Strategy for Developing Expert-System-Based Internet Protocols (TCP/IP)

William D. Ivancic
NASA Lewis Research Center
21000 Brookpark Road MS 54-8; Cleveland, Ohio 44135
Tel.: +1 216 433 3494; Fax: +1 216 433 8705; email: William.D.Ivancic@lerc.nasa.gov

ABSTRACT
The Satellite Networks and Architectures Branch of NASA's Lewis Research is addressing the issue of seamless interoperability of satellite networks with terrestrial networks. One of the major issues is improving reliable transmission protocols such as TCP over long latency and error-prone links. Many tuning parameters are available to enhance the performance of TCP including segment size, timers and window sizes. There are also numerous congestion avoidance algorithms such as slow start, selective retransmission and selective acknowledgment that are utilized to improve performance. This paper provides a strategy to characterize the performance of TCP relative to various parameter settings in a variety of network environments (i.e. LAN, WAN, wireless, satellite, and IP over ATM). This information can then be utilized to develop expert-system-based Internet protocols.

GENERAL PROTOCOLS
Any protocol is either an unreliable protocol or a reliable protocol. An unreliable protocol does not guarantee delivery of a message. There is no feedback from the receiver to the sender acknowledging that the transmission was received correctly. A reliable protocol provides such a feedback mechanism. Thus, a reliable protocol has a closed-loop control system embedded in its underlying structure. This control loop is what we propose to investigate via simulation in order to improve the efficiency of reliable protocols in a long-delay, error-prone environment.

TCP/IP
The TCP/IP protocol suite has been around since the 1970's and continues to evolve. Applications such as Telnet, electronic mail, file transfer all run over or are a part of the TCP/IP protocol suite. TCP/IP was developed to be robust and capable of performing in various network topologies from wired local area networks (LAN) to wireless mobile systems and satellites. Various TCP/IP protocols are reliable protocols and are based on the Transmission Control Protocol (TCP) of the TCP/IP protocol suite. Other TCP/IP protocols are unreliable protocols and are based on the User Datagram Protocol (UDP) of the TCP/IP protocol suite. UDP is an open-loop protocol and will not be considered in this paper. Instead, we will concentrate on the closed-loop Transmission Control Protocol of the TCP/IP suite. From this point on, when we refer to TCP we are referring to the Transmission Control Protocol of the TCP/IP suite.

TCP Control Loop Mechanisms
TCP has a number of control mechanisms to allow for efficient, reliable data transfer while controlling network congestion and maintaining network stability. General control mechanisms include: sliding window, congestion window, receive acknowledgment, retransmission timers, slow start and multiplicative decrease [1,2]. Additional control mechanisms that have been proposed - and in some cases implemented - include: selective acknowledgment and the addition of a timestamps option [3,4 and 5]

Sliding Window Protocol Concept
The sliding window protocol allows the network to be completely saturated with packets - within the limitations of the buffer structures and network delays. In the limit, up to a full window of data may be transmitted before an acknowledgment is received. Retransmission timers are set for each transmitted segment. If the transmission timer expires, one of following events has occurred. Either the transmitted segment was not received, the transmitted segment was in error, or the acknowledgment message was not received or was in error. In current implementations of TCP, for any of these occurrences, it is assumed that the lack of an acknowledgment was due to network congestion as no additional information is available about the network. To date, congestion has been the cause of the vast majority of unacknowledged packets as most networks are considered near error-free. This is not the case for wireless systems such as satellite networks and mobile communications systems.

Figure 1 shows a generalized TCP sliding window and segmentation. For error-free transmission, the common algorithm for segmentation is to pick the maximum segment size (MSS) that can be accepted by the receiver as well as passed through the network without fragmentation. Fragmentation occurs when the MSS is...
larger than the maximum transmission unit (MTU) that can be passed by the routers. Use of MTU discover is a possibility to help determine the optimal segmentation size [6], but may not be practical in a generic network—particularly if the network is relatively dynamic with regard to the transmission time. Furthermore, this segmentation algorithm may not be optimal for noise-prone links, as the larger segments may be detrimental to optimal throughput. Table 1 shows the results of having an errored link of $10^{-6}$ and the resulting errors derived from a binomial distribution function. $P_{ne}$ is the probability that the segment is error-free. $P_{se}$ is the probability that the segment contains one or more errors. For long messages made up of many segments, the probability of multiple retransmissions increases. Thus, from table 1 it is apparent that the segmentation size is one parameter that needs extensive research regarding errored links.

### Table 1: TCP Segmentation Size vs BER

<table>
<thead>
<tr>
<th>Seg (bytes)</th>
<th>$P_{ne}$</th>
<th>$P_{se}$</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>68</td>
<td>0.999456148</td>
<td>0.000543852</td>
<td>Min Allowed</td>
</tr>
<tr>
<td>256</td>
<td>0.997954095</td>
<td>0.002045905</td>
<td>Default</td>
</tr>
<tr>
<td>536</td>
<td>0.995721178</td>
<td>0.004278822</td>
<td></td>
</tr>
<tr>
<td>1024</td>
<td>0.991641459</td>
<td>0.008158541</td>
<td></td>
</tr>
<tr>
<td>1460</td>
<td>0.988387941</td>
<td>0.011612059</td>
<td>Max Ethernet</td>
</tr>
</tbody>
</table>

Equation 1 gives the theoretical maximum throughput for a TCP connection where $TPut$ is the maximum throughput, $RBuff$ is the receive buffer size, and $RTT$ is the round trip time. This is a theoretical limit and assumes no errors and no congestion in the transmission network. With standard TCP the buffer can be as large as 64 kbytes with most implementations providing even smaller windows [7]. From equation 1, it is apparent that for a given $RTT$, the only available parameter to vary in order to improve throughput is to increase the receive buffer (the window size).

**Slow Start**

The slow start algorithm is a congestion avoidance flow control algorithm used to control congestion and maintain stability in the network. This is basically an exponential ramping up of transmitted data segments into the network until one half of the full negotiated receiver buffer size is reached. At that point, the transmission of segments increases linearly.

**Multiplicative Decrease**

Multiplicative Decrease (also known as "congestion avoidance") is a congestion control algorithm. Any expiration of a transmission segment's timer currently assumes loss of a segment most probably caused by congestion. As a reaction to the onset of congestion, the congestion window is reduced by half and the retransmission timer is backed off exponentially. This provides a significant reduction in congestion, but may be triggered by an errored condition in a wireless network segment rather than congestion.

**Selective Acknowledgment**

Selective acknowledgment (SACK) is a technique that has been proposed primarily for LFN. The idea is to acknowledge all segments that have been received correctly so that only those segments that have not been the last received need to be retransmitted. In the current general implementation of TCP, an acknowledge occurs...
for the last successfully received segment. All valid
segments received out of order are not acknowledged
and therefore must be retransmitted. This results in
extremely large volumes of retransmitted data if large
window options are utilized. In addition, the congestion
control and avoidance algorithms are triggered resulting
in decreased performance for packets that were lost due
to errors or minor congestion. As an example, assume
that segment 5 of figure 1 has been lost due to error or
congestion. For selective acknowledgment, segments 1
through 4 would be acknowledged as would segments 6
through 8. Only segment 5 would be retransmitted. For
current general TCP implementations, segment 4 would
be acknowledged with a message that segment 5 is
expected next. Thus segments 5 through 8 would have
to be retransmitted.

Since additional information about the network is gained
by utilizing selective acknowledgment, some
improvements to the congestion control and avoidance
algorithms should be possible that incorporate the
additional knowledge.

**Fast Retransmission and Fast Recover**

Fast retransmission and fast recover are two
complimentary algorithms primarily used in
combination with SACK to improve data throughput.

Fast retransmission is an algorithm in which a segment
is retransmitted prior to its retransmission timer expiring
if multiple acknowledgments - usually 3
acknowledgments - of a previous segment have been
received. The idea being that if multiple
acknowledgments have been received, there must
have been an out-of-sequence segment at the
receiver resulting most probably from a dropped or
errored segment. This technique has been shown to
work well for both regular TCP acknowledgments as
well as selective acknowledgment [8].

Fast recovery compliments and is used along with
the fast retransmission. For the fast recovery
algorithm, multiplicative decrease congestion control
algorithm is implemented without initiating the
slow-start congestion control algorithm. It is
apparent that data is still flowing in the network
otherwise multiple acknowledgments would not have
been received; therefore, utilizing the slow-start
algorithm here is not appropriate.

**Timestamps Option**

A solution proposed to obtain accurate round trip time
measurements (RTTM) is to introduce a timestamp in
each data segment [4]. The receiver reflects these
timestamps back in acknowledgment segments and the
RTTM is performed by simple subtraction. Accurate
RTTM allow the retransmit timers to be accurately set
thus improving the overall TCP performance.

**Initial Window Option**

A proposal has been made to start off the TCP
connection with a window size of at least 1 segment plus
roughly 4 kbytes, one segment and 4380 bytes (3 x
1460), and be at most four times the initial segment size
[9]. This would enabling fewer round trip transactions
(send / acknowledge combinations) for short messages
as well as accelerating the slow start by 3 round trip
times. This proposed initial window size would only be
for the first round trip connection. After a
retransmission time-out, the sender would continue
to slow-start from a window of one segment. Whether this
will significantly improve TCP performance needs
further investigation.

**TCP TUNING**

The more we know about the network the better we can
tune TCP for optimal performance. For a LFN between
high end workstations [Figure 2a], the overall network
is usually known (i.e. the transmitting and receiving
hosts and the bandwidth of the network). Often, we
have control of the routers and switches and may use
TCP over ATM to guarantee link quality. Thus, all
parameters can be optimized for the known network.

![Figure 2: Network Architectures](image-url)
From a mobile or wireless system that has either the source or sink directly at one end of a wireless link [Figure 2b], we very likely know some general characteristics of that portion of the link, characteristics that most likely will dominate the TCP tuning algorithms and can therefore be tuned accordingly. The most challenging tuning scenario occurs when nothing of the network is known until the initial connection is made [Figure 2c]. After passing through a network cloud, data may pass through a satellite or a wireless link experiencing errors and/or long delays. Thus, unbeknownst to the transmitter and receiver at startup, they are utilizing an error-prone LFN, the characteristics of which can only be determined after an initial connection has been established.

RESEARCH TOPICS
From the various control loop mechanisms highlighted in the previous section, we anticipate that the most significant improvements for all TCP transmission will result from implementation of a combination of the timestamp option, selective acknowledgment and fast retransmission. We plan to investigate these techniques and the following questions via simulation:

1) Does selective acknowledgment along with fast retransmit significantly improve the performance of TCP over errored links as well as congested links?
2) Is the optimal segment size different for errored links verses congested links and is the TCP performance significantly improved by optimizing the segment size?
3) Does implementation of 4 segment startup significantly improve the performance of TCP?
4) Should particular options always be active or can they be dynamically activated depending on whether the link has a large delay-bandwidth product or if the link is error prone?
5) Is there a mechanism or information that can be obtained about the link that will allow particular options to be dynamically triggered depending on the link quality and the type or amount of data to be transferred?
6) If certain techniques are identified that dramatically improve TCP over errored links and dynamically changing links, can a probe be introduced to determine the dynamics of the link; thus, enabling "real-time" TCP tuning?

The output of these simulations can later be incorporated into protocol interoperability simulations involving TCP over ATM.

SIMULATION ARCHITECTURE
Figure 3 shows the general architecture for the proposed simulations. The link can be characterized by both an error and a delay component. The network can be congested at either or both the source or destination portions of the local area networks (LAN). Host A will be designated as the source while host B is designated as the sink. A group of hosts on either LAN will be represented utilizing a single host appropriately scaled in order to provide congestion to that portion of the network. Congestion will be generated using distribution functions as well as captured data from various LANs.

![Figure 3: Simulation Architecture](image-url)
CONCLUSIONS

An architecture has been presented that will enable simulation of TCP control loop algorithms under various congested, errored, and delay conditions in order to quickly access the potential improvements (or detriments) that these algorithms provide. Of particular interest is the combined use of SACK and Fast Retransmission on errored links. Information obtained from this research will allow us to suggest modifications to TCP such as dynamic reconfiguration and the introduction of a probe used to obtain link information after the initial end-to-end connections have been established. Promising algorithms will be implemented as modifications of the TCP host kernel software and tested in the NASA Lewis Research Center Satellite/Terrestrial Internet Protocol Testbed.

3 Floyd S., Romanow A.: RFC 2018 TCP Selective Acknowledgment Options, October 1996
5 O’Malley S., Peterson L.: RFC 1263 TCP Extensions Considered Harmful, October 1991
6 Mogul J., Deering S.: RFC 1063 Path MUT Discovery, November 1990
7 Kruse H., Ostermann S., Allman M.: High-Performance TCP/IP Applications for use over Fast Satellite Communications Channels, NASA Grant NCC3-430, August 10, 1996
8 Jacobson V.: Letter to the IETF end2end-interest working group, April 1990
9 Floyd S.: Letter to the IETF end2end-interest working group, February 1997
The Satellite Networks and Architectures Branch of NASA's Lewis Research is addressing the issue of seamless interoperability of satellite networks with terrestrial networks. One of the major issues is improving reliable transmission protocols such as TCP over long latency and error-prone links. Many tuning parameters are available to enhance the performance of TCP including segment size, timers and window sizes. There are also numerous congestion avoidance algorithms such as slow start, selective retransmission and selective acknowledgment that are utilized to improve performance. This paper provides a strategy to characterize the performance of TCP relative to various parameter settings in a variety of network environments (i.e. LAN, WAN, wireless, satellite, and IP over ATM). This information can then be utilized to develop expert-system-based Internet protocols.