. Duffy, A. A. McIntosh, M. Rosenstein, and W. Willinger. Statistical analysis of CCSN/SS7 traffic data from working CCS subnetworks. *IEEE JSAC*, 12:544–551, 1994.

[17] A. Erramilli and W. Willinger. Fractal properties in packet traffic measurements. In *Proc. St. Petersburg Regional ITC Seminar*, 1993.

[18] R. Fielding et al. Hypertext Transfer Protocol – HTTP/1.1. Technical Report draft-ietf-http-v11-spec-rev-03, HTTP Working Group, 1997. **URL**: http://www.ietf.org/internet-drafts/draft-ietf-http-v11-spec-rev-03.txt.

[19] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616, June 1999.

[20] Traffic Management Working Group. *ATM Forum Traffic Management Specification Version 4.0*. ATM Forum Technical Committee, 4.0 edition, April 1996. manual.

[21] A. Hung et al. A framework for ATM via satellite. In *Proc. IEEE GLOBECOM*, London, UK, 1996.

[22] J. Hutcheson and M. Laurin. Network flexibility of the IRIDIUM global mobile satellite system. In *Proc. 4th IMSC*, pages 503–507, Ottawa, Cananda, June 1995.

[23] International Electrotechnical Commission. *IEC-61883 Consumer Audio/Video Equipment - Digital Interface*, Feb 1998. http://www.iec.ch.

[24] V. Jacobson. Congestion avoidance and control. In *Proc. ACM SIGCOMM*, Stanford, CA, 1988.

[25] P. R. Jelenkovic, A. A. Lazar, and N. Semret. The Effect of Multiple Time Scales and Subexponentiality in MPEG Video Streams on Queueing Behavior. *IEEE Journal on Selected Areas in Comm.*, 15:1052–71, August 1997.

[26] L. Kleinrock and S. S. Lam. Packet Switching in a Slotted Satellite Channel. In *1973 National Computer Conference*, pages 703–710, New York, NY, 1973.

[27] S. V. Krishnamurthy, D. P. Connors, S. Sandhu, J. Soma, and S. K. Dao. Medium access control and hand-off support for a cellular network with mobile base-stations. In *submitted to Mobile Multimedia Communications Conference (MoMuC)*, July 1999.

[28] K. S. Kwak and K. J. Lim. A Modified PDAMA Protocol for Mobile Satellite Communications Systems. *IEEE Journal on Selected Areas of Communications*, 13(2), feb 1995.

[29] T. Le-Ngoc and S. V. Krishnamurthy. Performance of Combined Free/Demand Assignment Multiple-Access Schemes in Satellite Communications. *Intl. Journal of Satellite Communications*, 14:11–21, 1996.

[30] Kdd to launce two-way satellie internet service. Distributed by WirelessNetNow, April 1999.

[31] M. Macedonia and D. Brutzman. Mbone provides audio and video across the Internet. *Computer*, 27(4):30–36, 1994.

[32] B. Mah. An Empirical Model of HTTP Network Traffic. In *Proc. IEEE INFOCOM*, 1997.

[33] G. Maral. *VSAT Networks*. John Wiley and Sons, 1995.

[34] K. Maxwell. Asymmetric digital subscriber line: Interim technology for the next forty years. *IEEE Communications Magazine*, October 1996.

[35] S. McCanne. VINT Network Simulator(*ns*), version 2.0, 1998. http://mash.cs.berkeley.edu/ns/ns.html.

[36] S. McCanne and V. Jacobson. *vic*: A flexible framework for packet video. In *Proc. ACM Multimedia*, San Francisco, CA, 1995.

[37] J. Moy. Ospf version 2. RFC 1247, July 1991. 189 pages.

[38] A. Mukherjee. On the Dynamics and Significance of Low Frequency Components of Internet Load. *Internetworking: Research and Experience*, 5(4):163–205, dec 1994.

[39] John K. Ousterhout. *Tcl and the Tk Toolkit*. Addison-Wesley, 1994.

[40] Craig Partridge, 1998. Private Correspondence.

[41] V. Paxson and S. Floyd. Wide Area Traffic: the failure of Poisson modeling. *IEEE Trans. Net.*, 3:226–244, 1995.

[42] J. Postel. User Datagram Protocol. RFC 768, Aug. 1980.

[43] J. Postel. Transmission Control Protocol. RFC 793, Sept. 1981.

[44] John G. Proakis. *Digital Communications*. McGraw Hill, 1995. 3rd edition.

[45] R. Rom and M. Sidi. *Multiple Access Protocols : Performance and Analysis*. Springer Verlag, July 1990.

[46] O. Rose. Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems. In *Proceedings. 20th Conference on Local Computer Networks*, pages 397–406, 1995.

[47] B. K. Ryu and A. Elwalid. The importance of Long-Range Dependence of VBR video traffic in ATM traffic engineering: Myths and realities. In *Proc. ACM SIGCOMM*, San Francisco, CA, 1996.

[48] Hady Sallum. Local multipoint distribution service. *IEEE Communications Magazine*, October 1998.

[49] W. R. Stevens. *UNIX Network Programming*. Prentice-Hall, 1990.

[50] W. R. Stevens. *TCP/IP Illustrated*, volume I: The Protocols. Addison Wesley, 1994.

[51] M. A. Sturza. Architecture of the TELEDESIC satellite system. In *Proc. 4th IMSC*, pages 212–218, Ottawa, Cananda, June 1995.

[52] Andrew S. Tanenbaum. *Computer Networks*. Prentice Hall, 1996. Third Edition.

[53] J. S. Turner. New Directions in Communications (or which way to the information age?). *IEEE Communications Magazine*, 24:8–15, 1986.

[54] W. Willinger. Traffic modeling for high-speed networks: theory versus practice. In *Stochastic Networks*, IMA Volumes in Mathematics and Its Applications. Springer-Verlag, New York, 1994.

[55] W. Willinger, M. Taqqu, R. Sherman, and D. Wilson. Self-similarity through high-variability: Statistical analysis of Eternet LAN traffic at the source level. In *Proc. ACM SIGCOMM*, Cambridge, MA, 1995.