

International Telecommunication Union

ITU-R
Radiocommunication Sector of ITU

Report ITU-R S.2148
(09/2009)

**Transmission control protocol (TCP)
over satellite networks**

S Series
Fixed-satellite service



International
Telecommunication
Union

Foreword

The role of the Radiocommunication Sector is to ensure the rational, equitable, efficient and economical use of the radio-frequency spectrum by all radiocommunication services, including satellite services, and carry out studies without limit of frequency range on the basis of which Recommendations are adopted.

The regulatory and policy functions of the Radiocommunication Sector are performed by World and Regional Radiocommunication Conferences and Radiocommunication Assemblies supported by Study Groups.

Policy on Intellectual Property Right (IPR)

ITU-R policy on IPR is described in the Common Patent Policy for ITU-T/ITU-R/ISO/IEC referenced in Annex 1 of Resolution ITU-R 1. Forms to be used for the submission of patent statements and licensing declarations by patent holders are available from <http://www.itu.int/ITU-R/go/patents/en> where the Guidelines for Implementation of the Common Patent Policy for ITU-T/ITU-R/ISO/IEC and the ITU-R patent information database can also be found.

Series of ITU-R Reports

(Also available online at <http://www.itu.int/publ/R-REP/en>)

Series	Title
BO	Satellite delivery
BR	Recording for production, archival and play-out; film for television
BS	Broadcasting service (sound)
BT	Broadcasting service (television)
F	Fixed service
M	Mobile, radiodetermination, amateur and related satellite services
P	Radiowave propagation
RA	Radio astronomy
RS	Remote sensing systems
S	Fixed-satellite service
SA	Space applications and meteorology
SF	Frequency sharing and coordination between fixed-satellite and fixed service systems
SM	Spectrum management

Note: This ITU-R Report was approved in English by the Study Group under the procedure detailed in Resolution ITU-R 1.

Electronic Publication
Geneva, 2010

© ITU 2010

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without written permission of ITU.

REPORT ITU-R S.2148

Transmission control protocol (TCP) over satellite networks

(2009)

TABLE OF CONTENTS

	<i>Page</i>
1 Introduction	3
2 Satellite system reference models.....	3
2.1 Point-to-point links	3
2.2 VSAT networks	3
2.2.1 Star topology	3
2.2.2 Mesh topology.....	4
2.3 Broadband access.....	4
3 TCP limitations over satellite links	5
3.1 Bandwidth-delay product.....	5
3.2 Slow start and congestion avoidance	6
3.3 Fast retransmit and fast recovery	7
3.4 Effect of bit errors on TCP throughput.....	8
3.5 Asymmetric networks	8
3.6 Advice to designers.....	9
4 TCP enhancement methodologies	9
4.1 Variations of baseline TCP	9
4.2 Segment splitting methodologies.....	16
4.2.1 Two-segment splitting methodology	16
4.2.2 Three-segment splitting methodology	17
4.2.3 Discussion	19
4.3 Caching and spoofing	19
4.3.1 Caching	19
4.3.2 Spoofing.....	19
4.3.3 Spoofing and caching.....	19
5 Performance enhancing proxies	20
5.1 TCP spoofing	21
5.2 TCP splitting.....	21
5.3 Other PEP mechanisms.....	21
5.4 Implications of using PEP	22
5.4.1 End-to-end security	22
5.4.2 End-to-end reliability	22
6 Other transmission control protocols.....	22
6.1 Space communication protocol specification – transport protocol (SCPS-TP)..	22
6.2 Xpress transport protocol.....	22
6.3 Stream control transmission protocol	23
6.4 Comparison between transmission control protocols	24
7 Conclusion.....	24

List of acronyms

ACK	Acknowledgement
ATM	Asynchronous transfer mode
ARQ	Automatic repeat request
BDP	Bandwidth-delay product
BER	Bit-error ratio
cwnd	Congestion window (variable in TCP)
DACK	Delayed acknowledgement
ECN	Explicit congestion notification
FIN	Final segment (in a TCP connection)
GSO	Geostationary-satellite orbit
GW	Gateway
HEO	High-Earth orbit
IETF	Internet engineering task force
IP	Internet protocol
IPSEC	IP security protocol
ISP	Internet service provider
LAN	Local area network
LEO	Low-Earth orbit
MEO	Medium-Earth orbit
MPLS	Multiprotocol label switching
MTU	Maximum transmission unit
PAWS	Protect against wrapped sequence(s)
PEP	Performance enhancing proxy
RAM	Random access memory
RBP	Rate-based pacing
RFC	Request for comments (issued by the IETF)
RTT	Round trip time
RTTM	RTT measurement
SACK	Selective acknowledgment
SCPS-TP	Space communication protocol specification – transport protocol
SCTP	Stream control transmission protocol
SYN	Synchronous start segment (used to establish a TCP connection)
T/TCP	TCP for transactions
TCP	Transmission control protocol
UDP	User datagram protocol
VSAT	Very small aperture terminal
XTP	Express transport protocol

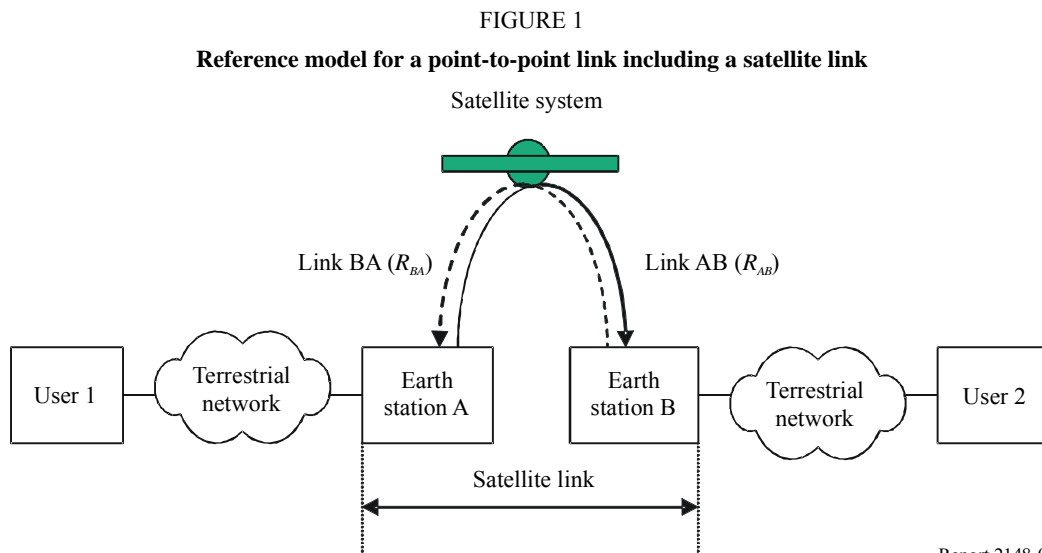
1 Introduction

This Report presents reference models of networks including a satellite link, to carry IP packets, followed by a description of the limitations of transmission control protocol (TCP) over satellite links, as well as, various methodologies to overcome the limitations.

2 Satellite system reference models

2.1 Point-to-point links

Figure 1 provides a reference model for a network carrying IP packet transmissions. The network consists of a satellite link and associated terrestrial networks between two end-users. The satellite link is bidirectional and consists of link AB (from earth station A to earth station B with an information bit rate, R_{AB}) and of link BA (from earth station B to earth station A with an information bit rate, R_{BA}). The terrestrial networks can employ various data link layer protocols (e.g., asynchronous transfer mode (ATM), frame relay, MPLS).



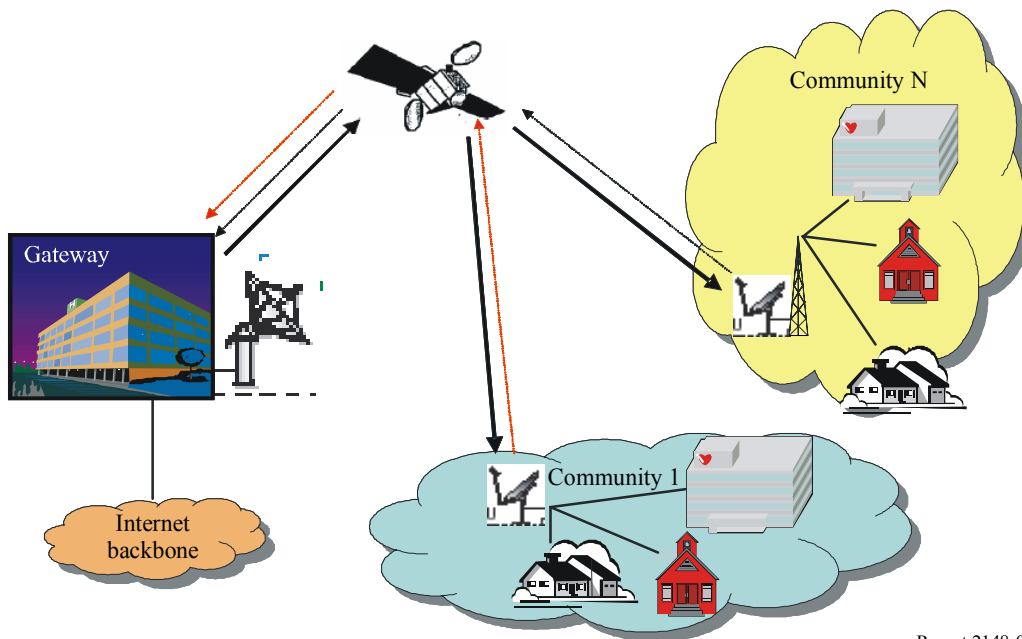
NOTE 1 – The reference model above considers only one satellite hop. Throughout this Report the techniques that segment the TCP connection to improve TCP performance over satellite links are described for one satellite hop. However an end-to-end connection may include several satellite hops. In this case, such techniques will have to be implemented over each individual satellite link.

2.2 VSAT networks

2.2.1 Star topology

Figure 2 depicts the standard star configuration in which signals from various remote users connect to a gateway earth station which in turn connects to terrestrial network.

FIGURE 2
Star topology

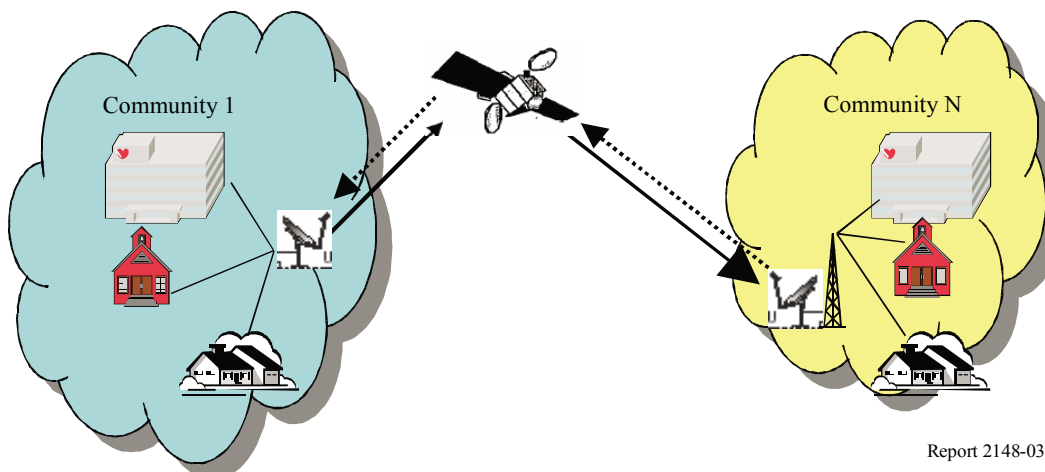


Report 2148-02

2.2.2 Mesh topology

Figure 3 illustrates a mesh configuration whereby any pair of earth stations can be connected directly via satellite.

FIGURE 3
Mesh topology



Report 2148-03

2.3 Broadband access

Even if not completely similar to very small aperture terminal (VSAT) networks, broadband access networks generally use the same topologies (i.e. star or mesh).

3 TCP limitations over satellite links

The TCP cannot distinguish the performance degradation caused by link errors from congestion. It assumes that any loss in the network is due to congestion only and the sender responds by reducing its packet transfer rate.

The baseline TCP (TCP Reno) specifies slow start, congestion avoidance, fast retransmit and fast recovery for congestion control. The TCP uses window flow control mechanism in which the transmission window allows the receiving TCP to control the amount of data being sent to it at any given time. The receiver advertises a window size to the sender. The window measures, in byte/s, the amount of unacknowledged data that the sender can have in transit to the receiver.

3.1 Bandwidth-delay product

The bandwidth-delay product (BDP) defines the amount of data a TCP connection should have “in flight” (data that has been transmitted, but not yet acknowledged) at any time to fully utilize the available channel capacity. The delay is the round trip time (RTT) and the bandwidth is the capacity of the bottleneck link in the path.

For links with a large BDP, such as in geostationary satellite networks, TCP senders and receivers with limited congestion/receive windows will not be able to take advantage of the available bandwidth. The standard maximum TCP window of 65 535 byte/s is not adequate to allow a single TCP connection to utilize the entire bandwidth available on some satellite channels. In a loss-free network the TCP throughput is limited by equation (1):

$$\text{Maximum throughput} = \frac{\text{Window size}}{\text{RTT}} \quad (1)$$

Therefore, when using the maximum TCP window size of 64 kbyte/s and satellite links with variable RTT, the maximum throughput is as follows:

TABLE 1
Maximum throughput according to RTT values

Satellite network type	RTT (ms)	Maximum throughput (kbyte/s)
LEO	~20	~3 200
MEO	~200	~320
HEO	~600	~110
GSO	~520	~120

NOTE 1 – The above-mentioned RTT do not take into account any buffer delay but are computed on the basis of the propagation delay.

IETF¹ Request for Comments 3150² recommends performance of network paths that traverse “very low bit-rate” links. It is applicable for any network where hosts can saturate available bandwidth, but the design space explicitly includes connections that traverse 56 kbit/s modem links or 4.8 kbit/s

¹ Internet Engineering Task Force.

² End-to-end Performance implications of slow links.

wireless access links. Some of the discussion is common with Request for Comments 2689³, using header compression. It focuses more on traditional data applications for which “best-effort delivery” is appropriate.

3.2 Slow start and congestion avoidance

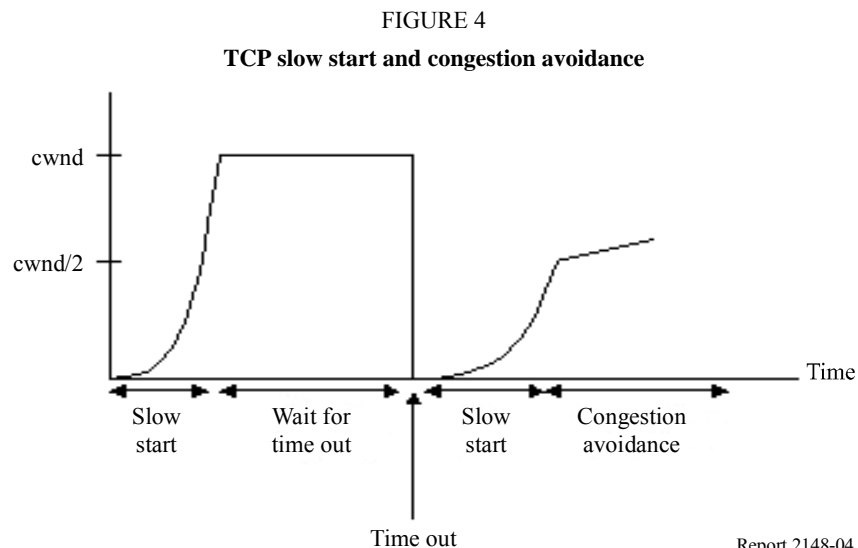
The TCP sender maintains a congestion window to measure the network capacity. The number of unacknowledged packets in the network is limited to this value (or to the receiver advertised window whichever is lower). At the start of a TCP connection, the congestion window is set to one TCP segment. It increases by one segment on the receipt of each new acknowledgment until it reaches its maximum value of 64 kbyte/s. The sender maintains a retransmission time out for the last unacknowledged packet. Congestion is detected by the expiration of the retransmission time out. When the timer expires, the sender saves the value of half the congestion window (called slow start threshold) and sets it to one segment. The sender then retransmits segments starting from the lost segment. The congestion window is increased by one segment on the receipt of each new acknowledgment until it reaches the slow start threshold. This is the slow start phase. After that, the congestion window increases by one segment every RTT. This results in a linear increase of the congestion window every RTT and is called the congestion avoidance phase. Figure 4 shows the slow start and congestion avoidance phases for a typical TCP connection (in Fig. 4, cwnd stands for congestion window).

The time required by the slow start mechanism to reach a bit rate B is given by equation (2):

$$\text{Slow start duration} = \text{RTT} \left(1 + \log_2 \frac{B \cdot \text{RTT}}{l} \right) \quad (2)$$

where l is the average packet length expressed in bits.

Table 2 shows the duration of slow start phase for various satellite orbits and different values of bit rates B , when $l = 1$ kbit.



³ RFC 2689 “Providing integrated services over low-bit rate links”.

TABLE 2

Duration of slow start for various satellite orbits

Satellite type	(RTT) (ms)	Slow start duration (s)		
		$B = 1$ Mbit/s	$B = 10$ Mbit/s	$B = 155$ Mbit/s
LEO	~20	0.05	0.11	0.19
MEO	~200	1.14	1.80	2.59
HEO	~600	4.36	6.35	8.73
GSO	~520	3.67	5.40	7.45

If the delayed acknowledgment mechanism is implemented then the time required by slow start to reach the bit rate B is given by the following formula:

$$\text{Slow start duration} = \text{RTT} \left(1 + \log_{1.5} \frac{B \cdot \text{RTT}}{l} \right) \quad (3)$$

It implies that the slow start duration becomes even longer compared to the previous case. Thus, delayed acknowledgements also waste capacity during the slow start phase.

In the congestion avoidance phase, the increase of data rate is a function of the bandwidth-delay product. In fact, during each RTT, the data rate is increased by $l/(B \cdot \text{RTT})$. So if a TCP connection is in the congestion avoidance phase and some additional bandwidth becomes available, this connection will not use it for a long time. This time will be longer in the presence of transmission losses. Therefore the congestion avoidance mechanism in satellite networks with high RTT performs lower than in a terrestrial network.

3.3 Fast retransmit and fast recovery

Currently TCP implementations use a coarse granularity (typically 500 ms) timer for the retransmission time out. As a result, during congestion, the TCP connection loses time waiting for the time out. In Fig. 4, the horizontal line (at the cwnd value) shows the time lost when waiting for a time out to occur.

During this time, TCP neither sends new packets nor retransmits lost packets. Moreover, once the time out occurs, the congestion window is set to one segment, and the connection takes several round trips to efficiently utilize the network. TCP Reno implements the fast retransmit and recovery algorithms that enable the connection to quickly recover from isolated segment losses.

If the network drops a segment, the subsequent segments arriving at the receiver are out-of-order. For each of them, the TCP receiver immediately sends an acknowledgement to the sender indicating the sequence number of the missing segment. This acknowledgement is called a duplicate acknowledgement. When the sender receives three duplicate acknowledgements, it concludes that the segment indicated by the acknowledgements has been lost and immediately retransmits the lost segment. The sender then reduces the congestion window by half plus three segments and also saves half the original congestion window value in the slow start threshold. For each subsequent duplicate acknowledgement, the sender increases the congestion window by one and tries to send a new segment. Effectively, the sender waits for half a round trip before sending one segment for each subsequent duplicate acknowledgement it receives. As a result, the sender maintains the network link at half capacity at the time of fast retransmit.

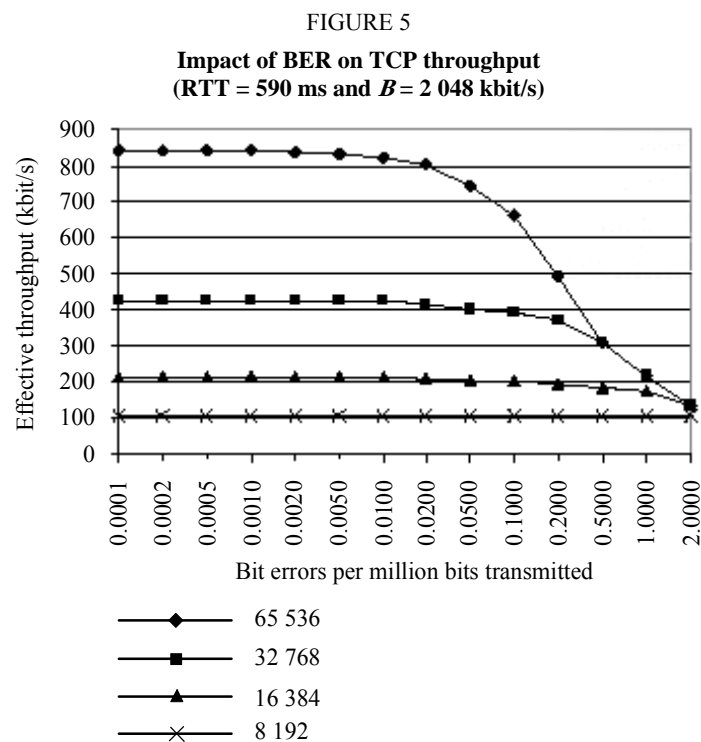
Approximately one round trip after the missing segment has been retransmitted, its acknowledgement is received (assuming the retransmitted segment was not lost). At this time,

instead of setting the congestion window to one segment and performing slow start, the TCP directly sets the congestion window to the slow start threshold. This is the fast recovery algorithm.

Fast retransmit and recovery mechanisms are also affected by long RTT as those encountered over satellite links. The multiple retransmission of duplicate acknowledgements results in a waste of bandwidth, which is a limited resource in satellite networks.

3.4 Effect of bit errors on TCP throughput

Because TCP has no mechanism to distinguish between congestion loss and transmission errors, TCP performs poorly in the presence of link errors and is more sensitive to these errors for larger window sizes (see Fig. 5). In order to achieve a larger throughput using TCP, the link should not experience any losses hence it should have a low BER (see Recommendation ITU-R S.1711).



IETF⁴ Request for Comments 3155⁵ describes the specific TCP mechanisms for environments with high uncorrected error rates such as satellite links. It discusses various methods to mitigate the problem without introducing intermediate devices in the connection. The discussion on TCP enhancement mechanisms includes slow start and congestion avoidance, fast retransmit and fast recovery (see § 3.3), and selective acknowledgments (see Table 3).

3.5 Asymmetric networks

Network asymmetry may degrade TCP performance. Limited available capacity on the reverse channel affects the transmission of acknowledgements to the TCP sender by flooding the reverse channel, which, in turn, limits the forward TCP throughput. Capacity required to carry acknowledgements will further reduce the reverse channel overall capacity and will therefore

⁴ Internet Engineering Task Force.

⁵ RFC 3155 “End-to-end performance implications of links with errors”.

reduce the one available for other TCP connections, especially those using an opposite direction of transmission.

IETF Request for Comments 3449⁶ describes TCP performance degradation in access networks, including bandwidth-asymmetric networks, packet radio and satellite links due to imperfection and variability in the acknowledgement feedback from the receiver to the sender. It details several mitigation techniques:

- to manage the channel used for the upstream bottleneck link carrying the acknowledgements, typically using header compression or reducing the frequency of TCP acknowledgements;
- to handle this reduced acknowledgement frequency to retain the TCP sender's acknowledgment-triggered self-clocking;
- to schedule the data and acknowledgement packets in the reverse direction to improve performance in the presence of two-way traffic.

3.6 Advice to designers

Two IETF⁷ requests for comments provides specific pieces of advice two link and networks designers:

- RFC 3819⁸ provides advice to the designers of digital communication equipment, link-layer protocols, and packet-switched local networks (collectively referred to as sub networks), to support the internet protocols. Various design issues described include TCP performance characteristics, quality of service fairness, delay, bandwidth asymmetry, congestion control, compression and security;
- RFC 3366⁹ provides advice to the designers of digital communication equipment and link-layer protocols employing link-layer automatic repeat request (ARQ) techniques. It provides discussion on various design choices and performance and efficiency improvement methodology for internet traffic users. It describes ARQ over a wide range of physical media, including cellular wireless, wireless LANs, radio links, and other types of channel. It also describes issues relevant to supporting IP traffic over physical-layer channels with varying performance, and where link ARQ is applicable.

4 TCP enhancement methodologies

4.1 Variations of baseline TCP

Several variations of TCP or TCP enhancements may be employed to mitigate the specific impairments of satellite links. The IETF proposed a number of enhancements documented in the RFC. Table 3 lists the TCP enhancements, their corresponding RFC numbers and abstracts describing the content of the RFC document(s). Table 3 also indicates what impairments caused by the satellite link (e.g. latency, large bandwidth delay product (BDP) or high BER) the enhancement can aid.

⁶ RFC 3449 “TCP Performance implications of network path asymmetry”.

⁷ Internet Engineering Task Force.

⁸ RFC 3819 “Advice for internet sub network designers”.

⁹ RFC 3366 “Advice to link designers on link automatic repeat request (ARQ)”.

TABLE 3
TCP enhancements

TCP enhancement	RFC number	TCP impairments over satellite links			Abstract
		Latency	Large BDP	Link errors	
Large initial window	3390 2581	Yes	Yes	No	<p>RFC 3390 “Increasing TCP’s initial window” (2002) It specifies an increase in the permitted initial window for TCP from one or two segment(s) to roughly 4 kbyte/s. It also discusses the advantages and disadvantages of such a change. It replaces RFC 2414.</p> <p>RFC 2581 “TCP congestion control” (1999) It defines the four intertwined congestion control algorithms: slow start, congestion avoidance, fast retransmit, and fast recovery. It additionally specifies how TCP should begin transmission after a relatively long idle period and discusses various acknowledgment generation methods.</p>
Byte counting	3390	Yes	No	No	<p>RFC 3390 “Increasing TCP’s initial window” (2002) Byte counting mechanism increases the congestion window based on the number of transmitted byte/s acknowledged by incoming ACK rather than by the number of ACK received. For long-delay paths in particular, this scheme has been shown to reduce the amount of time it takes to reach the optimal congestion window size.</p>
Window scaling	1323	Yes	Yes	No	<p>RFC 1323 “TCP extensions for high performance” (1992) It presents a set of TCP extensions to improve performance over large bandwidth-delay product paths and to provide reliable operation over very high-speed paths. It defines new TCP options for scaled windows and timestamps, which are designed to provide compatible inter-working with TCP not implementing the extensions. The timestamps are used for two distinct mechanisms: RTTM and PAWS.</p>

TABLE 3 (continued)

TCP enhancement	RFC number	TCP impairments over satellite links			Abstract
		Latency	Large BDP	Link errors	
Pacing TCP segments	2760	Yes	Yes	No	RFC 2760 “Ongoing TCP research related to satellites” (2000) RBP is a technique, used in the absence of incoming ACK, where the data sender temporarily paces TCP segments at a given rate to restart the ACK clock. Upon receipt of the first ACK, pacing is discontinued and normal TCP ACK clocking resumes. The pacing rate may either be known from recent traffic estimates (when restarting an idle connection or from recent prior connections), or may be known through external means (perhaps in a point-to-point or point-to-multipoint satellite network where available bandwidth can be assumed to be large). In addition, pacing data during the first RTT of a transfer may allow TCP to make effective use of high bandwidth-delay links even for short transfers. However, in order to pace segments during the first RTT a TCP will have to be using a non-standard initial congestion window and a new mechanism to pace outgoing segments rather than send them back-to-back. Pacing can also be used to reduce bursts in general.
TCP Vegas	N/A	Yes	Yes	No	TCP Vegas uses a modified slow start and a new retransmission mechanism. The modified slow start algorithm tries to find the correct congestion window size without resulting in any loss of segments.
DACK	1122	Yes	No	No	RFC 1122 “Requirements for Internet hosts – communication layers” (1989) Delayed acknowledgements are used by the TCP receiver enabling the acknowledgement of two received segments at a time thereby reducing acknowledgement traffic. However delaying too long may cause a time out and retransmission at the TCP sender side. The receiver should not delay the acknowledgement more than 0.5 s.

TABLE 3 (continued)

TCP enhancement	RFC number	TCP impairments over satellite links			Abstract
		Latency	Large BDP	Link errors	
TCP SACK	2018	Yes	Yes	Yes	<p>RFC 2018 “TCP selective acknowledgement options” (1996)</p> <p>TCP may experience poor performance when multiple packets are lost from one window of data. With the limited information available from cumulative acknowledgments, a TCP sender can only learn about a single lost packet per round trip time. An aggressive sender could choose to retransmit packets early, but such retransmitted segments may have already been successfully received. A SACK mechanism, combined with a selective repeat retransmission policy, can help to overcome these limitations. The receiving TCP sends back a SACK to the sender informing the sender of data that has been received. The sender can then retransmit only the missing data segments.</p>
	2883				<p>RFC 2883 “An extension to the selective acknowledgement (SACK) option for TCP” (2000)</p> <p>It extends RFC 2018 by specifying the use of the SACK option for acknowledging duplicate packets. When duplicate packets are received, the first block of the SACK option field can be used to report the sequence numbers of the packet that triggered the acknowledgement. This extension to the SACK option allows the TCP sender to infer the order of packets received at the receiver, allowing the sender to infer when it has unnecessarily retransmitted a packet. A TCP sender could then use this information for more robust operation in an environment of reordered packets, ACK loss, packet replication, and/or early retransmit time outs.</p>

TABLE 3 (continued)

TCP enhancement	RFC number	TCP impairments over satellite links			Abstract
		Latency	Large BDP	Link errors	
TCP New Reno	3782	Yes	Yes	Yes	<p>RFC 3782 “The New Reno modification to TCP’s fast recovery algorithm” (2004)</p> <p>RFC 2581 introduces the concept of partial acknowledgments (ACK which cover new data, but not all the data outstanding when loss was detected) in the absence of SACK. RFC 3782, replacing RFC 2582, describes a specific algorithm for responding to partial acknowledgments, referred to as New Reno. While the initial basic algorithm had no mechanism to prevent superfluous multiple “Fast Retransmits” that may occur after a timeout, RFC 2582 already contained a “Careful” variant that avoid these needless “Fast Retransmits”. RFC 3782 specifies the “Careful” variant of New Reno’s Fast Retransmit and Fast Recovery algorithms as the basic version of New Reno TCP.</p>
ECN	3168	Yes	Yes	Partly	<p>RFC 3168 “The addition of explicit congestion notification (ECN) to IP” (2001)</p> <p>It specifies the incorporation of ECN to TCP and IP, including ECN’s use of two bits in the IP header, by setting in the routers a congestion experienced bit in the IP header of packets from ECN-capable transports.. It also describes what modifications are needed to TCP to make it ECN-capable. In satellite links, it may help to distinguish the cause of a packet loss: link errors or network congestion. It also addresses issues to IP tunnels, notably IPSEC ones. It replaces RFC 2481.</p>

TABLE 3 (continued)

TCP enhancement	RFC number ⁽¹⁾	TCP impairments over satellite links			Abstract
		Latency	Large BDP	Link errors	
Header compression	2507	No	No	Yes	<p>RFC 2507 “IP header compression” (1999)</p> <p>It describes how to compress multiple IP headers and TCP and UDP headers per hop over point to point links. The methods can be applied to IPv6 base and extension headers, IPv4 headers, TCP and UDP headers, and encapsulated IPv6 and IPv4 headers. Headers of typical UDP or TCP packets can be compressed down to 4-7 byte/s including the 2 byte/s UDP or TCP checksum. This largely removes the negative impact of large IP headers and allows efficient use of bandwidth on low and medium speed links. The compression algorithms are specifically designed to work well over links with nontrivial packet-loss rates.</p>
Path MTU discovery	1191	Yes	Yes	No	<p>RFC 1191 “Path MTU discovery” (1990)</p> <p>It describes a technique for dynamically discovering the MTU of an arbitrary internet path. Path MTU Discovery allows TCP to use the largest possible packet size, without incurring the cost of fragmentation and reassembly. Increasing TCP congestion window is segment based, rather than byte based and therefore, larger segments enable TCP senders to increase the congestion window more rapidly, in terms of byte/s, than smaller segments.</p>
	2488				<p>RFC 2488 “Enhancing TCP over satellite channels using standard mechanisms” (1999)</p> <p>Path MTU Discovery may cause a delay before TCP is able to start sending data. Satellite delays can aggravate this problem. However, in practice, Path MTU Discovery does not consume a large amount of time due to wide support of common MTU values. Additionally, caching MTU values may be able to eliminate discovery time in many instances.</p>

TABLE 3 (end)

TCP enhancement	RFC number	TCP impairments over satellite links			Abstract
		Latency	Large BDP	Link errors	
T/TCP	1644	Yes	Not relevant	Not relevant	<p>RFC 1644 “T/TCP – TCP extensions for transactions. Functional specifications” (1994)</p> <p>This memo specifies T/TCP, an experimental TCP extension for efficient transaction-oriented (request/response) service. This backwards-compatible extension could fill the gap between the current connection-oriented TCP and the datagram-based UDP.</p>
FEC	2488	Not relevant	Not relevant	Yes	<p>RFC 2488 “Enhancing TCP over satellite channels using standard mechanisms” (1999)</p> <p>TCP provides reliable delivery of data across any network path, including network paths containing satellite channels. While TCP works over satellite channels, FEC indirectly allows TCP to more effectively the available channel capacity by correcting link errors prior to TCP layer.</p>

4.2 Segment splitting methodologies

TCP segment splitting is a scheme where an end-to-end network connection is divided into multiple TCP connections or segments. Typically the segments are divided between terrestrial and satellite components. Moreover the TCP connection over the satellite segment can be modified in order to overcome the satellite link impairments.

The segment splitting scheme is generally implemented in a gateway installed before and/or after the satellite modem (depending on the type of splitting). Although one end-to-end TCP connection is not maintained, the end-users can still communicate with each other without being aware of the gateway function since it emulates a single TCP connection.

4.2.1 Two-segment splitting methodology

The two segment splitting technique divides end-to-end TCP connections into two segments. The network topology as well as the protocol stack associated with this method is depicted in Fig. 6. The comparison of the TCP sequence between the standard TCP and the two-segment splitting technique is depicted in Fig. 7.

In both segments, a standard TCP is used for communications between the TCP sender/receiver and the gateway. When the gateway in earth station A receives a data packet from User 1 on segment 1 and forwards it to User 2 on segment 2, it returns an acknowledgement to User 1 regardless of whether the gateway receives an actual acknowledgement from User 2.

FIGURE 6
Two-segment splitting

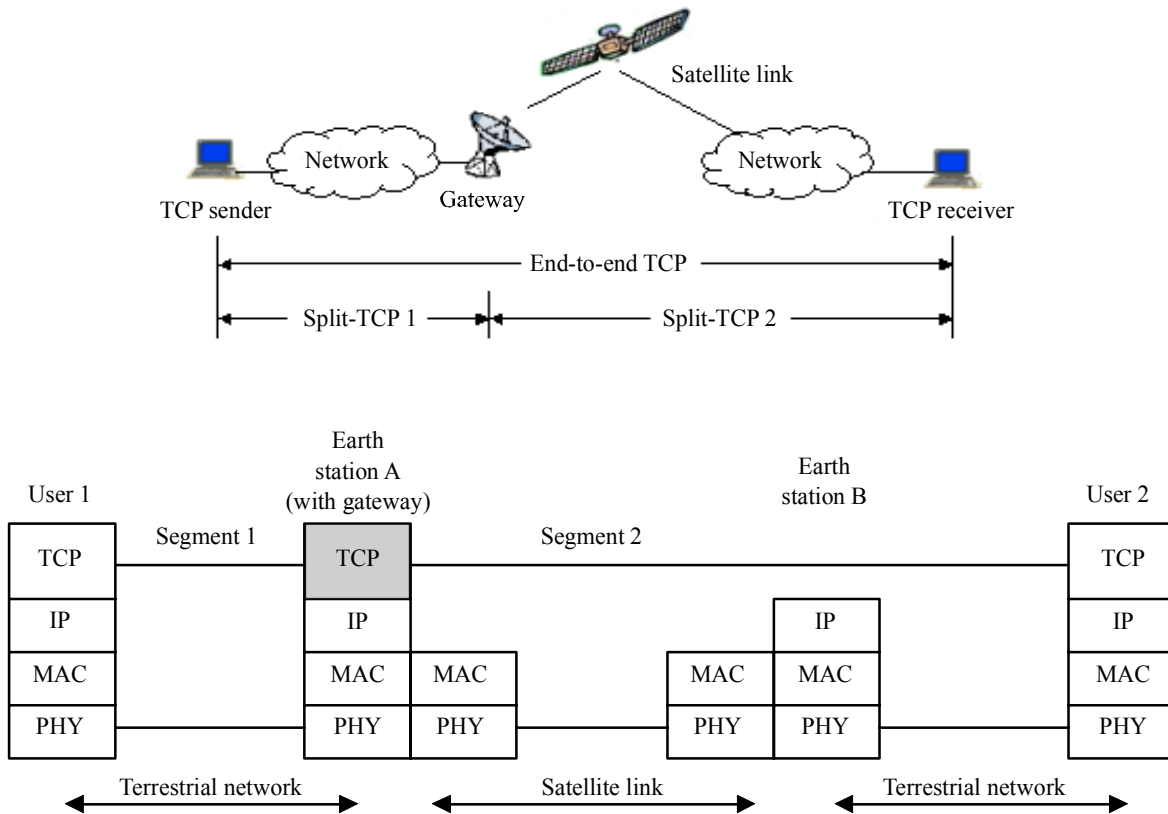
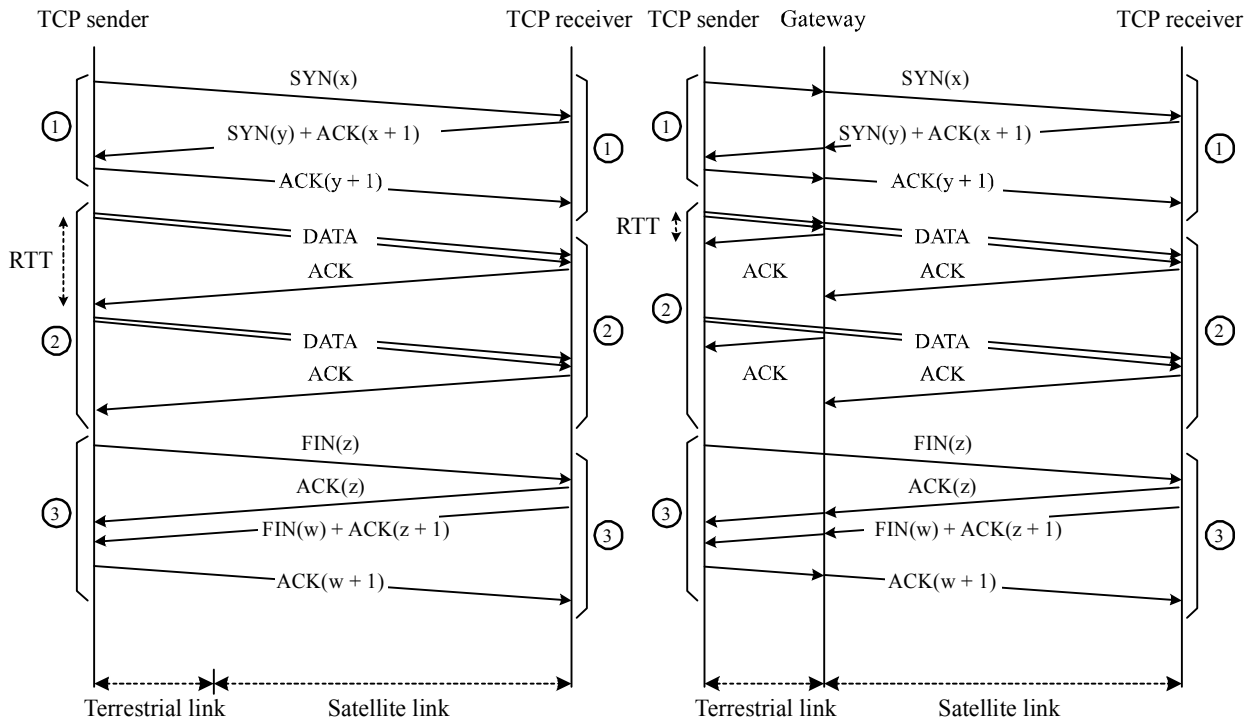


FIGURE 7

Comparison of standard TCP and two-segment splitting technique



a) Communication sequence of standard TCP

b) Communication sequence with two-segment splitting

- ① Connection establishment phase
- ② Data transfer phase
- ③ Connection release phase

Report 2148-07

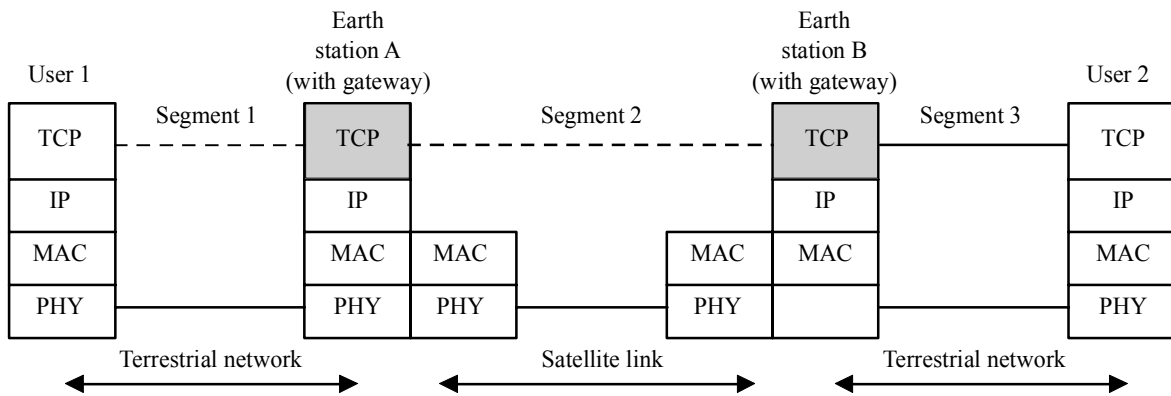
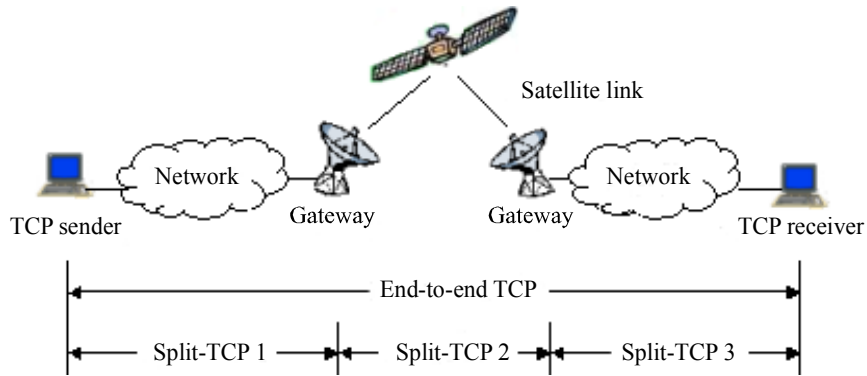
The purpose is to solicit a data transmission from User 1, which enhances the throughput of the end-to-end TCP connection. In this method, throughputs of TCP connections are enhanced only in one direction (in this example from User 1 to User 2). One main advantage of this method is the gateway is only required at the earth station on transmission side (i.e. earth station A). This method is especially suitable for satellite networks with star topologies (see Fig. 2) because the enhancement of the TCP throughput is only needed in one direction (hub to VSAT). In addition, implementing gateway functions to many remote stations is not economically feasible.

4.2.2 Three-segment splitting methodology

The three-segment splitting technique divides end-to-end TCP connections into three segments. Figure 8 shows the network topology as well as the protocol stack for the three segment splitting technique in which terrestrial segments (segments 1 and 3) employ a standard TCP whereas the satellite segment (segment 2) implements an optimized protocol. The TCP sequence of three-segment splitting is shown in Fig. 9. The acknowledgements are generated by the gateway instead of waiting for those of the end-user. In this case, the gateway performs as a proxy.

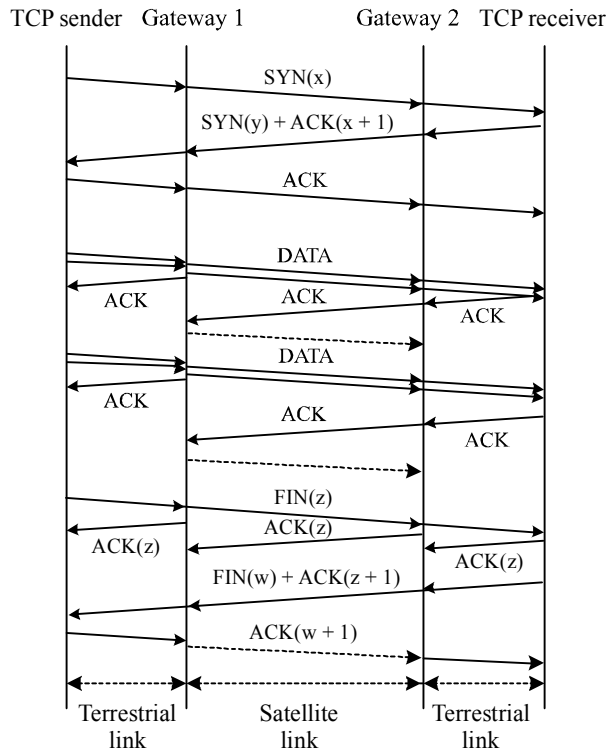
In this method, throughputs of TCP connections are enhanced in both directions. This method is suitable for point-to-point networks such as connection of an ISP to the IP backbone (see Fig. 1).

FIGURE 8
Three-segment splitting



Report 2148-08

FIGURE 9
Three-segment splitting technique



Communication sequence with three-segment splitting

Report 2148-09

4.2.3 Discussion

The adoption of segment-splitting methods is relevant when R_{AB} and R_{BA} in Fig. 1 exceed 256 kbit/s. When the gateway function is activated, the aggregation of TCP throughputs¹⁰ will exceed 70% of the information rate of a satellite link (R_{AB} and R_{BA} in Fig. 1) in both directions under the conditions with a BER of 10^{-8} and a round trip time of 700 ms. The maximum number of enhanced TCP connections depends on the hardware configuration of the gateway (e.g. CPU speed, available RAM).

4.3 Caching and spoofing

4.3.1 Caching

A cache server saves web pages or other files which are frequently used by users. When end-users request access to these web pages or other files, the access request will be connected to the cache server without retrieving the data on the Internet. Therefore caching can reduce the line cost and use efficiently the bandwidth. The cache server is connected directly to the Internet. Usually the saved data in cache server are periodically deleted to protect the cache server memory from data overflow.

4.3.2 Spoofing

In TCP spoofing, a router (gateway) near the source sends acknowledgements for TCP segments to give the source the illusion of a short delay path which speeds up the TCP sender's data transmission. The gateway then suppresses the actual acknowledgement stream from the satellite host and sends any missing data due to link errors or congestions spoofing. This technique is similar to the segment splitting ones described in § 4.2.

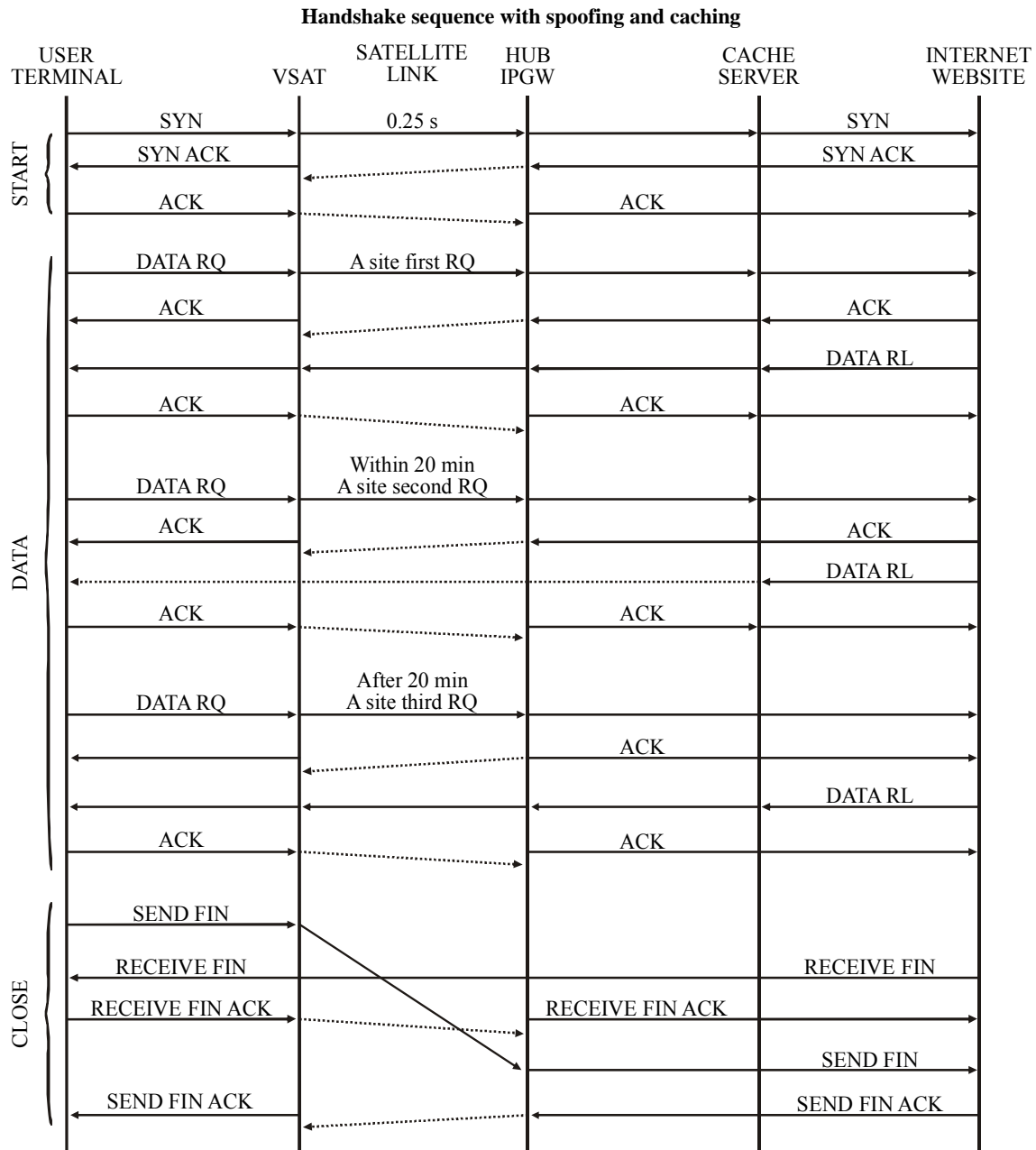
Use of spoofing reduces delay, for example in the case of a geostationary satellite, by about 1.25 s ($250 \text{ ms} \times 5$). The distance between the User Terminal and HUB IPGW corresponds to the satellite link.

4.3.3 Spoofing and caching

Since spoofing is often not effective enough to overcome satellite transmission delays and to improve adequately TCP performance over satellite networks, it is used in conjunction with caching. The satellite network is connected to the selected websites via an IPGW and a cache server. Fig. 10 shows handshake protocol of spoofing plus caching implementation within the satellite network.

¹⁰ The aggregation of TCP throughputs is defined as the sum of instantaneous throughputs of end-to-end TCP connections in the network.

FIGURE 10



Report 2148-10

NOTE 1 – Dashed line denotes an ACK message that is locally generated by the IDU, HUB or Internet website and not transmitted over the satellite link. RQ stands for request and RL stands for release. “SEND FIN” is the FIN flag sent by the TCP sender and “RECEIVE FIN” is the FIN flag sent by the TCP receiver.

5 Performance enhancing proxies

Performance enhancing proxies (PEP) represent a *de facto* solution for TCP over satellite links (see Request for Comments 3135 “Performance enhancing proxies intended to mitigate link-related degradations” (2001)). There are several types of PEPs that can be implemented at any protocol layer. Typically PEPs are implemented at the transport or application layers.

Some PEPs operate at the data link layer but are out of the scope of this Report. Most of transport layer PEPs are designed to interact with TCP and to mitigate the shortages encountered by TCP over satellite links. Such PEPs are transparent for end-to-end application protocols.

PEP implementations can be symmetric or asymmetric, and are sometimes classified depending on their degree of transparency. At one end, PEP implementations are completely transparent to the end systems, transport end points and/or applications, and require no modifications to end systems. In the case of non-transparency, PEP can require modifications to one or both of the end users.

There are two main PEP implementations: TCP spoofing (see § 4.3 and § 5.1) and TCP splitting (see § 4.2 and § 5.2). In both cases the goal is to shield high-latency or lossy satellite network segments from the rest of the network while remaining transparent to the applications.

5.1 TCP spoofing

The principle of TCP spoofing is a router (gateway) near the source sending back acknowledgements for TCP segments in order to give to the source the illusion of a short delay path, which speeds up the TCP sender's data transmission. The gateway then suppresses the actual acknowledgement stream from the satellite host and is also responsible for retransmitting lost data.

TCP spoofing requires a symmetric routing: data segments and associated acknowledgements have to pass through the same paths. Otherwise, the gateway is not be able to manage the TCP connections on the satellite link.

5.2 TCP splitting

The principle of the splitting approach is to isolate the satellite link from terrestrial links in order to have distinct connections. Splitting functionalities are implemented in the satellite gateways, as for TCP spoofing.

Isolating the satellite part of an end-to-end connection enables to use another transport protocol, more fitted to the features of a satellite link, while the end users continue to use the traditional TCP stack for connecting to the satellite gateways.

5.3 Other PEP mechanisms

Other PEP mechanisms include:

- acknowledgement spacing by decreasing the rate at which acknowledgements are sent back to the TCP source, this mechanism prevents their loss in gateway buffers, which would trigger TCP entering into a recovery phase, where TCP throughput is degraded;
- acknowledgement regeneration allows reducing the number of acknowledgements transmitted over the bottleneck link by filtering them at one side of the link and by regenerating the correct sequence of acknowledgements at the opposite side, before sending them to the TCP source;
- local acknowledgements and local retransmissions are performed by the satellite gateway, that stores all the packets of the connections transiting through this gateway. Instead of requesting the TCP source to retransmit lost data, the satellite gateway directly retransmit the stored data, hence decreasing the required time to retransmit;
- tunnelling is a technique where data packets are encapsulated in order to “force” them to go through a specific link. At the end of the “tunnel”, the encapsulation overhead is removed before forwarding the data packets to the end-user;
- header and/or payload compression aims at reducing the required link capacity to transmit data, hence improving the link efficiency and reducing the transmission latency;
- in a link shared between various applications, priority-based multiplexing aims at favouring more “urgent” data transfers, such as those associated to interactive applications, by delaying simultaneous bulk data transfers.

5.4 Implications of using PEP

PEP mechanisms have appeared because, in many operational scenarios, the protocol stack implemented at the end-user terminals can not be changed (e.g. in the most common commercial operating systems, almost no control over the TCP variants is given to the user). In such situations, TCP performance can be optimized only by acting on the satellite part of the IP network. However PEP mechanisms break the end-to-end connection between the IP sender and receiver, which has implications on security and reliability.

5.4.1 End-to-end security

PEP are not able to work with any encrypted transmission such as IPSEC since they need to read IP packet headers and, in some implementations, generate IP packets on behalf of an end system. In general, security mechanisms at or above the transport layer (e.g. TLS or SSL) can be used with PEP.

5.4.2 End-to-end reliability

In architectures involving the use of PEP, applications can not rely on lower level (e.g. TCP) acknowledgements to guarantee reliable end-to-end delivery. TCP PEP generally do not interfere with application layer acknowledgements.

6 Other transmission control protocols

The sources of TCP performance degradation over satellite links and various TCP enhancements have been discussed in the previous sections of this Report. This section introduces other transmission control protocols and compares them with TCP.

6.1 Space communication protocol specification – transport protocol (SCPS-TP)

SCPS is a protocol suite specifically designed to overcome shortfalls occurring in satellite IP-based links and to provide reliable data transfer. SCPS is standardized by the Consultative Committee of space data systems (CCSDS). SCPS-TP modifies TCP in such a way that it uses a congestion control algorithm that does not depend on packet loss as a way to signal congestion in the network. SCPS-TP can react to explicit signals of two sources of packet loss, congestion and link outages.

The ability for SCPS-TP to tailor its response to the nature of the loss allows for better network utilization and better end-to-end performance without harming the overall network stability. Several satellite tests were conducted where losses were not caused by congestion and found SCPS-TP's throughput remained high by avoiding the congestion control response and by providing enhanced information about data loss via the SCPS-TP selective negative acknowledgment (SNACK) option. Among several experiments conducted to measure and verify the performance of SCPS-TP, were two "bent-pipe" tests and one on-board processing test in which the spacecraft hosted the SCPS software. Results proved that SCPS-TP is well suited to the long delay, potentially high bit-error-rate environment of satellites. Using options such as the Header compression, SNACK and TCP time stamps provides throughput improvements varying according to the link conditions.

6.2 Xpress transport protocol

The Xpress transport protocol (XTP) is a next generation transport protocol specially designed to support high-speed networks and multimedia applications. It was also designed for networks exhibiting conditions such as long latency, high loss rate and asymmetric bandwidth.

XTP was designed to provide a wide range of communication services built on the concept that protocol mechanisms can be combined to produce appropriate paradigms within the same basic framework. A key feature of XTP is its ability to allow the application to select its required type of service (i.e. reliable or not). Error control, flow control and rate control are each configured to meet the communication requirements. Due to its efficient control and error handling algorithms, combined with its ability to operate over IP, XTP is able to provide performance gains even when acting as a transparent replacement for TCP, UDP and other existing networking/transport protocols. Especially in congested networks and over high-speed networks, XTP provides significantly higher throughput than TCP.

XTP supports prioritization of packet processing at both the sender and receiver using pre-emptive priority scheduling. Functional features of XTP also include: reliable multicast, reliable datagram, multilevel priority message scheduling, efficient connection management (requiring only 3 packets), selectable error control based on both positive and negative acknowledgements, flow and rate control (i.e. end-systems or intermediate routers can specify their maximum acceptable bandwidth and burst size) and selective acknowledgement. XTP provides an optional selective retransmission algorithm for loss recovery (i.e. when the receiver detects gaps in the sequence of received packets, it transmits back to the sender a list of missing packets, allowing the sender to retransmit only the lost packets).

6.3 Stream control transmission protocol

The stream control transmission protocol (SCTP)¹¹ was initially designed to transport public switched telephone network (PSTN) signalling messages over IP networks, but is capable of broader applications. SCTP is a reliable transport protocol operating on top of a connection-less packet network (e.g. based on IP). The design of SCTP is based on standard features of TCP, such as window-based congestion control, error detection and retransmission. Its features include:

- acknowledged, error-free, non-duplicated data transfer;
- packet validation;
- path management and data fragmentation to conform to discovered path MTU size;
- sequenced delivery of user messages within multiple streams (with an option for order-of-arrival of individual user messages);
- network-level fault tolerance through supporting of multi-homing at either or both ends of an association;
- congestion avoidance;
- chunk building;
- resistance to flooding and masquerade attacks.

SCTP uses as default some optional features of TCP described in Table 3:

- its fast retransmit algorithm is based on selective acknowledgements;
- during the slow start or congestion avoidance phases, the congestion window is increased through the “byte counting” mechanism.

SCTP can also be viewed as a layer between the TCP-based user application and the connection-less, packet-based IP network. In this case, SCTP permits the reliable transfer of user messages between peer SCTP users. It performs this service within the context of an association between two SCTP endpoints. SCTP also provides connection-oriented service to the end points.

¹¹ See IETF Request for Comments 4960 “Stream control transmission protocol”.

6.4 Comparison between transmission control protocols

Table 4 provides a comparison of the four protocols described in this Report: TCP, SCPS-TP, XTP and SCTP.

TABLE 4
Comparison of transport control protocols

Characteristics	TCP	SCPS-TP	XTP	SCTP
Full-duplex	Bidirectional data transfer	Bidirectional data transfer	Bidirectional data transfer	Bidirectional data transfer
Connection	Connection oriented 3-way handshake	Connectionless Connection-oriented service	Connectionless using datagram service	Connectionless-oriented service
Overhead	High	Moderate	Low	High
Flow control	Sliding window	Same as TCP	Yes	Same as TCP
Congestion control	New Reno Vegas Tahoe	TCP Vegas	Network assisted end-to-end	TCP Vegas

7 Conclusion

The longer latency nature of satellite links may degrade the performance of some specific TCP mechanisms. However, various TCP enhancement methodologies have been designed and proved to overcome this performance degradation in satellite environments:

- a number of optional mechanisms available for baseline TCP improves its performance over satellite links;
- when TCP options cannot be set-up by the end-user, segment splitting, caching/spoofing and a few other methodologies can be implemented in performance enhancing proxies often located at the satellite gateways.

Three alternatives to TCP that perform better over satellite IP links have been developed but are currently less widely implemented.
