

Performance Of Common Data Communications Protocols Over Long Delay Links

An Experimental Examination

Hans Kruse

McClure School of Communication Systems Management
Ohio University
9 S. College Street
Athens, OH 45701

1. Introduction

This paper presents a number of experimental findings regarding the efficiency of terrestrial protocols when they are used over a satellite link at data rates of about 1.5 Mbits/sec. In particular, we present measurements on TCP/IP and a theoretical model which explains the observed performance and points towards possible ways in which this protocol stack can be used more efficiently on a satellite channel.

Two factors combine to motivate such a study at this time. First, satellite communications continue to play an important role in business communication networks. Today's increasingly complex applications demand ever more bandwidth and sophisticated connectivity. In rural areas of the US, this connectivity can be achieved quickly and economically through satellite links, at least until the terrestrial network has a chance to catch up. Outside the developed countries, satellite communications may be the long-term solution to increasing communications needs. Finally, communications to mobile stations such as trucks, ships, or airplanes, demand satellite based solutions.

The second factor is the unexpected longevity of today's "legacy" protocol stacks. While the Internet represents a large TCP/IP installed base, it has always been assumed that OSI-compliant protocols would replace TCP/IP during the migration of the Internet to commercial use. Instead it appears that commercial users are choosing to implement TCP/IP networks, and to connect to the Internet in its present form. The SNA protocol, associated with large mainframes, was expected to disappear in favor of client/server solutions. Commercial users are instead opting to retain the mainframe as a database server and a batch processor.

It seems therefore very likely that both TCP/IP and SNA will not only remain prevalent in the corporate network, but they will have to be used over satellite links as the network is extended into areas without adequate terrestrial infrastructure.

Both the TCP/IP and SNA measurements were conducted over the NASA ACTS satellite. Preliminary results from the SNA/SDLC experiments have been reported previously[6]. In section 2 we describe the setup used to evaluate TCP/IP performance. Section 3 compares the experimental results to a performance model and

discusses the causes for the observed performance limits. We summarize our findings and point to future work in section 4.

2. TCP/IP Experimental Setup and Results

The TCP/IP tests were performed at the ACTS Master Control Station (MCS) located on the NASA Lewis Research Center in Cleveland, Ohio. The ACTS satellite system has been described in detail elsewhere[1] , [7] . For these tests, two traffic terminals at the MCS each provided a T1 (1.544Mbits/sec) interface. The traffic terminals share a 5m transmit and receive antenna and provide essentially error free channels. No transmission error were observed during our tests.

Figure one shows the logical configuration of the test network[5] . Network A operates with the 27.5Mbits/sec traffic terminal number 2, and is designated as “A/27.5” in the figure. Network B/110 operates with the 110Mbits/sec traffic terminal 1. Both A/27.5 and B/110 are Ethernet networks operating over AUI wiring and AUI multiport transceivers. The two networks can be connected using a pair of Cisco Systems 2500 series routers. The router connection is established either over a terrestrial “bypass” wire, or over the satellite. The data rate of the connection can set between 64kbits/sec and 1.536Mbits/sec in 64kbits/sec increments.

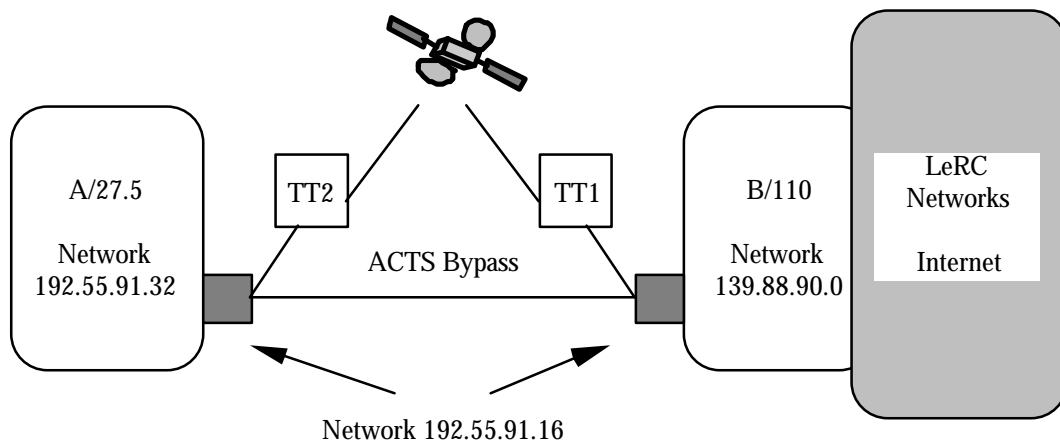


Fig. 1: The network configuration for the TCP/IP tests.

Figures 2a and 2b show the detailed configurations of the two networks. Note that the A/27.5 network is completely self-contained, except for the router connection. The B/110 network has a link to the Internet via the Lewis Research Center internal network. In our tests we were therefore able to completely control traffic on the isolated network only. All tests were repeated at several different times to eliminate the influence of unrelated traffic on the B/110 network. Very few variations in performance were observed, and we believe that such traffic did not play a role in our results.

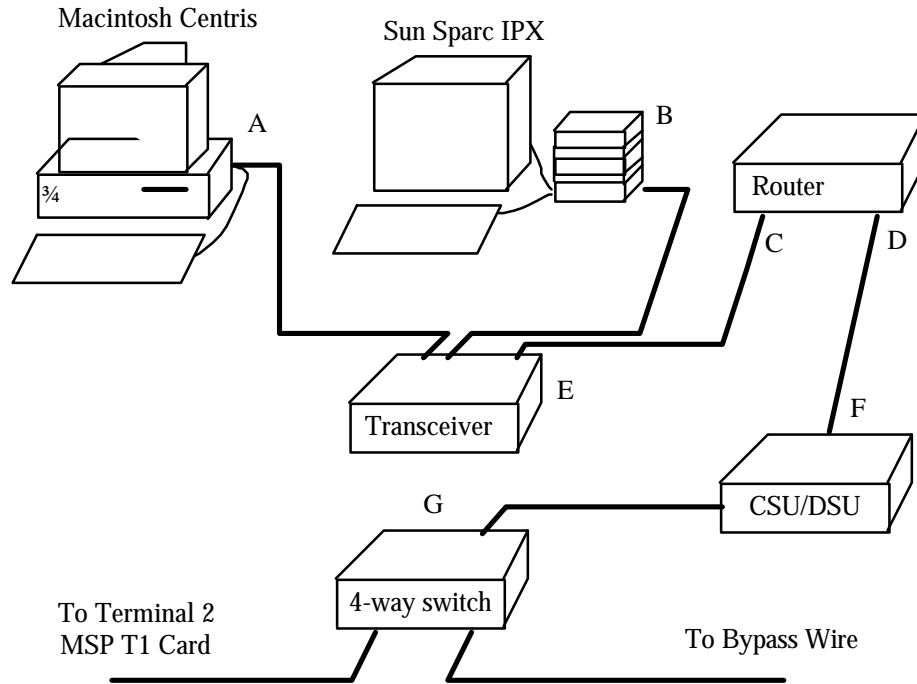


Fig. 2a: The A/27.5 test network.

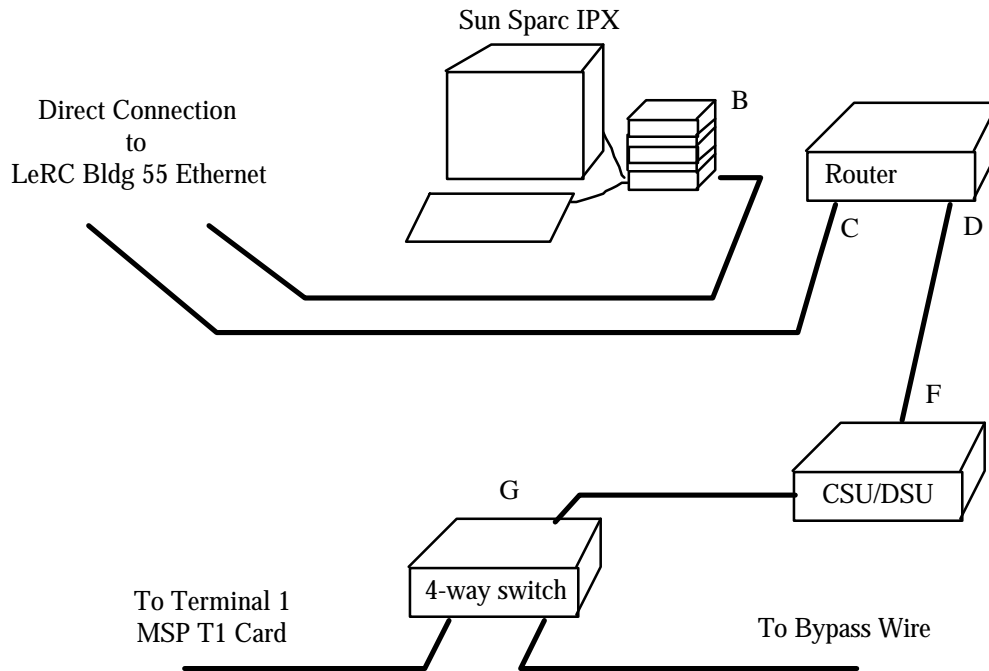


Fig. 2b: The B/110 test network.

The speed of the router connection is set using the DSUs (F). The switch (G) selects the satellite or the terrestrial link. This selection is made past the DSUs to insure that all components which might effect performance are present in both the terrestrial and the satellite channel.

All tests were conducted by establishing FTP connections between the SunSparc workstations (B in figure 2). Both stations use SunOS with unmodified commercial versions of TCP/IP and FTP as supplied by the manufacturer. In each configuration to be tested, three different files are transferred using FTP. The files contain 45kbytes, 100kbytes, and 2Mbytes of ASCII text. Transfer times are recorded using the statistics provided by FTP on the A/27.5 network. In addition, the Mac Centris workstation (A in figure 2a) is used to capture and time-stamp all Ethernet packets during the file transfer. The packets are stored for further detailed analysis.

Using the Unix "Ping" facility, we measured a round-trip delay time over the satellite link of 556ms to 565ms. The terrestrial connection reports a round trip delay of 4ms.

Figure 3 below shows the results obtained in the FTP timing measurements. The throughput is defined as the file size divided by the elapsed time required for the transfer. As such it does not include overhead, and we do not expect the throughput to equal the channel data rate.

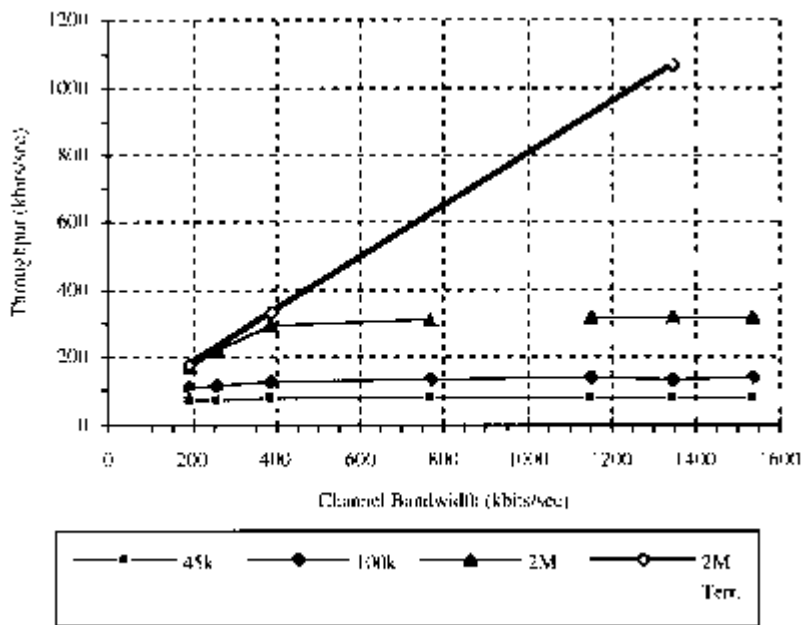


Fig. 3: Measured throughput (as defined in the text) for the transfer of different size files at different channel data rates. The "2M Terr." line is the terrestrial baseline case, all other measurements were made over the satellite channel.

It is striking that the throughput over the satellite link tops out at about 300kbits/sec, regardless of the channel data rate. This effect is expected[2-4] if TCP uses a small window size; we will further examine this issue in the next section.

More puzzling is the pronounced difference in the throughput for different file sizes. Since this is suggestive of a "start-up" phenomenon, we have used the packet trace for

the transfer of a 2Mbyte file at 1.344Mbits/sec to examine if the throughput is time dependent during the file transfer.

Figures 4 and 5 show the number of kbytes transferred as a function of the elapsed time relative to the start of the file transfer.

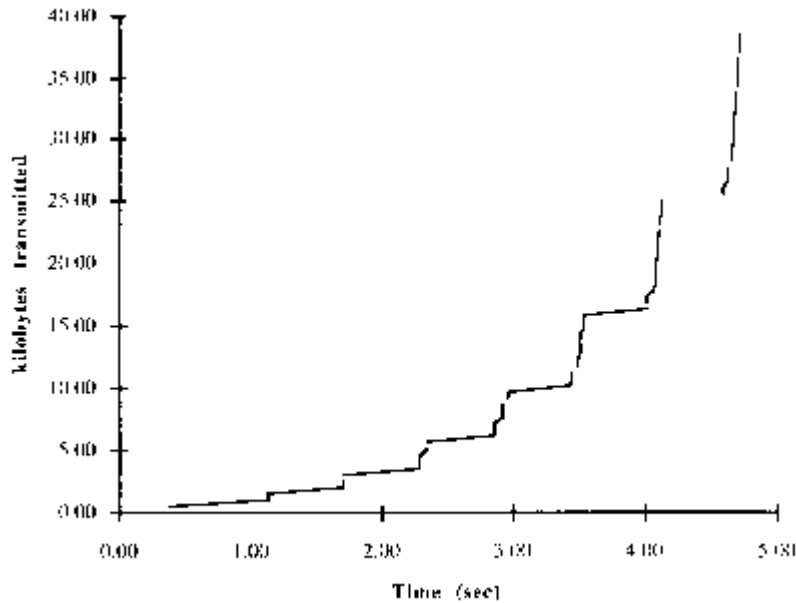


Fig. 4: Amount of data transferred as a function of elapsed time from the start of the file transfer. The data is being sent on a 1.344Mbits/sec satellite channel. The data shown here covers the first 5 seconds of the file transfer.

While both figures cover a time interval of 5 sec, more than 5 times as much data is transferred during the later time interval. Both figures show characteristic “waiting intervals”, or horizontal lines in the graph where little or no data is transferred. These are caused by the flow control window having been exhausted; in this case the sending device has to wait for acknowledgments before transmitting further packets.

Figure 4 shows that much less data is transferred between waiting intervals during the early part of the file transfer. This behavior is consistent with the “slow start” algorithm[11] which must be obeyed by each new TCP connection. Since each FTP file transfer opens a new TCP connection, the “slow start” is invoked for every file[8, 9]. In the next section we model the slow start algorithm to gain a more general understanding of its effect.

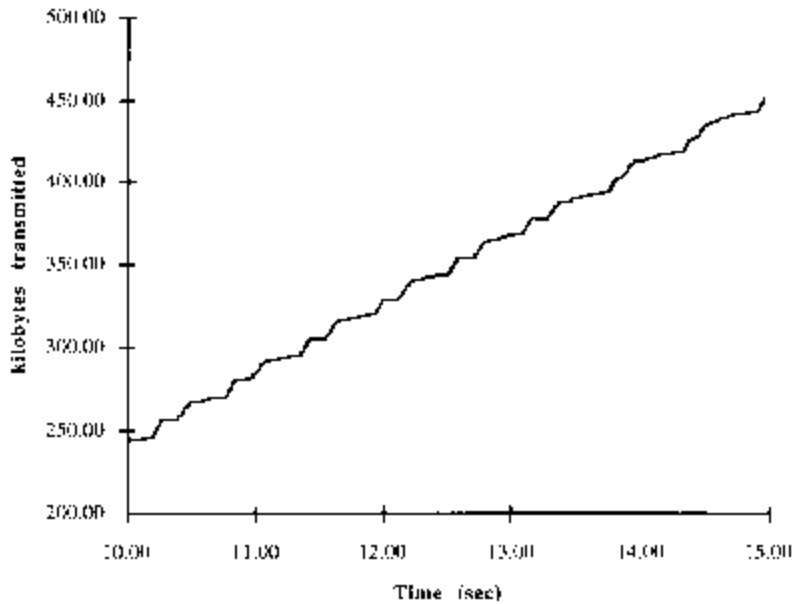


Fig. 5: Same as figure 4, but at a later time in the file transfer.

3. Performance Model and Comparison to the Experiment

The TCP/IP protocol has been examined in detail in the literature[10] . We present here a model which is simplified for the case of a geostationary satellite link, and a link data rate around 1 Mbits/sec. In this case, a large number of frames are in transit over the link before an acknowledgment can be returned. For example, given a T1 link (1.536Mbits/sec) and a round-trip delay of 560ms as measured in our configuration, 107.5kbytes can be in transit before the first acknowledgment arrives.

Most TCP implementations are not capable of window sizes this large (the “extended window option” is required). In addition, even if TCP permits a large window size, it is up to the higher layer applications to take advantage of a large window; many commercial applications do not.

In the case of an error-free link the packet transmission sequence is then dominated by the transmission of a window of information, followed by a wait for the first acknowledgment. Figures 4 and 5 show experimental evidence for this sequence. Figure 6 shows a graphical representation of the time sequence.

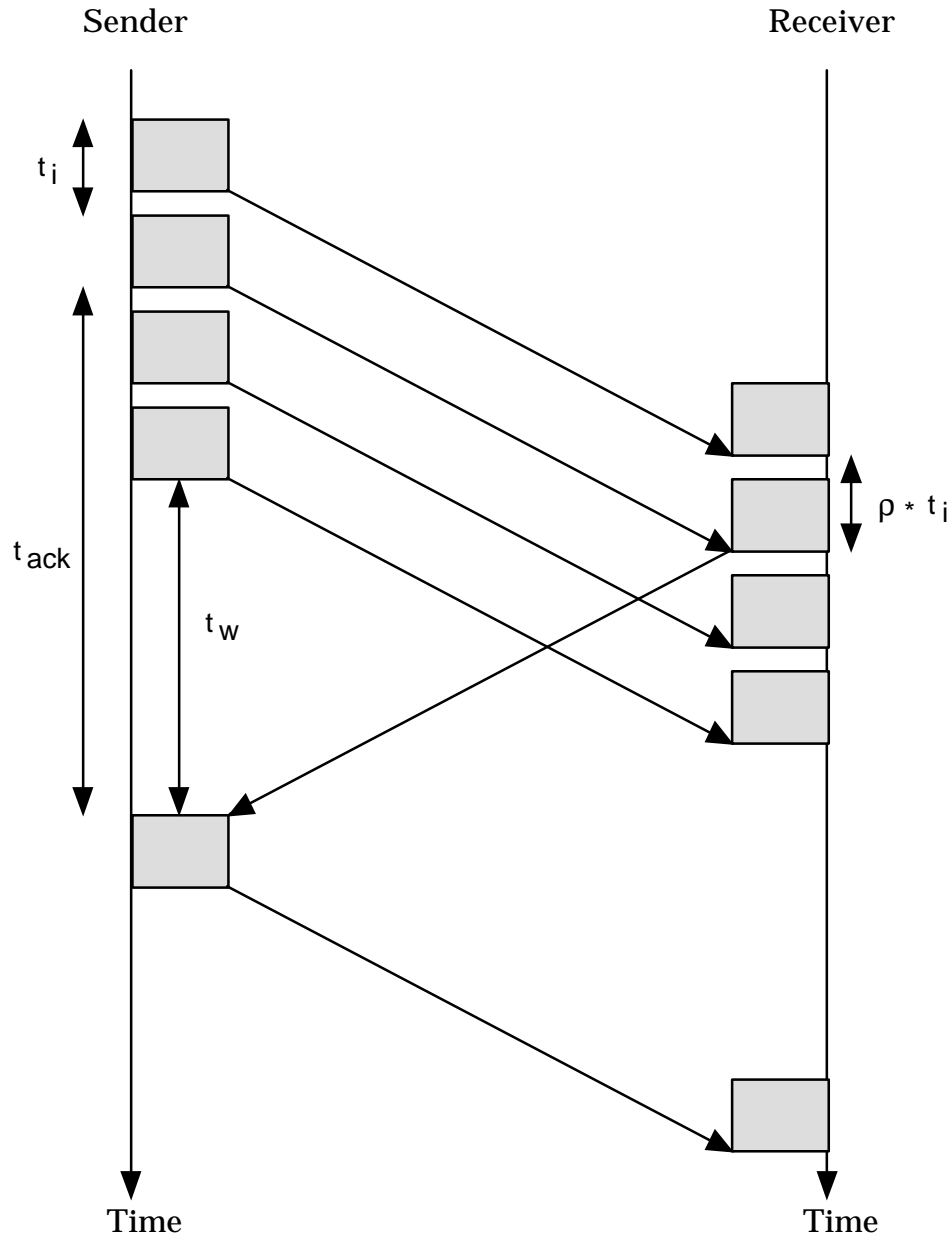


Fig. 6: The sequence of TCP segment transmissions over a satellite link. The time to acknowledge a segment is longer than the time required to transmit all segments permitted by the flow-control window.

From the basic structure shown in Figure 6, we develop the model as follows:

- The application in question (in our case FTP) will set a fixed size flow-control window at the time it requests a TCP connection. The window size (in bytes) will be denoted by W .
- Each frame transmitted over the satellite will correspond to a TCP segment. In a simple network like our test configuration this assumption holds; in more complex situations with fragmentation of segments, the total amount of overhead will

differ, but the principal conclusions from the model will still hold. The total frame length is denoted by l , given as

$$l = l_d + l_{ov} \quad (3.1)$$

with l_d denoting the amount of application data in the frame, and l_{ov} representing the total overhead from TCP through lower protocol layers.

- The time to return an acknowledgment will be labeled t_{ack} , and is given by

$$t_{ack} = 2 \cdot t_p + \frac{l_s}{c} \quad (3.2)$$

where t_p is the one-way propagation delay time, l_s is the total length of an acknowledgment including all overhead, and c is the data rate of the router-to-router channel.

- The number of segments that TCP can send out before it must wait for an acknowledgment is

$$M = \frac{W}{l_d} \quad (3.3)$$

- We will denote by r the number of TCP segments the receiver will “skip” on average before sending an acknowledgment.
- The time required to transfer one segment onto the communications channel is given by

$$t_i = \frac{l}{c} \quad (3.4)$$

- We can now compute the wait time, or “gap” t_w , as shown in figure 6, as

$$t_w = (t_{ack} + r \cdot t_i) - (M - 1) \cdot t_i \quad (3.5)$$

The first term in eq. (3.5) measures the total time required to receive an acknowledgment after a segment has been placed on the outbound communications channel. The second term is related to the number of segments that can still be transmitted before the sender has to wait for the acknowledgment.

- Finally, we compute the “virtual” transmission time per segment as

$$t_v = t_i + \frac{t_w}{M} \quad (3.6)$$

The inverse of t_v will give us the average throughput in segments per unit time, and can be converted easily to a data rate.

Based on our experimental observations, we will set $t_p = 280$ msec, and $r = 1$. Our workstation and router configurations result in $l_d = 512$ bytes, $l_s = 46$ bytes, and $l = 558$ bytes including TCP, IP, and HDLC overhead.

Table 1 shows the effective throughput on a full T1 (1.536Mbits/sec) and a 768kbits/sec (one-half T1) channel, for various window sizes.

| W (bytes) | Throughput using 1.536Mbits/sec (kbits/sec) | Throughput using 768kbits/sec (kbits/sec) |
|--------------|--|--|
| 5,000 | 70.7 | 69.9 |
| 25,000 | 353.3 | 349.6 |
| 65,000 | 918.6 | n/a |
| 95,000 | 1343.0 | n/a |

Table 1: Model throughput for different flow-control window sizes.

The “n/a” entries denote cases where there is no wait time, and where the model is therefore not applicable. The model results confirm the known conclusions that the flow-control window limits throughput regardless of channel speed.

Our FTP implementation uses $W = 24,576$. The model results in Table 1 compare well with the measured throughput of 321kbits/sec for the transfer of a 2Mbyte file over the 1.536Mbits/sec channel; the model cannot, however, explain the much lower throughput for the transfer of the smaller files.

In order to model the slow-start algorithm which we believe to be responsible for the lower experimental throughput, we expand the model as follows:

- The slow start algorithm begins with an effective window size of 1 segment. Each acknowledgment received adds an additional segment to the window for each segment acknowledged. In our analysis, all segments in a window are transmitted before the acknowledgments arrive. Therefore, each time an acknowledgment closes a wait time period, it also doubles the window size, up to the flow-control window size, W .
- We define a time-dependent number of segments in the current window as M_j . M_j will replace M in equations (3.5) and (3.6). M_j is defined as follows:

$$\begin{aligned} M_0 &= 1 \\ M_j &= \min(2 \cdot M_{j-1}, M) \end{aligned} \quad (3.7)$$

- Assuming that n cycles (i.e. sequences of transmitting the window and waiting for an acknowledgment) are required to transmit a file, we write, equivalent to equation (3.5), an expression for the total wait time during the file transfer,

$$T_w = \sum_{j=0}^n \left[(t_{ack} + r \cdot t_i) - (M_j - 1) \cdot t_i \right] \quad (3.8)$$

- The virtual transmission time per segment is then defined in analogy with equation 3.6, as

$$t'_v = t_i + \frac{T_w}{\sum_{j=1}^n M_j} \quad (3.9)$$

Figure 7 compares this model with some of the experimental results reported earlier for channel data rates of 1.536Mbits/sec and 384kbits/sec. The agreement between experiment and model is good, albeit less so for the very large file size. We assume that random additional delays and overhead traffic on the network keeps the achievable throughput somewhat below the model results.

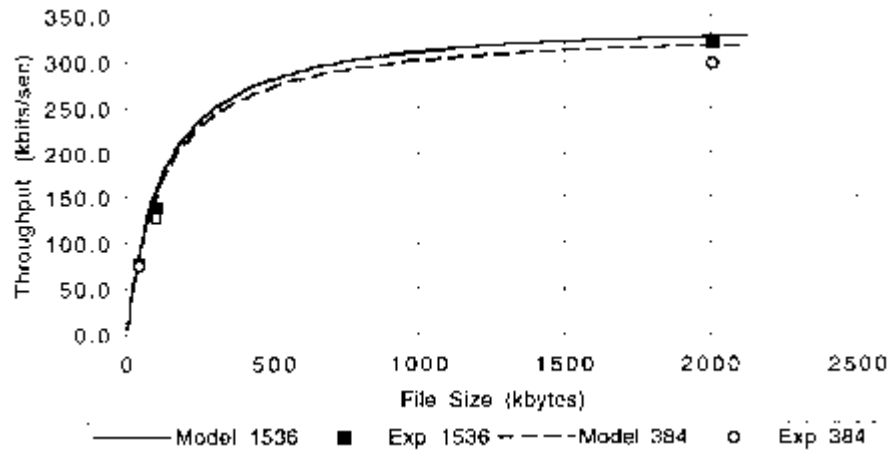


Fig. 7: Comparison of the slow-start model and the experimental results obtained at channel speeds of 1.536Mbits/sec (solid line and filled squares), and 384kbits/sec (dashed line and open circles).

It is also notable that the channel data rate has almost no impact on the achievable throughput as long as the time to transmit a full flow-control window remains less than the round-trip delay time.

Figure 8 shows the model predictions for an FTP implementation with a larger TCP window size. We compare the window size used for our experiment to the largest TCP window size available without the extended window size option. Note that the throughput for transfers of large files is much higher for the larger window size, consistent with the results stated earlier in the section. However, there is no appreciable improvement for file sizes below 100kbytes, since the slow-start algorithm does not permit the full window to be used until very late in the file transfer.

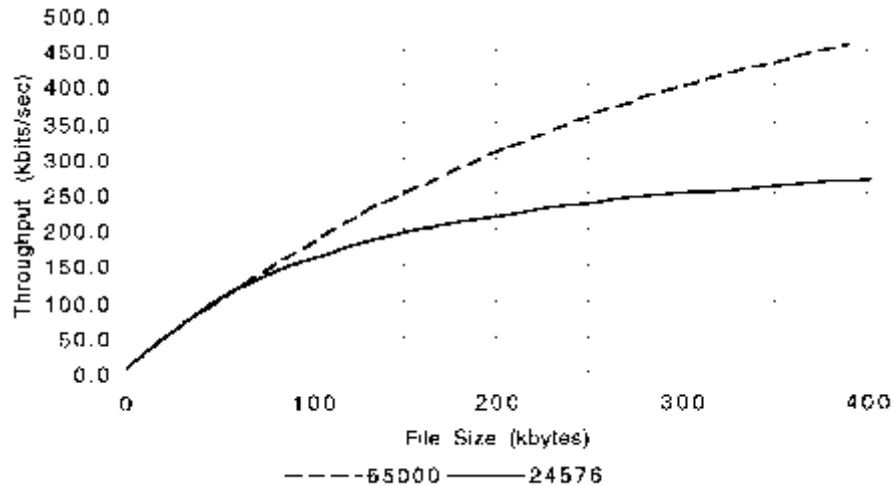


Fig. 8: Model predictions for achievable throughput using larger TCP window sizes. The lines are labeled with the flow-control window size in bytes.

4. Conclusions

In this paper we have presented experimental measurements of transmission efficiency of the TCP protocol over a geosynchronous satellite link at T1 speeds. We have also shown that the observed efficiency is determined by two factors: the TCP window size, which determines the achievable throughput for very long, sustained transmissions, and by the slow-start algorithm, which dominates during the first 100kbytes of the transmission.

We believe these results to be significant to future users of the TCP/IP protocols and applications on satellite links. Many applications of the Internet involve the use of FTP, HTTP (the main protocol in the World Wide Web), and other, similar protocols. The common factor in these protocols is that they use the opening and closing of TCP connections to distinguish “units” of information. These units are files in the case of FTP, and linked objects in the case of HTTP.

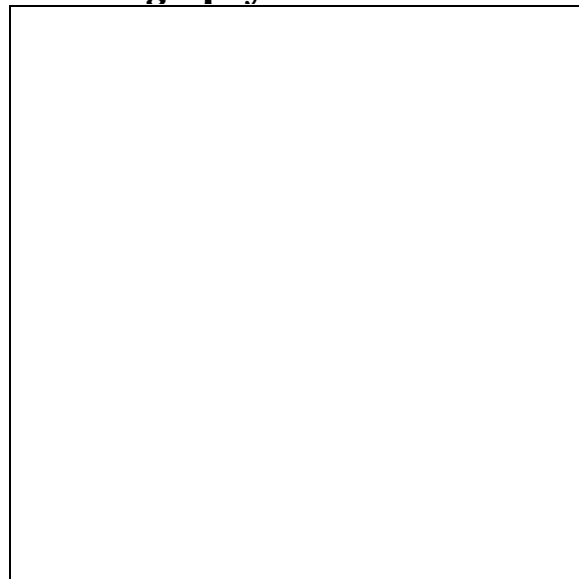
Since many of the units of information in these applications are 100kbytes or less in size, the TCP window size may be much less important to efficiency than previously assumed. In contrast, the “slow-start” congestion control mechanism plays a very large role in determining the throughput on the network.

Given the latency of the standards setting process, no quick solutions are in sight. The combination of the delay in determining an appropriate solution, and the delay until the changes are made in commercial products, suggests that the underlying TCP slow start algorithm will be in use for several years to come.

We suggest that it may be appropriate to begin examining the applications using TCP to determine if modifications can be made here to circumvent the performance

limitations inherent in the current TCP on satellite links until a TCP-based solution can be found and widely implemented.

5. Bibliography



- [1] Bauer, R. and T. vonDeak. *Advanced Communications Technology Satellite (ACTS) and Experiments Program Descriptive Overview*. NASA Lewis Research Center 1991.
- [2] Jacobson, V. and R. Braden. *TCP Extensions for Long-Delay Paths*. LBL 1988; Internet [RFC 1072](#).
- [3] Jacobson, V., R. Braden, and D. Borman. *TCP Extensions for High Performance*. LBL 1992; Internet [RFC 1323](#).
- [4] Jacobson, V., R. Braden, and L. Zhang. *TCP Extension for High-Speed Paths*. LBL 1990; Internet [RFC 1185](#).
- [5] Kruse, H. *Design of TCP/IP Data Communications Demonstration Capabilities*. Ohio University 1994; General Report [OU-CSM-HK94-GR001](#).
- [6] Kruse, H. *Disaster Recovery Via ACTS: Final Report to the NASA ACTS Experiments Office*. Ohio University 1994; Technical Report [OU-CSM-HK94-TR002](#).
- [7] Kruse, H. *T1 Data Communications on an Intelligent Ka Band Satellite: Initial Results from the ACTS Project*. in *2nd International Conference on Telecommunication Systems Modeling and Design*. 1994. Nashville.
- [8] Postel, J. *TRANSMISSION CONTROL PROTOCOL*. ISI 1981; Internet [RFC 793](#).

- [9] Postel, J. and J. Reynolds. *FILE TRANSFER PROTOCOL (FTP)*. ISI 1985; Internet RFC 959.
- [10] Schwartz, M., *Telecommunications Networks: Protocols, Modeling and Analysis*, 1988, Addison-Wesley.
- [11] Stevens, W.R., *TCP/IP Illustrated: The Protocols*, Vol. 1. 1994, Addison-Wesley.

