PERFORMANCE ENHANCEMENT FOR TCP/IP ON A SATELLITE CHANNEL*

J. Scott Stadler and Jay Gelman

MIT Lincoln Laboratory
Lexington, MA

ABSTRACT- *Worldwide usage of the Internet is currently growing at an exponential rate, resulting in a dramatic increase in the demand for the transmission of Internet data via satellite. The TCP/IP protocol suite, which forms the basis of the Internet, performs sub-optimally when confronted with a geostationary satellite link. This may result in poor performance as seen by an interactive user and/or inefficient utilization of precious satellite resources. In response to military needs, MIT Lincoln Laboratory has developed the Wireless IP Suite Enhancer (WISE) which dramatically improves the performance of TCP/IP when it is extended via a satellite link. Tests of a prototype unit conducted on the Satellite Networking Test-bed at Lincoln Laboratory show that WISE yields nearly optimal utilization of the satellite link.*

1 INTRODUCTION

The extension of TCP/IP networks via satellite has a number of important applications in both the commercial and military markets. A satellite has the unique advantage that it provides an instant communications infrastructure to almost anywhere in the world. This is especially important when it is necessary to connect rural areas, whose population density would not warrant the construction of a large wired plant, into the global network. In the military realm even if an existing wired plant exists, it may not be reliably accessible. Here the ability to quickly set up and tear down tactical networks is of paramount importance to prevent troops from out-running their communications capabilities.

While the use of satellites provides the most flexible way to globally extend networks, there are a number of issues that need to be addressed. These issues revolve around the fact that most protocols were optimized to run on terrestrial networks. This is due to the fact that virtually all commercial networks are terrestrial based. Even in the military were wireless networking is of paramount importance, standard protocols, despite there inherent problems, have been retained to capitalize on their huge commercial base. The primary differences between terrestrial and satellite connectivity are the link latency, error rate, and asymmetry. Terrestrial links also typically have much more available BW than their satellite counterparts making satellite BW a precious resource that can not be squandered.

This paper focuses on the extension of TCP/IP networks over geostationary satellites. The work evolved out of an ongoing program aimed at providing data communication capability to future packet switching EHF payloads [1-2]. While many of the solutions presented herein were targeted to that environment, they have a broad applicability to existing circuit oriented satellite systems and could easily be optimized for use in those environments. The paper begins with a description of the problems encountered when a TCP/IP network is extended over a satellite link. Next, the Wireless IP Suite Enhancer (WISE) is and described in detail. Performance results obtained on the EHF Networking Testbed are then presented.

2 TCP VIA SATELLITE

The TCP/IP protocol suite that forms the basis of the Internet was designed to operate over an extremely large range of environments. Despite this robustness, degraded levels of performance may be experienced when the assumptions inherent in its algorithms are violated. For instance, the high delay-bandwidth product and potentially higher bit error rate of a satellite result in a situation in which the satellite link is not efficiently utilized and the TCP/IP performance (as perceived by an interactive user) may be poor. The reduced efficiency and Quality of Service (QoS) can be attributed to three characteristics of a satellite link; Bit Errors, Latency, and Asymmetry each of which will be discussed in the following paragraphs.

2.1 Bit Errors

Most networking protocols were designed for use in a terrestrial environment were the BER is extremely low (typically less than 10^{-10}). On a satellite link, the raw BER is much higher typically ranging from 10^{-2} to 10^{-6}. Forward Error Correction Coding (FEC) is then used to reduce the BER to a level that is acceptable for the application at hand. This leads to an important system design trade as the benefit of lower BER achieved via the use of FEC comes at the price of increased complexity and reduced channel capacity by a factor equal to the code rate. A system designer must determine if it is more cost effective to make the satellite link look like a terrestrial link (from a BER perspective) or to enhance the network protocols so that they can operate in an environment with a higher BER.

Typically, this trade is performed by noting that the addition of an outer code (usually Reed-Solomon) can reduce the BER from typical operational levels of about 10^{-6} to less than 10^{-10} at a cost of only 5-15 percent overhead. This analysis is flawed as it neglects the increased complexity in using the outer code. The correct way to make a comparison is to fix the coding scheme (i.e., the complexity) and then determine the amount of additional energy needed to reduce the BER from its nominal operating point to less than 10^{-10}. Next it must be determined if

* This work was sponsored by the Department of the Air Force under contract number F19628-95-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the author and are not necessarily endorsed by the United States Air Force.

this energy is best put to use by reducing the BER, increasing the link capacity, reducing the antenna size, increasing the number of users that can share a transponder, or by relaxing some other system parameter.

MIT Lincoln Laboratory (MIT/LL) has successfully demonstrated efficient network extension of TCP/IP via satellite at a BER of 10^{-5} using protocol enhancing techniques. For many codes that are in use today, 1-2 dB of energy are needed to reduce the BER from 10^{-5} to 10^{-10}. This energy could alternatively be used to increase the link capacity by 25-58 percent or equivalently to reduce the antenna diameter by 10-20 percent. Note that the potential gain depends heavily on the code used and that as improved codes (particularly Turbo Codes) come into widespread use the surplus energy to be had by operating at a higher BER will diminish.

2.2 Latency

Latency in a terrestrial environment is typically very low (prorogation time across the US is about 30 ms). The latency through a geo-stationary satellite hop (up and down) is about 260 ms resulting in a Round Trip Time (RTT) of about 520 ms. plus coding delays and any additional terrestrial based latencies. The large discrepancy between RTTs in a terrestrial and satellite environment is the primary impediment to efficient network extension in many scenarios of interest. Short of changing orbital parameters (as is the case with LEO systems), nothing can be done to reduce this delay leaving protocol enhancement/boosting as the only option.

TCP operation is affected by several different interactions that occur when it is subject to a high delay-bandwidth link. TCP uses a self-clocking flow control mechanism that insures that new segments are not injected into the network until confirmation that old segments have left the network has been received. The sequence number space used to identify segments is 16 bits long with each count representing 8 bits yielding a maximal delay-bandwidth product of 524 Kbps for standard implementations. For a 520 ms RTT this implies that the maximal data rate that can be utilized by a single TCP connection is about 1 Mbps. Note that TCP implementations can elect to use large windows that increase this dramatically as will be described in Section 4.1.

The TCP flow control mechanism uses an algorithm known as Slow Start. The purpose of this algorithm is to insure that a TCP source dose not start injecting packets into the network at a high rate until it can reasonably expect that they will not cause congestion. It does this by never injecting more than one congestion window worth of unacknowledged data into the network. The congestion window is set to one segment when the connection is established and is increased by one segment for every acknowledgment (ACK) that is received (In fact the algorithm is significantly more involved [3], but this limited description is sufficient for the purpose at hand.). This means that a TCP connection will begin by sending a single segment then it will wait 520 ms and send two segments, followed by four segment 520 ms later, and it will continue in this manner until the satellite link is full. It therefore takes several seconds before a user will receive the full benefit of the satellite links capacity. In some cases, the TCP connection will be finished before the congestion control algorithm has completely ramped up and optimum performance is never achieved. In addition, if multiple TCP connections are sharing a satellite link and one or more of the connections close, the amount of time it takes the remaining connections to absorb the excess bandwidth can be significant (i.e., Connections will be operating in congestion avoidance mode). It is also well known that TCP is unfair to connections that have longer RTTs, giving larger percentages of network resources to connections with lower latencies.

2.3 Asymmetry

Unlike terrestrial networks, satellite links often operate in an asymmetric mode where they receive at a higher data rate than they transmit. This is due to the fact that satellite terminals (especially portable/mobile) are often uplink power limited. Therefore they have a maximum uplink transmit capability, but can receive as many signals on the downlink as desired. In some instance, the asymmetry is simply used to match an asymmetry in actual data flow between locations (e.g., Many WWW servers send out vast quantities of information in response to short requests). It is not uncommon for the ratio of downlink to uplink capacity to approach 100 or more.

TCP's self-clocking mechanism also interacts with asymmetry. This is because most TCP segments sent on the high rate link result in an ACK being sent on the low rate link. The result is that the low rate link may become congested with ACKs causing a reduction in the amount of data being sent on the high rate link even if the high rate link is not congested.

3 GOALS

There are a number of goals that must be considered in the evaluation of alternate methods of improving TCP/IP performance on satellite links. These include transparency, backward compatibility, efficiency, and scalability. A brief description pertaining to the issues associated with each goal is given below.

Transparency – By transparency, we mean that the existence of a protocol enhancer/booster will have no impact on the connections that pass through the enhancer with the exception that the performance will be improved. In addition, the end user should not need to have any knowledge that the enhancer/booster is being used nor should the end user need to follow special procedures to obtain performance improvements. It is also important that the integrity of the protocol not be broken. Therefore if a protocol guarantees end-to-end reliability, the performance enhancer should not compromise that integrity.

Backward Compatibility – The protocol enhancer must work with the existing Internet and intranet infrastructures. It is unrealistic to expect the entire Internet community to change/upgrade their protocol stacks and or applications to accommodate users who access the network via satellite. It is important to note that individual applications have control over several parameters that can dramatically affect performance on a satellite link, therefore if the enhancement technology is not application independent some applications may work well and some may not. Even in the limited scope of Private Network Extension, it is often difficult to get an entire organization to standardize on protocols/applications that would be optimal for extension via satellite.

Efficiency – While satellite links have many advantages over terrestrial links with respect to their ability to provide instant infrastructure, they are typically more expensive than terrestrial alternatives. Because of this it is important that the satellite link be used in as efficient a manner as possible. This implies that the enhancer should correct any aspects of the communication protocols that would cause the satellite link to be used inefficiently.

Scalability – The enhancing approach should scale to high data rates and large numbers of users. While approaches with limited scalability may satisfy near term needs, they can quickly become obsolete in a market that has been experiencing exponential growth.

It is unlikely that an enhancing technique that satisfies all of the goals listed above exists. It is therefore the system designers job to trade off the design goals in a manner that is consistent with the available technology to achieve the goals that are most important to a given application.

4 EXISTING SOLUTIONS

The difficulties encountered in extending TCP/IP via satellite outlined in Section 2 have been recognized for some time now. As such, several methods of overcoming them have been proposed. In this section, we will give a broad overview of the various proposals and evaluate them with respect to the goals outlined in Section 3.

4.1 IETF Standards Track

The Internet Engineering Task Force (IETF) is the standards body responsible for maintaining the integrity of the Internet. Several TCP protocol modifications that have been proposed and incorporated into the TCP specification are aimed at enhancing performance in a high delay-bandwidth environment. Ironically, these modifications grew out of a need to support ultra high-speed fiber networks with small latencies as opposed to moderate rate satellite networks with long latencies although they can improve performance in either environment.

The IETF also has a working group designated TCP Over Satellite whose purpose is to identify existing mechanism that will enable TCP to work better in a satellite environment. In addition ongoing research that may potentially end up in the TCP specification is being evaluated with respect to satellite links.

The TCP protocol modifications that enhance performance in a satellite environment are detailed in RFC 1323 "TCP Extensions for High Performance", RFC 2001 "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", and RFC 2018 "TCP Selective Acknowledgement Options". These algorithms will enhance the performance of TCP connections on a satellite link in many instances. There are however several limitations. These protocol enhancements are ELECTIVE. This means that a given TCP implementation may choose not to implement some or all of the enhancements. If **both** hosts participating in a TCP connection do not implement the enhancements, then neither will experience an improvement in performance. Furthermore if the applications (i.e., client/server) are not developed with these options in mind, then they may not be able to take advantage of them (e.g., If an application overrides a large window default with a small window size). Finally because these modifications were not developed with high latency in mind, no modifications to the Slow Start algorithm have been proposed and the effects described in Section 2.2 remain unchecked.

4.2 Transport Protocols for Satellite Environments

A second approach to TCP in a satellite environment is to recognize that TCP has several fundamental deficiencies when operating via satellite and that the best way to fix them is to propose a new transport protocol. Several efforts are currently underway in this vein. The first is the Satellite Transport Protocol (STC) which is based on SSCOP and was developed at Berkeley. A second group has developed a set of protocols known as the Space Communications Protocol Standards, which is a set of standards that span the entire protocol stack. These protocols are based on the TCP/IP suite and contain the enhancements needed to make them operate efficiently in space. A final effort is the Xpress Transport Protocol (XTP) which was not specifically developed for use with satellites, but has many of the characteristics needed to be effective in that environment (i.e., Error correction and flow control are decoupled.). While changing the transport protocol can clearly improve TCP performance, backward compatibility is completely sacrificed making this approach unattractive.

4.3 Link Layer Protocols

Several Link layer protocols have also been proposed. The Lincoln Laboratory Link Layer (LLLL) was developed by analyzing TCP operation and constructing a protocol that does everything possible at the link layer to improve TCP performance [2]. The LLLL also is responsible for providing reliable connections in the WISE system as will be described in Section 5. Another class of link layers is known as TCP-Aware Link Layers. These Link Layers look at the TCP header, performing fast retransmission and deleting duplicate ACKs at the link layer. This prevents TCP from reducing its congestion window when a retransmission is necessary. In general, link layer approaches can overcome effects due to bit errors on the satellite channel, but there is little that can be done to address the impact of the TCP Slow Start algorithm.

4.4 TCP Spoofing

TCP Spoofing is a technique by which TCP ACK information is manipulated to overcome the problems associated with high latency. In this approach a gateway at the outskirts of the satellite segment of the network will look at the information in the TCP headers. It will then send ACKs to the source and take responsibility for delivering that data itself. When the real ACKs are received from the destination, they will be deleted to prevent the source from becoming confused. This approach is still subject to the window size limitations outlined in Section 2.2 but it can overcome the effects of the Slow Start algorithm.

4.5 TCP Splitting

TCP splitting is an approach that uses a gateway at the periphery of the satellite network to convert TCP traffic into an

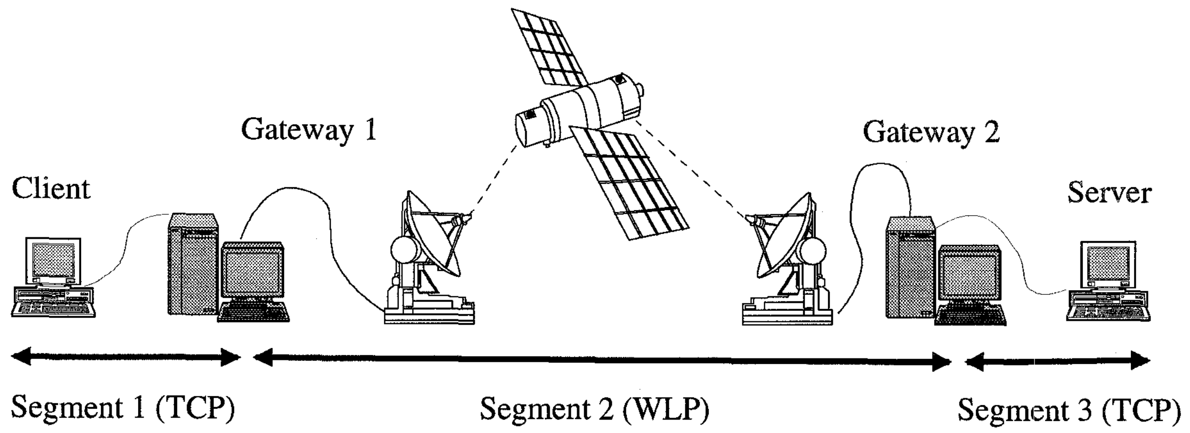


Figure 1 – WISE System Configuration

intermediate protocol that is well suited for the satellite environment. On the other end of the satellite link, the protocol will be converted back to TCP. In some instances the protocol converter simply converts between versions of TCP. This approach can be used to extend the usefulness of the satellite friendly TCP electives outlined in Section 4.1. Here, incoming TCP connections that may not be operating with protocol stacks/applications appropriate for use with satellites will be converted to a connection that takes full advantage of the TCP elective RFCs.

A second approach to TCP splitting involves using a protocol other than TCP on the satellite segment of the link. The second protocol may be specifically tailored for use in a satellite environment. By replacing TCP with another protocol, the performance may be dramatically enhanced. The only draw back to this type of approach is that it must be possible to look at the TCP headers. This means that it will not work with encryption techniques that encrypt the transport header unless the gateway is a trusted system.

5 WIRELESS IP SUITE ENHANCER

The Wireless IP Suite Enhancer (WISE), which was developed at Lincoln Laboratory, improves the performance of the TCP/IP protocol suite in a wireless environment, increasing the wireless link utilization and dramatically improving the performance. The WISE approach consists of software that is added to gateways at the periphery of the wireless segment of the network. WISE operates by transparently splitting the TCP connection into three segments, client to gateway, gateway to gateway via the wireless link, and gateway to server, Figure 1. The client to gateway and gateway to server segments use terrestrial connections and operate using unmodified TCP/IP protocols. The gateway to gateway connection, however, uses a special Wireless Link Protocol (WLP) developed according to the physical characteristics of the wireless link at hand. The WISE software is responsible for converting TCP to WLP upon entering the wireless sub-network and back to TCP upon exiting.

There are many advantages to converting TCP to another protocol at the WISE gateway. Since there is no concept of TCP on the high delay and/or error rate segment of the network, the detrimental effects of latency and errors on TCP are avoided and link utilization is greatly increased. The TCP fairness problem is also circumvented, as TCP connections do not span the satellite. The TCP/IP headers are replaced with a much shorter WLP header and compression of the TCP/IP data may be performed so that fewer bytes need to be sent over the wireless segment. In addition, encryption can be used to protect the data from eavesdropping. Finally, the system can be implemented without making any changes to the TCP/IP code on the gateway, and NO changes of any kind are required to the end users protocols or applications.

The WISE system is very flexible and may be configured in several different manners depending on the network topology and the type of encryption that will be used. In a transparent configuration, it may be inserted anywhere within an IP network (subject to some weak topological constraints) and will boost performance without making any changes outside of the wireless segment. It can also be configured in a "last hop" mode for individual wireless hosts (In this case, WISE must reside on the wireless host.). WISE may also be used in systems with asymmetric and/or mixed media links and will drastically reduce system complexity with respect to existing methods for this environment.

WISE has two components, the connection splitting mechanism and the Wireless Link Protocol. For the case of a satellite link, the Lincoln Laboratory Link Layer (LLLL) was chosen as the WLP. The LLLL provides a reliable connection oriented service to the higher layers, which matches well with the service provided by TCP. Both the LLLL and the connection splitting mechanism are detailed below.

5.1 Satellite Protocol

The Lincoln Laboratory Link Layer provides reliable and ordered connection oriented delivery of packets across a satellite link. It also incorporates features that enhance TCP performance and efficiency. Note that the LLLL will enhance TCP performance when used outside the scope of WISE. This is important, as the WISE system will not be able to split some connections (i.e., Any connection for which the TCP header cannot be read). While the performance of these connections will not improve to the same extent as connections that are split, they will still be enhanced by the LLLL.

The approach taken by the LLLL is threefold. When a new connection is created get data flowing as quickly as possible. Once data starts flowing prevent TCP's flow control algorithm from reducing the flow. This is necessary because the flow control algorithm will confuse errors with congestion and reduce flow when no congestion exists. Note that when congestion is present on either the terrestrial or satellite portions of the link, flow will be reduced as it should be. Finally, errors must be corrected in as efficient a manner as possible.

The guidelines outlined in the previous paragraph are accomplished using a combination of fragmentation and selective repeat ARQ both of which are performed at the link layer. The use of fragmentation decouples the TCP segment size from the link layer packet size. This is important because the larger the TCP segment size, the quicker the flow control algorithm will inject data into the network. On the other hand, larger packets are more susceptible to errors and result in larger amounts of data being retransmitted when an error does occur. Fragmentation allows large TCP segments to be used, while at the same time retaining the benefits of smaller link layer packets. Note that the fragmentation is restricted to the wireless segment of the network and that it is done transparently to IP (similar to the fragmentation of IP packets into ATM cells for IP/ATM).

Selective Repeat ARQ is used to hide any link errors from TCP and thus prevents TCP from miss-interpreting errors as congestion. Link layer acknowledgments containing the entire state of the receive buffer are periodically sent from the receiver to the sender. Because the ACKs are interrupt driven as opposed to data driven, the link layer will work well on asymmetric links (Note that TCP ACKs are still data driven so TCP may still experience asymmetry problems when the connection is not split.). When a packet does need to be retransmitted, it is retransmitted several times (typically 2-3) to insure that the packet will not need more than a single retransmission (This is a very simple form of FEC.). This is important as repeat transmissions will typically cause TCP's timers to expire, negating any benefit of link layer retransmissions. Selective repeat ARQ results in packets being received out of order therefore packet reordering is performed prior to defragmentation by the receiving link layer process.

The use of the LLLL in WISE is two fold. For connections that can be split, it simply provides a reliable connection from one WISE node to its peer node at the other end of the satellite. For connections that cannot be split, it provides an enhanced link that aids TCP by shielding it from link errors.

5.2 Connection splitter

The splitting of TCP connections is illustrated in Figure 2. Here, one or more remote users may communicate with each other or with the terrestrial network via a satellite link. The protocol stack configurations are indicated along the bottom. Note that both of the end users have standard commercial TCP/IP protocol stacks and will run without modification. In addition, they need no knowledge that they are communicating through a wireless link and will not need to use any special procedures. At the periphery of the wireless portion of the network, gateways will be used. Since these gateways will be small in number and they will be aware of the presence of the wireless link, the burden encountered in configuring them will be small and within the scope of the organization providing the "TCP/IP via Satellite" service.

The modified gateways will perform the protocol translation from TCP to LLLL for incoming packets and LLLL to TCP for outgoing packets. The LLLL is responsible for providing a reliable connection oriented link between gateways. Error control may be accomplished using either forward error control coding or with the use of ARQ as outlined above. Note that because WISE performs true protocol conversion, there are no TCP or IP headers to be transmitted on the wireless link, reducing the required BW.

The system operation is as follows. IP packets not containing TCP segments (or whose TCP headers cannot be read) that arrive at the periphery of the wireless network will go up the protocol stack to the IP layer where the standard routing functions will be performed. The packet will go down the protocol stack through the LLLL and over the wireless link. Any errors encountered during transmission will be corrected by the two peer LLLL layers transparently to IP. TCP packets on the other hand, will pass all the way up the protocol stack to the WISE server. Here the TCP connection will be terminated and a virtual circuit will be set up through the LLLL to the WISE server on the other side of the wireless link. This will cause the receiving WISE server to establish a TCP connection to the intended recipient. Once the connections are all established the data is passed over the wireless link to the receiving WISE

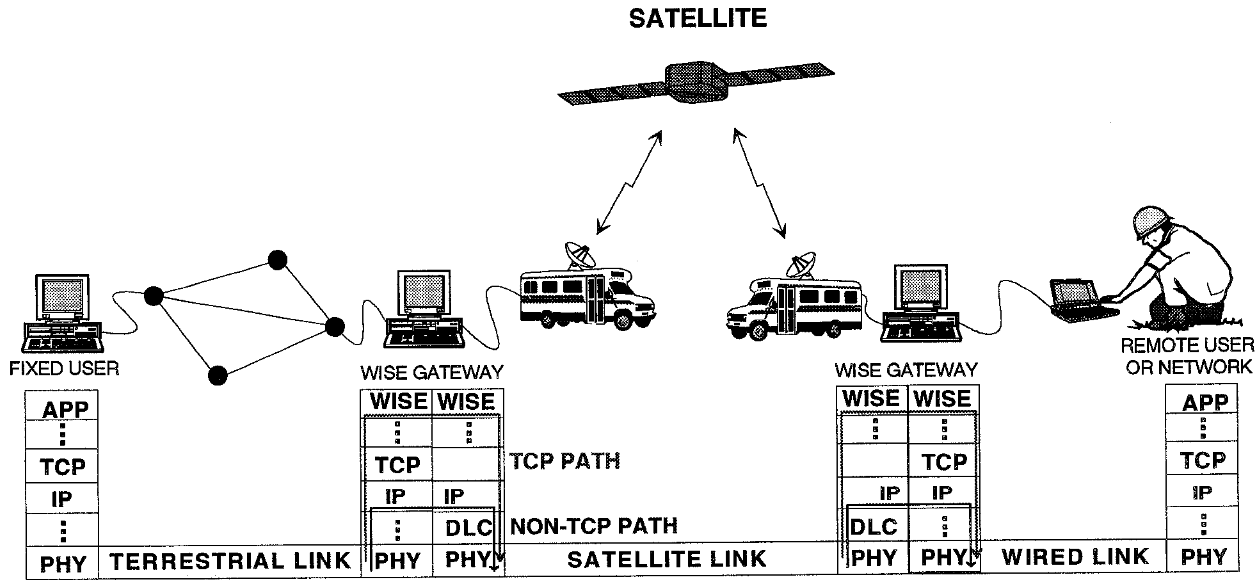


Figure 2 -WISE Protocol Stack

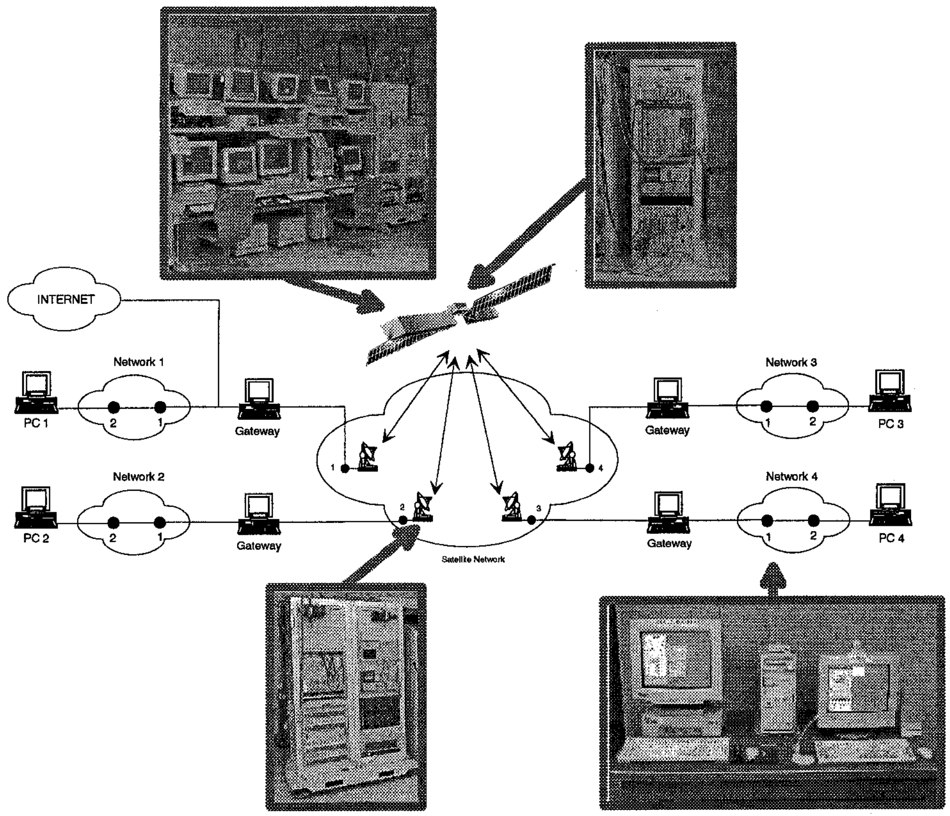


Figure 3 - EHF Networking Test-bed

server where it is relayed (via the TCP connection) to the intended recipient.

WISE has a number of advantages over other approaches. Because the TCP connection is terminated before the wireless link, none of the problems associated with sending TCP via wireless are encountered. TCP and IP headers as well as TCP segments containing only ACKs are not sent on the wireless link reducing the needed BW and eliminating problems associated with asymmetry. In addition, WISE can perform arbitrary compression and/or encryption of the TCP data prior to sending it on the wireless link. Most importantly, WISE is completely backward compatible with existing protocols and applications on the user machines (RFCs 1323, 2001, 2018 may be used but are not needed) and requires no modification to the TCP/IP code on the gateway machines.

Encryption techniques that encrypt the TCP header may have an impact on WISE operation. WISE is fully compatible with application layer encryption and Secure Socket Layer (SSL) encryption, but when operating in a transparent mode, WISE relies on an ability to gather information from the TCP header. If this information is encrypted, then WISE will not be able to split the connection, and only LLLL performance gains will be experienced. IPSEC is one such instance where transparent splitting is not possible. There are, however several approaches to making WISE work with IPSEC. Because many firewall systems require the same visibility into TCP headers as WISE, several equipment manufactures are implementing non-standard IPSEC that leaves the transport headers unencrypted. Clearly if this practice gains acceptance, then the problem is solved. For implementations where the transport header is encrypted, IPSEC can be handled if the gateways can be considered trusted hosts. This would be the case for the extension of private networks for example.

5.3 Processing

In addition to enhancing the TCP connection, WISE has the capability to perform other processing on the data such as compression and encryption. The WISE architecture has the capability to allow TCP connections to be compressed independently or in bulk. As outlined below, it is usually advantageous to compress independently and encrypt in bulk.

Compression – WISE eliminates TCP/IP headers on the satellite portion of the link for the connections it can split. Other traffic (that cannot be split) is sent with the headers in tact. The two cases need to be handled separately. For split connections, WISE can compress each connection separately. This allows adaptive compression algorithms to develop better models of the source being compressed and the compression ratio to be improved. This is important, as a model that works well for text may not achieve a high degree of compression for binary data. Consider bulk compression when it is supporting two connections, an ASCII file transfer and an executable file transfer. The adaptive model will track back and forth between the statistics of the text file and those of the executable file without ever being optimal for either. With independent compression, the model adapts to the statistics of a single type of data yielding optimal compression.

For traffic that must be handled in bulk, such as connections that cannot be split, it is necessary to use an adaptive model that has a small time constant so that it can adjust to changing statistics rapidly enough to allow some compression. Note that in both the case of per connection and bulk compression, it is imperative that the algorithm has some way of disabling itself when it encounters packets that would be expanded due to an unsuccessful compression attempt. This is especially important when dealing with encrypted packets as good encryption implies that on average compression will not be achievable. In the WISE system, bulk compression will be used on connections that cannot be split and still contain the IP header. The compression algorithm will therefore take advantage of the structure of the IP header and may also use the protocol type field to initialize to different dictionaries based on the expected content of the packet.

Encryption – While there are many benefits of performing compression on a per connection basis, encryption is best performed in bulk. There is little advantage to separating connections cryptographically unless there is a large disparity between the protection required by each connection. Even when this is true, it is probably better to bulk encrypt everything at the highest level of protection than to suffer the overhead of multiple encrypted streams. In some extreme cases, it may be necessary to encrypt on a per connection basis (for example when data leakage from one connection to another is of concern), but when this is the case, it usually will have implications that reach far beyond the satellite gateway.

6 SATELLITE NETWORKING TEST-BED

A Satellite Networking Test-bed has been developed by Lincoln Laboratory to test the performance of various protocols and architectures in a realistic satellite environment. The test-bed is depicted in Figure 3. The test-bed consists of a space segment, a terrestrial network, and several remote networks. The topology can be configured as needed to support any testing scenario of interest. The satellite segment is a hardware emulation (at RF) of a processing satellite with an onboard access controller that dynamically allocates resources. It also has an onboard packet switch that is capable of switching packets from various uplinks to the appropriate downlinks. Note that while the satellite segment offers several advanced capabilities, it can also be configured to look like a standard transponded circuit based system and both the LLLL and WISE can operate in these environments as well. The test terminal can be configured to emulate one or more standard terminal types and includes FEC capabilities. The gateway is responsible for interfacing a terrestrial network consisting of one or more unmodified hosts to the satellite segment of the network. It will implement the LLLL and/or WISE. Finally, the user host is an unmodified personal computer running commercial protocols and applications. The test-bed has been use to demonstrate both the LLLL and wise with applications such as Telnet, FTP, and WWW browsers. In addition, the test-bed is instrumented to allow the characterization of system performance at any layer in the protocol stack from physical to application.

7 PERFORMANCE RESULTS

This section gives a sampling of some of the test results obtained on the Satellite Networking test-bed when the LLLL and WISE are used to enhance TCP performance in a tactical scenario. The system parameters are as follows; 100 Kbps link, 10^{-5} BER, and a 1.0 sec. RTT. The results for WWW applications appear below. The graphs contain three plots; TCP alone, TCP/LLLL, and TCP/WISE. For the TCP alone case, TCP was optimized by tuning its parameters while for the other two cases, the default TCP parameters were used. The TCP implementation used did not support the elective RFCs. The results show that TCP/LLLL performs transfers significantly faster (about a factor of 2) than TCP alone and that WISE performs them faster yet (about a factor of 6). The instantaneous utilization plots show how well the satellite channel is being used. The utilization starts at zero when the connection is made, increases and levels off during the transfer, and then returns to zero when the transfer is complete. TCP utilization is poor, averaging about 10 percent. With the LLLL, the utilization increases but becomes limited by the window size set by the application at about 40 percent. With wise, we see a very rapid turn on to nearly 100 percent utilization and a very fast transfer. Similar results have been obtained for other applications such as FTP.

8 CONCLUSIONS

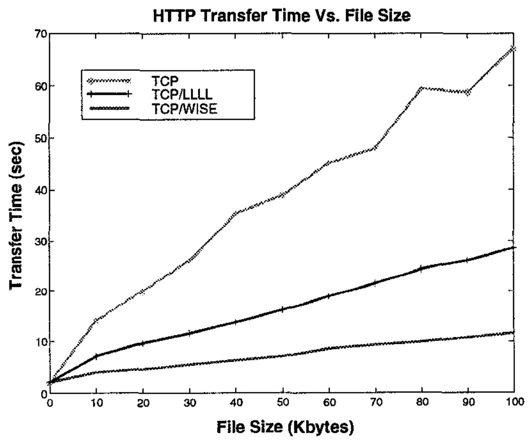
MIT Lincoln Laboratory has devoted considerable effort to solving the problems associated with extending networks via satellite. Lincoln's research program has resulted in the development and test of a TCP enhancer that significantly improves performance in a satellite environment by incorporating additional software at the satellite gateways. The research is ongoing and several enhancements will be added as the work progresses and the users' needs are further assessed.

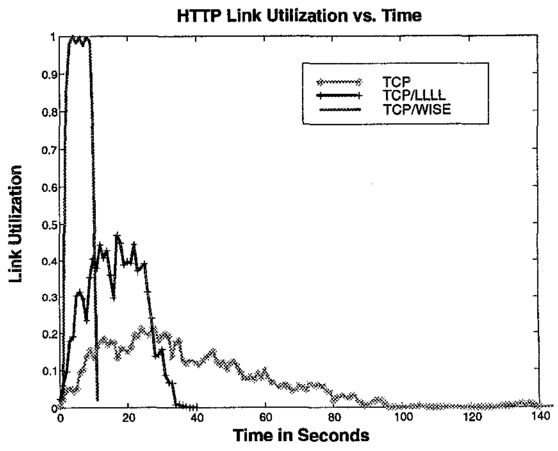
References:

[1] J. S. Stadler and E. Modiano, *"An On-Board Packet Processing Architecture for the Advanced EHF Satellite System"*, MILCOM '97

[2] J. S. Stadler, *"A Link Layer Protocol for Efficient Transmission of TCP/IP via Satellite"*, MILCOM '97

[3] W. R. Stevens, *"TCP/IP Illustrated, Volume I"*, Addison-Wesley 1994


HTTP Transfer Time Vs. File Size
TCP
TCP/LLLL
TCP/WISE
Transfer Time (sec)
File Size (Kbytes)


HTTP Link Utilization vs. Time
TCP
TCP/LLLL
TCP/WISE
Link Utilization
Time in Seconds