

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE SECRET		3. REPORT TYPE AND DATES COVERED MAJOR REPORT
4. TITLE AND SUBTITLE TCP/IP OVER ATM VIA SATELLITE LINKS			5. FUNDING NUMBERS	
6. AUTHOR(S) 2D LT KUEHNE TIMOTHY P				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) UNIVERSITY OF COLORADO AT COLORADO SPRINGS			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) THE DEPARTMENT OF THE AIR FORCE AFIT/CIA, BLDG 125 2950 P STREET WPAFB OH 45433			10. SPONSORING/MONITORING AGENCY REPORT NUMBER FY99-21	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION AVAILABILITY STATEMENT Unlimited distribution In Accordance With AFI 35-205/AFIT Sup 1			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)				
14. SUBJECT TERMS				
15. NUMBER OF PAGES			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT		18. SECURITY CLASSIFICATION OF THIS PAGE		19. SECURITY CLASSIFICATION OF ABSTRACT
20. LIMITATION OF ABSTRACT				

TCP/IP over ATM via Satellite Links

19990120 017

Tim Kuehne

04 May 1998

Master of Engineering
Space Operations

All rights reserved. No part of this report may be reproduced, in any form
or by any means, without the permission of the author.

DTIC QUALITY INSPECTED 3

Abstract

Living in the Information Age means that the more information is generated, the more new information is required. A large contributor to the rapid increase of new information and its ease of accessibility is the Internet. The backbone of the Internet is the TCP Internet protocol. New emerging network technology, ATM, achieves data throughput never before possible. Layering TCP/IP over ATM will increase throughput performance that TCP/IP could not achieve alone. With the growing desire to use satellites as a medium for the Internet as well as the large amounts of digitized communication that already occurs daily via satellites, the TCP/IP proposes to become a standard for satellite communication. The high data rates of ATM, however, is very appealing for use on satellite links in an effort to make them more cost effective. Obstacles exist which prevent TCP/IP over ATM in its current state from becoming an efficient way to send information to and from satellites. This paper looks at the unique challenges of networking via satellite links, discusses the impact of the satellite links on TCP/IP over ATM and the limitations of TCP/IP which are caused, and investigates solutions to these limitations.

Table of Contents

Background	4
Introduction.....	6
ATM	6
TCP	9
IP.....	10
TCP/IP	11
TCP/IP via ATM.....	12
Specific Topics of Review	14
Discussion of Limitation.....	15
BER	15
Multiple Access Techniques	16
Limitations of TCP	19
Public TCP/IP over ATM Satellite Links	22
Private TCP/IP over ATM Satellite Links	23
Discussion of Applications	25
Conclusions/Recommendations.....	26
Bibliography	28
Appendix A: Error Probability Calculations.....	30

Background

“I know of but one single means of increasing the prosperity of a people that is infallible in practice that I believe one can count on in all countries as in all spots. This means is naught else but to increase the ease of communication between men...”

Alexis de Tocqueville, 1835 [1]

In a very few short years, the Internet has changed the way in which much of the world conducts business and its day to day affairs. The Internet now brings the distant close to the average person. The near real-time speed is especially powerful, allowing an Internet user in Florida to view a computer-generated camera picture from France with minimal delay. Carrying much of the burden of the information superhighway is TCP/IP (Transmission Control Protocol/Internet Protocol). TCP/IP is the predominant format which computers use to communicate and navigate via the Internet. This protocol is the means by which information is divided up into packets and sent with the addresses of the sender and recipient. TCP/IP allows for a dynamic approach to networking wherein the sender does not know the directions to the receiver. Dynamically routed messages allow messages to avoid congestion or outages that may exist at any given time. TCP/IP facilitated much of the Internet's success in making data exchange possible, whether it be world-wide or local and also provides some communication with satellites in orbit.

Asynchronous Transfer Mode (ATM) is a new, emerging networking technology for broadband-ISDN (Integrated Services Design Network) systems. Its high speed and adaptability to various types of data make it very appealing to use. However, for it to gain acceptance for use by the Internet, not simply corporate networks, it must be able to transport TCP/IP traffic transparently. The high speed of ATM networks combined with the versatility that satellite links bring seems to be a perfect coupling. There are some unique characteristics, however, which make communication between a computer on Earth and a computer in space difficult to do via

TCP/IP over ATM. Functions that make TCP/IP and ATM highly effective here on Earth are based on assumptions that are not true with regards to communication via space-based satellites. These factors include distance, delay, and reliability of the connection, as well as increases in throughput requirements and throughput capabilities.

Introduction

ATM

Asynchronous Transfer Mode (ATM) is the network protocol for the broadband integrated services digital network (B-ISDN). ATM allows B-ISDN to sustain much higher data rates than the original ISDN. B-ISDN was developed to exploit the enormous bandwidth that optical fibers offer [1]. A list of services that a broadband service is expected to support is provided by Akimaru, Finley and Niu [1], ordered from highest consumption of bandwidth to lowest:

- Interactive video (real-time)
- Recorded video
- Still images
- Facsimile
- Interactive voice
- Recorded voice
- Interactive data
- Non-interactive data

ATM supports the concurrent transmission of voice, video, and data. ATM's superiority in handling of the concurrent transfer of heterogeneous data is due to its ability to support bursty traffic on the network [2 OpnetATM3].

ATM is a high-speed, connection-oriented computer networking technology. ATM's connection oriented nature is in contrast with IP, which provides for connectionless data transfers. Each computer in an ATM network is connected to an ATM switch via optical fibers and an ATM network card that is installed in the computer. Typical sizes of small ATM

switches connect 16 or 32 computers [3]. Switches connect with other switches to form larger networks. In order for two computers to transfer data with each other using ATM, they must establish a virtual circuit in the network.

Communication between two computers in an ATM network is analogous to placing a telephone call [3]. The client starts the transaction by specifying the address of the host. The client sends this request to the switch that it is connected to. The switch first checks to see if that address is on its switch. If so, the host is contacted with the request. If not, the switch passes the request to switches that it is connected to. This process is repeated until the host is found or until it has been determined that the address cannot be reached. The host decides whether to accept the connection or not. It sends an acknowledgment back accepting or declining the request. The client waits until it receives a reply with either the connection path or a message that the connection could not be made. These connections are referred to as virtual channels. The identification of a virtual channel is comprised of two parts, the virtual path identifier (VPI) and the virtual channel identifier (VCI) [14]. The VCI comprises two bytes of the five-byte ATM header and the VPI comprises one byte of the header.

Miller compares a virtual path to a large telephone cable in which all circuits end at a central office [14]. A virtual path is a group of virtual channels which have the same end point. The endpoint is not necessarily a computer, but could be a switch that the channels have in common. The VPI helps to make the switching process more efficient. The VCI is used to differentiate the data in the same virtual path. Thus a VPI may be compared to an area code of a telephone number and the VCI may be compared to the rest of the telephone number.

The ATM switch to which the client is connected sends the identifier for the virtual connection that is to be used for the connection to the client. The same occurs between the host

and the ATM switch to which it is connected. The VCI and VPI are not unique numbers for the entire path between the client and host, rather they are only unique for the connection between a computer and a switch, or between a switch and another switch. Once the connection has been established, data sent between the two computers will not contain client and host addresses. The client will send data to its switch using the VCI/VPI pair given to it. The switch then knows how to map that VCI/VPI pair with another VCI/VPI pair which may connect to either the host or another switch. This process repeats until the data arrives at the host.

During this establishment of a connection, not only must the two computers agree on the quality-of-service (QoS) for the connection, but every switch in the connection must also agree on the requested QoS. This minimum QoS is then guaranteed for the duration of the connection. After the client and host have complete their transaction, the ATM switch is notified and the connection is terminated. This is equivalent to hanging up the telephone at the end of a conversation. The connection identifier may then be reused for a future connection, but does not necessarily need to include the same computers that were in that connection.

ATM transports data in fixed-size packets called cells. Each cell is 53 bytes, in which 48 bytes are for use of transporting data and 5 bytes are for the cell header. ATM takes advantage of the high reliability of modern networks by reducing the overhead (cell header size) required to transport data. ATM provides very little error correction tools, opting instead to allow higher layers in the stack to provide error correction (e.g. TCP). The standard size of ATM cells allows switches in the network to operate significantly faster, thus enabling ATM to achieve much higher data rates than if cell sizes were of variable length.

TCP

The Transmission Control Protocol (TCP) is a connection-oriented transport layer protocol. TCP provides error detection and correction which permit reliable delivery of data on an unreliable network (2 OpnetTCP 379). TCP uses positive acknowledgements with retransmission to provide reliability. In the simplest form of positive acknowledgements, the sender transmits a packet and waits to find out if it arrived successfully. The sender does not send any more packets until confirmation from the receiver that the sent packet arrived. Once packet arrives at the receiver, the receiver sends the sender an acknowledgement that the packet arrived. If an acknowledgement is not received in a specified period of time, the sender assumes that the packet was lost and resends the packet. After receiving an acknowledgement, the sender sends the next packet and the process continues. Simple positive acknowledgements, however, is inefficient in bandwidth utilization as there is a waste of time while the sender waits for an acknowledgement when it could be transmitting further packets.

TCP overcomes the inefficiencies of simple positive acknowledgements by using sliding windows. Sliding windows uses the concept of positive acknowledgements, but instead of transmitting one packet at a time, multiple packets are allowed to be transmitted. For example, if the window size is 16, then the first 16 packets of a transmission would be allowed to be transmitted before an acknowledgement is received. As acknowledgements are received by the sender, the window "slides" and allows the same number of packets to be transmitted as acknowledgements were received. An optimum window size is dependent on the connection, but it keeps the network busy without exceeding the buffer capabilities of the receiver. To help the sender determine how big its window size should be, each acknowledgement from the

receiver has a window advertisement field which specifies how much room is left in its buffer [3]. As room in the buffer increases, the window size increases and more packets are allowed to be transmitted.

TCP packages data in variable size segments. Finding the optimum segment size is difficult. Coupled with IP, the total size of the header is 40 bytes [3]. If the segment is too small, bandwidth is not used efficiently because of the large percentage of each segment that is used by the headers. When the segment is too large, fragmentation occurs. The problems of fragmentation are discussed in the section discussing IP. The optimum segment size is the largest segment size allowed without causing fragmentation.

IP

The Internet Protocol (IP) is a connectionless network layer protocol. IP operates at the same layer as ATM does, but in contrast to ATM, IP is connectionless and may interconnect heterogeneous networks. IP packages data for transmission in packets called datagrams. These datagrams are analogous to ATM cells. However, IP datagrams have variable length. Since IP connects heterogeneous networks, each network may have a differing maximum transfer unit (MTU), which is the largest size that a datagram may be. In the event that a datagram encounters a network that only supports a smaller MTU than what is needed for the datagram, the datagram is fragmented into smaller chunks that fit into the MTU window. Once fragmentation occurs, the packets are not defragmented until they arrive at their final destination.

Data transfer using the IP protocol incurs a time penalty when fragmentation occurs. Time is required to fragment each datagram into smaller sized packets and time is required to reassemble these fragments. Also, since fragments are not reassembled until the final

destination, if an error is found in a fragment enroute, the error may not be corrected until the reassembled datagram is inspected. This long wait is required because fragments are unaware which original datagram they belong to and are thus unable to specify which block of data should be resent. The host computer must resend the entire datagram which once again must be fragmented and defragmented enroute to the final destination. This retransmission also wastes bandwidth. It is more than likely the case that the other fragments of the original erred datagram arrived successfully and thus do not need to be resent.

TCP/IP

The way in which TCP/IP sends and receives data is based on the OSI seven layer network model. The OSI model consists of the following layers, from lowest (hardware) to highest (software): physical, data link, network, transport, session, presentation, and application. TCP/IP, however, uses a five layer model by combining the session, presentation, and application layers into one layer, which is called the application layer and calls its transport layer the internet layer.

The application layer is the first step in the process of data transmission. Examples of the application layer include e-mail programs and web browsers. Using e-mail as an example, the user writes a message and clicks the send button. The contents of the message are sent down to the transport layer. TCP works at the transport layer. TCP divides the data into frames and attaches headers to each frame which provide information necessary for error correction and ordering instructions. The transport layer then passes the data to the internet layer. This is where IP works. IP is responsible for addressing and routing the data. IP takes TCP's frames and packages them with a header, forming IP datagrams. Datagrams are forwarded to the data

link layer and physical layer. The physical layer and the data link layer are often combined and thought of as one layer as it is difficult to separate the two layers. The data link layer puts the data on the medium being used, e.g. copper wire, optical fiber, satellite links. The physical layer consists of computers' network interface cards and the connection medium that is used. Upon arriving at the destination, the process is reversed. Packets are combined into IP datagrams. IP then removes its header and forwards the data up to TCP. TCP checks the data for accuracy. If the data did not arrive successfully, TCP sends a request back to the originating computer to retransmit that frame. Once the data has successfully arrived, TCP forwards the data to the application layer. In this example, the user is the recipient of an e-mail and can now read the message.

TCP/IP via ATM

TCP/IP is a very popular protocol and is the standard for today's Internet and other applications. ATM networks and services provided by B-ISDN are rapidly becoming popular in today's networks and are predicted to become the Internet standard for tomorrow. In order to ease the transition from TCP/IP to ATM networks, it is necessary to transport existing network traffic using the new format. Although ATM native mode will eventually become the networking standard, eliminating the need for TCP/IP, it is necessary to look at how TCP/IP datagrams may be transported as ATM cells.

In order to send TCP/IP datagrams via ATM, the ATM adaption layer 5 (AAL 5) must be used [3,14]. It is the AAL that packs outgoing data into ATM cells at the host and unpacks cells and reassembles the data at the destination. Since the maximum size of a TCP/IP datagram is 64

KByte, and an ATM cell has a 48 byte payload, datagrams would have to be significantly fragmented without the features of AAL 5. AAL 5 allows an incoming datagram to be of any size, between 1 byte and 64 KB [3]. Using AAL 5, sending datagrams via ATM networks is then a straightforward process.

The data to be sent is formatted into IP datagrams and then the datagrams are sent to the ATM adaption layer, where they are divided to fit into an ATM cell's 48 byte payload. AAL5, in a process known as datagram encapsulation, generates a trailer, which contains information about the datagram, divides each datagram into cells, and then sends the cells followed by the trailer [3]. The receiving computer passes the cells up to AAL5 which then strips off the header of each cell and reassembles the datagram. AAL5 knows that it has received the entire datagram when it receives the trailer. The trailer is used to verify that nothing was lost or corrupted during transmission. The IP datagrams are then given to the TCP layer for inspection and error control. Once all of the data has successfully arrived, the data is then forwarded up to the application layer as a complete file. Although it may seem impractical to send IP datagrams via an ATM network because of the additional packaging and overhead, there is a distinct advantage in speed. ATM networks have a higher throughput than IP networks because of the high-speed ATM switches. Another way in which it is faster to send datagrams using ATM networks is more subtle. By using ATM, there is no need to fragment IP datagrams. The cause for fragmentation, as discussed previously in the IP section, no longer exists when datagrams are packaged as ATM cells.

Specific Topics of Review

As the Internet continues to become bigger, faster, and better, new technologies must be developed to support this progress. The one technology that will allow the Internet to explode into a tool that is truly global in scope will be networking computers and/or computer networks via satellite. This paper will focus on the transmission of data using the TCP/IP protocol over an ATM network via satellite links. Figure 1 shows conceptually how this concept will be performed in practice [4].

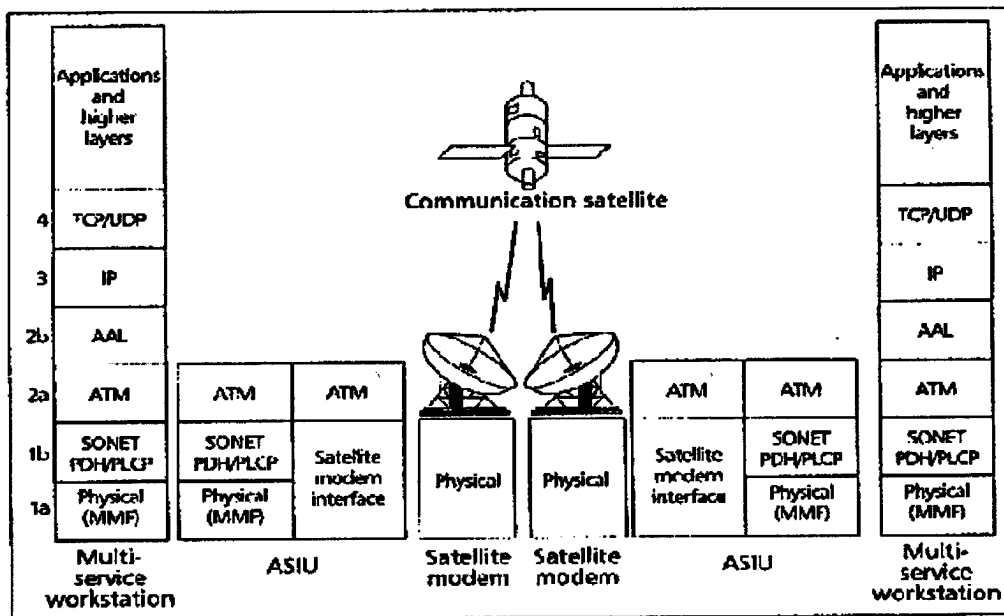


Figure 1. Graphical depiction of how TCP/IP over ATM via satellite links is implemented [4]

Utilizing satellite links, then Internet will have more bandwidth and will be able to reach parts of the Earth that the current wired infrastructure cannot reach. There are obstacles to overcome, however, in order to integrate satellite links into a seamless connection with terrestrial networks. All of the protocols that are being considered, namely TCP, IP, and ATM, were designed for terrestrial links. Properties of the terrestrial networks were assumed in designing those protocols are not inherent in satellite links. Chief among the assumptions of the networking protocols are low delay time and high quality of service over a reliable medium.

Discussion of Limitation

ATM was designed to operate in an environment that provided a high quality of service (QoS). QoS parameters include cell loss ratio, maximum and mean cell transfer delay, and cell delay variation [4]. Using these parameters, end-to-end performance of an ATM network may be quantified [4]. ATM transmissions are very sensitive to cell loss and thus was designed specifically with optical fibers in mind. Fiber can provide the high QoS that ATM needs to be fast and efficient by providing bit error rates (BER) of 10^{-10} and higher. The impact of an errored bit is much greater for an ATM cell than it is for a TCP/IP packet. This is due to ATM's assumption of a low BER, whereas TCP/IP was designed for unreliable networks. Error correction in an ATM header is small and can only handle single bit errors, but not error bursts. Error burst is a problem associated with satellite transmissions due to environment variations in the link between Earth and the satellite.

BER

To determine how often an ATM cell has a single errored bit, take the number of bits in a cell and find the probability that one of those bits has been errored. An ATM cell has 53 bytes, thus there are 424 bits per cell. Assuming the bit error rate is 10^{-10} , the probability that one bit in an ATM cell is 4.24×10^{-8} . ATM's header can correct a single bit error per cell. Assuming a 155 Mbps ATM network operating continuously, the probability of an errored cell translates into one bit for every 10 gigabits transferred or one cell every 64 seconds. ATM corrects errors if it is only a single bit that is in error. Therefore it is critical to know how many of these errored cells are cells with only single bit errors and how many are cells with more than one bit errors. Without an additional protocol to provide error checking like TCP, these bits will be uncorrectable, thus corrupting the cell which corrupts the entire transmission. The frequency which a cell has more than one bit in error is 4.746×10^{-15} which translates into once every 18.26

years or one bit in every 89327.1 terabits! With such a high probability of successful transmission, it is apparent why ATM, which was designed for the optical fiber medium with a BER of 10^{-10} [5], only has the capability to correct single bit errors.

However, the space link is not nearly as reliable as optical fibers. Current space modem technology allows for a BER of 10^{-5} [4,6]. The probability that a cell is in error via the space link is 0.0044231, or nearly 100,000 time greater than that of the optical fiber transmission. The probability that a cell has more than one error is 8.94×10^{-6} , almost two trillion times greater than the optical fiber transmission! This translates into one cell for every 111,827 cells transmitted having more than a single bit in error or an occurrence every 0.3059 seconds. For this reason, it is imperative that a protocol such as TCP be used in order to permit ATM transmission via satellite links.

Multiple Access Techniques

The choice of which satellite link access scheme to use for multiple connections also places restrictions on how well the link layer is utilized. There are inefficiencies in the various schemes used in the link layer in regards to the transmission of ATM data. Three of the most popular schemes are frequency-division multiple access (FDMA), time-division multiple access (TDMA), demand-assignment multiple access (DAMA), and code-division multiple access (CDMA).

FDMA divides the entire bandwidth of the satellite link into subchannels. Each subchannel is the same size and has its own frequency band. Since each subchannel has its own frequency band, there is no interference between subchannels. Depending on the setup, each earth station could be allotted its own subchannel, as in Figure 2. [4]

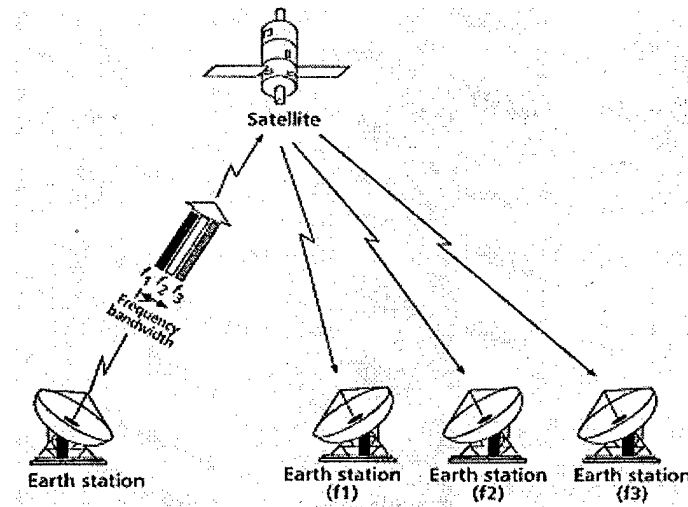


Figure 2 A picture of an FDMA system [4]

Since each subchannel is a portion of the entire bandwidth, a major advantage of FDMA is that smaller antennas are required. However, there is a lot of wasted bandwidth when implementing ATM with FDMA as ATM is dynamic by nature. ATM supports bursty traffic and may require more bandwidth than what has been allotted to a subchannel. Additionally, FDMA requires guard bands, which are allocations of the bandwidth that separate each subchannel. FDMA is too inflexible and wastes too much bandwidth to be a viable satellite access method for ATM transmissions.

In contrast to FDMA which divides the entire bandwidth by frequency, TDMA divides the entire bandwidth of the satellite link by time. In the normal implementation of TDMA, each channel is divided into time slots of equal size. Thus, each channel in a TDMA system gets the full bandwidth, but for a fraction of time. A picture detailing how TDMA works may be seen in Figure 3.

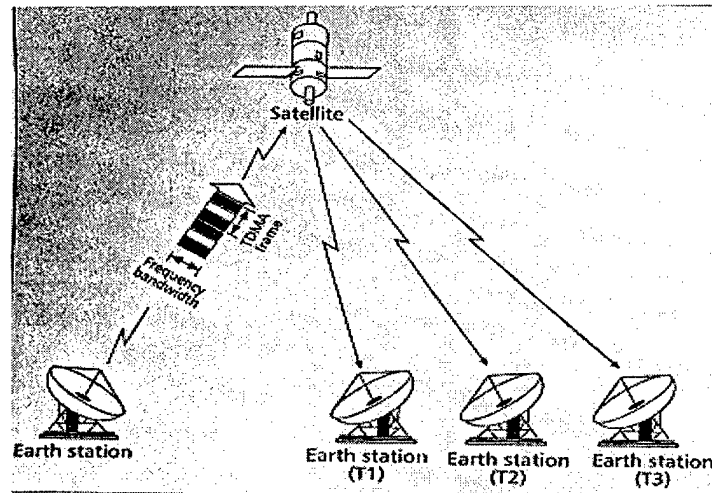


Figure 3 A picture of an TDMA system [4]

Since each antenna must handle the entire bandwidth, antenna sizes are larger than those required in an FDMA system. A significant advantage of TDMA, however, is that since each earth station is given access to the entire bandwidth, bursty ATM network traffic is supported very well. The problem is that since ATM traffic is bursty, it is often the case that a channel is not fully utilized, thus wasting bandwidth.

A solution to the problem of wasted bandwidth is provided by DAMA. As its name implies, it assigns the channels based on the demand. DAMA provides the means to dynamically allocate and reallocate satellite power and bandwidth based on the needs of the system [4]. For a satellite based network with multiple earth stations with varying network demands, bursty ATM traffic for example, DAMA is a very cost effective alternative. There is complexity involved in implementing DAMA to dynamically allocate the resources, but this complexity will pay for itself in improved overall network performance and in savings of space-link costs by reducing wasted bandwidth.

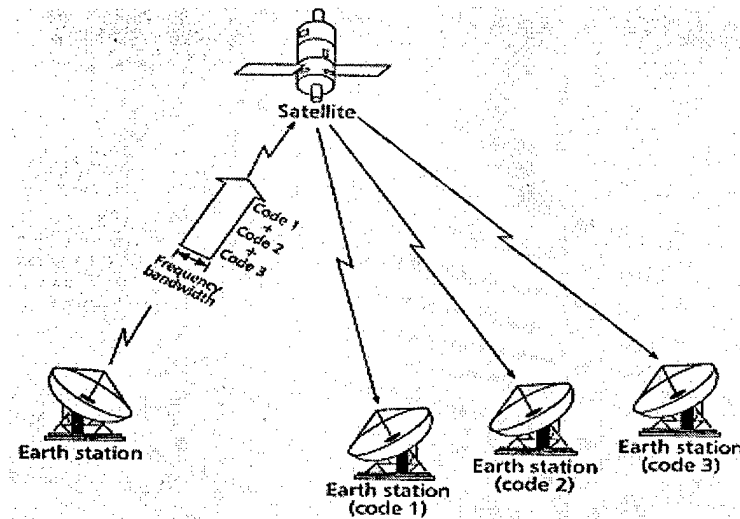


Figure 4 A picture of an CDMA system [4]

Finally, CDMA is a multiple access technique that spreads transmissions over the time-frequency domain [4] as shown in Figure 4. Instead of giving a user a specified frequency or time slot to use, CDMA is implemented by giving each user a pseudo-random code [15]. CDMA is a spread spectrum technique that takes a signal to be transmitted and spreads it over a band that is much larger than is required to transmit the signal [15]. The composite signal is retrieved by correlating the pseudo-random code with its pseudo-random spreading sequence. CDMA is very resistant to interference and for this reason it is popular for military systems as an anti-jamming technique.

Limitations of TCP

Taking TCP into consideration, a TCP connection has a maximum throughput defined by the following formula [1]: $\text{Max throughput} = (\text{receive buffer size}) / (\text{round trip time})$. Throughput is thus dependent on the buffer size of the receiving computer and on the delay time between the two computers. Delay time for satellite transmissions is dependent on what type of orbit the satellite is in. There are three major categories of satellite orbits, low earth orbit (LEO),

medium earth orbit (MEO), and geosynchronous orbit (GEO). Typical one way propagation delay times between Earth and a LEO satellite is between 20-25 milliseconds (Survey 40). This propagation delay time increases to 110-130 milliseconds for a MEO satellite and for GEO satellites the time is 250-290 milliseconds [4,12].

Using the previous throughput equation for a TCP connection via a geosynchronous satellite, maximum throughput equals 64 Kbytes/585ms, equivalent to 112,000 bytes per second or 896,000 bits per second [2]. The upper bound on the maximum throughput for a TCP connection is not dependent on the bandwidth of the medium that is being used, rather it is dependent on TCP's 64 Kbyte window size that it uses for transmission. A full T1 channel, frequently used for space communication, has the capability to transmit 1,536,000 bits per second. The TCP Internet protocol cannot fully utilize a T1 channel connection between a ground station and a satellite. This inefficiency is a major reason why it has been difficult to implement TCP for use in space communication. It is plain to see, however, that increasing the window size of TCP would allow the satellite connection to use a T1 channel to its full capacity.

Another feature of TCP which makes it great for Earth communication, but not for satellite links is the feature known as positive acknowledgments. TCP establishes a reliable connection between two computers by acknowledging that each packet was successfully received. The long delay which is present in satellite communication (585 ms) is not conducive to the numerous acknowledgments which provide information about segment loss. Instead, a feature called selective acknowledgments, SACK, should be used for space operations using TCP. SACKs are to be generated by the receiving computer that would then inform the sending machine about which packets arrived and which packets the sender needs to retransmit [11]. Thus information concerning the reliability of the transfer takes less time and produces less congestion.

Further variants of the often used TCP protocol deal with the features known as slow start, fast transmit, and fast recovery. These features help TCP to control traffic flow in normal use, but provide many difficulties in space links. TCP/IP standards conferences made these

mechanisms mandatory in 1989 [14]. The purpose of slow start is to “gradually increase the rate at which the sender injects data into the network” [12]. The slow start algorithm starts by sending a packet and waiting for an acknowledgment for its receipt. Upon this acknowledgment, the sender sends two packets. This exponential increase of network traffic continues until the receiver’s window size has been reached or when a packet is lost. Slow start is not an effective technique for satellite links because of the wasted time and bandwidth required before full load is achieved [12]. With the great delay of the satellite transmission, TCP requires 3.5 seconds to reach its maximum throughput using slow start. Congestion avoidance is a feature that sends out an extra packet with each round trip to find out how much bandwidth is available on the network [12]. This protocol is based on the premise that a network is too congested when a packet is lost. When a packet is lost, the connection is put into slow start and allowed to build up until network traffic reaches one half of what it previously was.

One technique to increase the efficiency of TCP addresses the problems of TCP’s current methods of congestion avoidance using slow start. Fast recovery speeds up the process of lost packet detection. Fast recovery replaces TCP’s normal recovery process from a packet loss. Instead of throttling network traffic back to slow start, fast recovery reduces the packet sending rate in half and then begins congestion avoidance. By doing this, slow start is avoided and time and bandwidth are not wasted. Since it takes slow start 3.5 seconds to build up to full transmission speed, this period where bandwidth is not fully utilized is eliminated by fast recovery.

Fast recovery and selective acknowledgements have helped to improve TCP’s adaptability for satellite links. A final feature that has boosted the performance of TCP is a larger transmission window. As previously stated, the size of TCP’s transmission window is 64 Kbyte, allowing a maximum throughput of approximately 60% of a T1’s capacity. In order to more fully utilize the capacity of the T1 channel used in satellite links, TCP’s transmission window needs to be enlarged. To change the size of TCP’s window would require formal changes and would encounter many compatibility problems with the now-standard protocols.

An application level solution, however, simulates a virtual window that is larger. The application would be a modified version of FTP, which uses multiple TCP connections [2]. By using multiple TCP connections, the throughput of a single TCP connection can be multiplied. Multiple connections of TCP achieve an efficiency of approximately 90% of the optimal throughput, a significant increase over the single TCP connection previously used.

Public TCP/IP over ATM Satellite Links

Implementing TCP/IP over ATM satellite links for use by the general public that are surfing the web requires that the TCP/IP packets must retain their integrity. This means that modifying the protocol for the link is not an option. Formal changes are required to increase TCP's window size. It is clear to see that throughput of TCP/IP over ATM via satellite links is severely limited by TCP. Without using multiple channels as previously discussed, TCP does not even reach 1 Mbyte/sec. This is totally inefficient for an ATM network that can achieve 155 Mbits/sec and higher. The bottleneck is TCP's requirement that every packet be acknowledged and that no packets may be sent beyond the number in its address space.

A technique must be developed to get around TCP's limited window size, without disturbing the integrity of the transmitted data. I propose placing a computer between the satellite modem and transmit antenna. This computer would acknowledge the receipt of all of the data being sent, tricking the sending computer. The sending computer will thus continue to send data. The new intercessary computer would acknowledge the receipt of the packets and send the data onto the satellite. The problem now is that this data that is in the satellite link has nothing to care for it. The sending computer is satisfied because it received acknowledgements for the transmitted data and the receiving computer has no idea of the data's existence. Should

the link fail, this data will be lost with no means of recovery. Since the intercessary computer acknowledges the receipt of the data, it is accepting responsibility of the data. Therefore, the intercessary computer must assume responsibility for the data. The intercessary computer must store all of the data that has been transmitted until real acknowledgements return. This computer must screen out the acknowledgements, otherwise the sending computer will be confused by receiving two acknowledgements of sent data. The intercessary computer must also add a five bit header that will differentiate the packet streams. Since $155\text{Mbits/sec} = 20\text{Mbytes/sec}$ and TCP's maximum throughput is 1 Mbyte/sec, there will be 20 of sets of transmitted data that have the same packet numbering. This 5 bit header is a minimum which allows up to 32 different groups of data. Should a 2.4 Gbit/sec network be implemented over satellites, this header would have to be 9 bits. Thus we can see that this is a temporary fix until formal changes to the TCP/IP protocol are made and adopted by the computer community.

Private TCP/IP over ATM Satellite Links

A private TCP/IP over ATM satellite link implies that the owner of the links may use a standard separate from public standards. Thus, the TCP protocol may be modified in this scenario without the need for formal changes to be approved. The obvious modification to make is increasing TCP's window size from its current 16 bit field. As discussed, the 16 bit field for window size yields a maximum throughput equal to $64\text{ Kbytes}/585\text{ms}$, equivalent to 112,000 bytes per second or 896,000 bits per second. Being unable to overcome the laws of physics, the delay time may not be decreased for a geosynchronous scenario. In order to achieve 155 Mbps speed, TCP's window size must be increased to 11.7 Mbytes, which requires a 24 bit field. Using this increased window size makes it critical, however, that a quality congestion avoidance

technique be implemented.

The exponential growth in unacknowledged packets caused by increasing the window size means that there is an increased chance of packet loss. Fast recovery, as previously discussed, is a much better congestion avoidance technique than the original slow start. Two factors, however, may render fast recovery ineffective for this scenario. In fast recovery, each packet that is errored causes the transmission rate to be cut in half. The increased probability of a packet in that window becoming errored in transmission means that fast recovery will reduce the transmission more often. In addition, to the increased number of packets, it must also be remembered that the nature of errors in a satellite link is not random as in physical links, but rather they are bursty. So with the increase in number of packets out on the link, it may be assumed that when there is an errored packet, there are multiple errored packets and fast recovery will slow down the transmission rate in the same manner that slow start did, which is very inefficient. An investigation should be conducted to see the impact of the increased errors caused by the increased window size. Further, a modification to fast recovery should be considered which looks at the percentage of packets that are lost rather than the number of packets that are lost. It is not efficient to treat a single packet with the same importance in a window size of 64 Kbytes as a single packet in a window size of 11.7 Mbytes. It should be investigated to find if there is an optimum percentage of packets that are lost in which fast recovery techniques should be started.

Discussion of Applications

The applications of TCP/IP over ATM via satellite links are numerous and beneficial. By using satellite links, computers may be networked in ways that were not previously possible. Remote areas of the world will only be able to have access to the Internet through a satellite network backbone. ATM provides a much faster delivery standard of network data than ever before possible. Using ATM over satellite links will provide both the capability to transmit and the capability to distribute television world-wide via the Internet. Layering TCP/IP over ATM will allow current Internet applications to be downloaded at speeds requiring a fraction of the time currently needed. Large imagery files may also be transmitted at high speeds to and from locations all over the world with a high reliability that the received image is the exact copy of the transmitted image. This application is of particular interest to the military. A reconnaissance aircraft or a reconnaissance ground team will be able to take pictures of various points of interest and be able to instantaneously transmit those images to incoming aircrews or anywhere else in the world. Intelligence work in the military will be allowed to become more real-time than is currently possible.

Conclusions/Recommendations

TCP/IP over ATM via satellite links is a integral part of tomorrow's computer networks. ATM is a relatively new computer networking technology that is revolutionizing the way computers are networked and is expanding the possibilities of networking applications. Another new technology, satellite network links, is also revolutionizing how computers are networked. It is imperative that ATM is able to operate over satellite links in order to be able to seamlessly integrate satellite links into the Internet backbone. TCP/IP over ATM is critical for two reasons. First, TCP/IP is the current standard of the Internet. In order for ATM to gain acceptance in the Internet environment, it must be able to transparently transport TCP/IP. It is the only way that a large ATM infrastructure may be developed. Second, ATM was designed to operate in a low BER environment such as optical fiber. ATM does not perform well over satellite links because it only has the capability to correct single bit errors. TCP provides the error detection and correction capability that ATM requires in order to operate effectively over satellite links. However, TCP/IP has a slow throughput over satellite links due to its transmission window size.

Two solutions have been discussed to overcome TCP's limited throughput over satellite links. The first is to increase the size of TCP's transmission window. In order to support an ATM network over a satellite link, the transmission window size must be increased from 16 to 24 bits. Modifying the protocol, however, is only an option for private networks that do not interface with the Internet since formal acceptance of the modified protocol would be required in order to change the protocol for general Internet use. Another possibility to solve the problem of TCP's small window size is the use of an intercessary computer that would be placed between the source computer and the satellite. This computer would receive all the transmissions from the source computer and acknowledge their receipt very quickly. TCP's window size would not limit throughput between these two computers. The intercessary computer is then responsible for sending the data over the satellite. This computer would wrap the data in another layer and

send it. On the downlink side of the connection, there would be another intercessory computer that knows the intercessory protocol. The intercessory protocol would be stripped off at this point and the data would be sent to its destination. This method will require a little more overhead to transmit the same amount of data, but the performance gained will more than cover the additional overhead. The great advantage of this method is its compatibility with current Internet protocol standards. I have not tested or proven this method, but leave it as topic for further research.

Bibliography

1. Akimaru, Haruo, Marion R. Finley Jr., and Zhisheng Niu, "Elements of the Emerging Broadband Information Highway," IEEE Communications Magazine, June 1997, pp 84-94.
2. Opnet Pubs
3. Comer, Douglas E. Internetworking with TCP, Third Edition, Volume 1, Prentice Hall, New Jersey, 1995, pp 303-324.
4. Akyildiz, Ian F. and Seong-Ho Jeong, "Satellite ATM Networks: A Survey," IEEE Communications Magazine, July 1997, pp 30-43.
5. Ayangoglu Ender, Kai Y Eng, and Mark J. Karol, "Wireless ATM: Limits Challenges, and Proposals," IEEE Personal Communications, August 1996, pp. 18-34.
6. Partridge, Craig and Timothy J. Shepard, "TCP/IP Performance over Satellite Links," IEEE Network, September/October 1997, pp. 44-49.
7. Lakshman, T.V. and Upamanyu Madhow, "The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss," IEEE/ACM Transactions on Networking, Vol 5, No.3, pp 336-350.
8. Wright, David J., "Voice over ATM: An Evaluation of Network Architecture Alternatives," IEEE Network, September/October 1996, pp. 22-27.
9. Sanchez, Jaime, Ralph Martinez, and Michael W. Marcellin, "A Survey of MAC Protocols Proposed for Wireless ATM," IEEE Network, November/December 1997, pp. 52-62.
10. Agrawal, Prathima, Eoin Hyden, Paul Krzyzanowski, Partho Mishra, Mani B. Srivastava, and John Trotter, "SWAN: A Mobile Multimedia Wireless Network," IEEE Personal Communications, April 1996, pp 18-33.
11. Chien, Fang, Helen Chen, and Jim Hutchins, "A Simulation Study of TCP Performance in ATM Networks," IEEE, 1994 pp 1217-1223.
12. Allman, Mark, Hans Kruse and Shawn Ostermann, "An Application-Level Solution to TCP's Satellite Inefficiencies," Proceedings of the First International Workshop on Satellite-

based Information Services (WOSBIS), November 13, 1996.

13. Allman, Mark, Chris Hayes, Hans Kruse, and Shawn Ostermann, "TCP Performance over Satellite Links," Proceedings of the Fifth International Conference on Telecommunications Systems, March 1997.
14. Miller, Mark A., Analyzing Broadband Networks, Second Edition, M&T Books, New York, 1997, pp 403-578.
15. Pritchard, Wilbur L., Henri G. Suyderhoud, and Robert A. Nelson, Satellite Communication Systems Engineering, Second Edition, Prentice Hall, Englewood Cliffs, New Jersey, 1993, pp 361-394.

Appendix A: Error Probability Calculations

BER is 10 ⁻¹⁰ 1.0E-10	10 ⁻⁹ 1.0E-09	10 ⁻⁸ 0.00000001	10 ⁻⁷ 0.0000001	10 ⁻⁶ 0.000001	10 ⁻⁵ 0.00001	10 ⁻⁴ 0.0001	10 ⁻³ 0.001
53 bytes per cell, 8 bits = 1 byte = 424 bits/cell 424							
Probability that a cell is in error 1-(1-BER) ^(#bits/cell)							
4.24000E-08	4.24000E-	4.23999E-06	4.23991E-05	4.23910E-0	0.0042310	0.041515723	0.345714934
23584904.02	2358491.1	235849.5546	23585.40449	2358.98942	236.34823	24.0872598	2.892556561
64.51612454	6.4516144	0.645162652	0.064517494	0.00645297	6.46527E-	6.58903E-05	7.91254E-06
9.999999304	1.0000002	100.0002111	10.00021151	1.00021151	100.21164	10.21299816	1.226443982
Gbit	Gbit	Mbit	Mbit	Mbit	Kbit	Kbit	Kbit
Probability that only a single bit in a cell is error #bits/cell*BER*(1- BER) ^(#bits/cell-1)							
4.24000E-08	4.24000E-	4.23998E-06	4.23982E-05	4.23821E-0	0.0042221	0.040643798	0.277694563
Probability that more than one bit in a cell are in error: ((Probability that a cell is in error)-)probability that a single bit is in error))							
4.74660E-15	7.85426E-	8.98298E-12	8.96718E-10	8.96508E-0	8.94241E-	8.71925E-04	0.068020371
Cells per error:							
2.10677E+14	1.273E+13	111321650259	1115177872	11154391.6	111826.66	1146.887594	14.70147815
-10	-9	-8	-7	-6	-5	-4	-3
Assuming a 155 Mbps ATM network 155 365566.0377							
	cells per second						

576303812.5	34828038.4	304518.5787	3050.551083	30.5126585	0.3059000	0.003137293	4.02157E-05
18.26196582	1.1036339	3.524520587	50.84251805	30.5126585	0.3059000	0.003137293	4.02157E-05
years	years	days	minutes	seconds	seconds	seconds	seconds

bits per error							
8.93271E+16	5.398E+15	4.72004E+13	47283541784	4729462074	47414506. ⁴	486280.3399	6233.426734
89327.09093	5398.3459	47.20037971	472.8354178	4.72946207	47.414506	0.48628034	
Tbit	Tbit	Tbit	Gbit	Gbit	Mbit	Mbit	

bytes per error							
1.11659E+16	6.747E+14	5.90005E+12	59104427230	591182759.2	5926813.3	60785.04249	779.1783417