

RECOMMENDATION ITU-R S.1711

**Performance enhancements of transmission control protocol
over satellite networks**

(Question ITU-R 263/4)

(2005)

Scope

Most of the current IP transmissions use transmission control protocol (TCP) as transport protocol. However the TCP protocol presents some short comings when used in satellite networks. Therefore various techniques, collectively referred to as “TCP performance enhancements”, were developed in order to overcome these limitations. This Recommendation presents an overview of these techniques, briefly describing them and indicating the areas where they improve the performance of TCP over satellite networks. Test results and measurements are also provided in order to get a more accurate view of the effectiveness of some of these techniques.

The ITU Radiocommunication Assembly,

considering

- a) that fixed-satellite systems are being used increasingly for Internet Protocol (IP) packet transmissions, in particular providing broadband applications directly to users in addition to their role as backbone links;
- b) that transmission of IP packets on satellite links requires performance objectives different from those contained in Recommendation ITU-T G.826 and Recommendations ITU-R S.1062 and ITU-R S.1420;
- c) that the performance of transmit control protocol (TCP) may suffer from degradation due to long satellite transmission delay, which affects the quality of service of end-users’ applications;
- d) that the enhancement of TCP performance is therefore critical in designing satellite links to carry IP packets;
- e) that radio frequency resources are not efficiently utilized without implementing the performance enhancement of TCP over satellite in some network environments,

noting

- a) that enhancing the performance of TCP may not be required for low throughput links (see Annex 1),

recommends

- 1 that the reference models, set out in Annex 1 of this Recommendation, should be considered as a basis when developing methods to enhance TCP performance over satellite links;
- 2 that system designers use Annex 2 as guidelines when implementing TCP connections in networks including satellite links.

NOTE 1 – Annex 3 contains a set of tests and measurements that were carried out to assess the effectiveness of several of the methods described in Annex 2 and provides valuable information for satellite system designers.

CONTENTS

	<i>Page</i>
Annex 1 – Satellite system reference models	4
1 Scope	4
2 Reference models	4
2.1 Point-to-point links	6
2.2 VSAT networks	7
2.2.1 Star topology	7
2.2.2 Mesh topology.....	7
2.3 Broadband access.....	8
3 TCP limitations over satellite links	8
3.1 BDP.....	8
3.2 Slow start and congestion avoidance.....	8
3.3 Fast retransmit and fast recovery	10
3.4 Effect of bit errors on TCP throughput.....	10
Annex 2 – TCP enhancement methodologies	11
1 Scope	11
2 Variations of baseline TCP	11
3 Segment splitting methodologies	17
3.1 Two-segment splitting methodology	17
3.2 Three-segment splitting methodology	19
3.3 Discussion.....	20
4 Performance enhancing proxies	20
4.1 TCP spoofing	21
4.2 PEP mechanisms.....	21
4.3 Implications of using PEP	21
4.3.1 End-to-end security	21
4.3.2 End-to-end reliability	21
Annex 3 – Tests and measured performance of TCP enhancements.....	21
1 Scope	21
2 TCP performance with splitting enhancement	22
2.1 Single TCP connection performance tested without any improvement	22
2.1.1 Single TCP connection performance testing.....	22
2.1.2 TCP without performance enhancement.....	22
2.1.3 Performance of single TCP connection without performance enhancement.....	22

2.2	Two-segment splitting	23
2.2.1	Configuration of satellite networks.....	23
2.2.2	Tests results	25
2.2.3	Conclusions	31
2.3	Three-segment splitting	31
2.3.1	Configuration of satellite networks.....	31
2.3.2	Test procedures	31
2.3.3	Test results	32
2.3.4	Conclusions.....	36
3	TCP performance over a satellite ATM network	36
3.1	Network configuration.....	37
3.2	TCP performance in ATM-based satellite-only network	38
3.3	TCP behaviours in heterogeneous networks including satellite link.....	40
3.3.1	Simulation of buffer size.....	41
3.3.2	TCP performance with traffic control in TCP sender	43
3.4	Conclusions.....	45
4	TCP performance in satellite access networks	45
4.1	Network architecture and configuration	45
4.1.1	Network architecture.....	45
4.1.2	Configuration	46
4.2	Results of performance measurement.....	48
4.2.1	Availability.....	49
4.2.2	Throughput.....	53
4.2.3	Traffic.....	55
5	Application protocol measurements (FTP and HTTP).....	57
5.1	Satellite ATM network configuration.....	57
5.2	FTP throughput over OC-3 satellite link	58
5.3	HTTP throughput over OC-3 satellite link	60
5.3.1	HTTP 1.0 with non-persistent connections.....	60
5.3.2	HTTP 1.0 with “keep-alive” option	60
5.3.3	HTTP 1.1 without pipelining	60
5.3.4	HTTP 1.1 with pipelining	61
5.3.5	Test results	61
5.4	Conclusions.....	64

Annex 1

Satellite system reference models

1 Scope

This Annex presents reference models of networks including a satellite link, to carry IP packets, followed by a description of the limitations of TCP over satellite links.

2 Reference models

List of acronyms

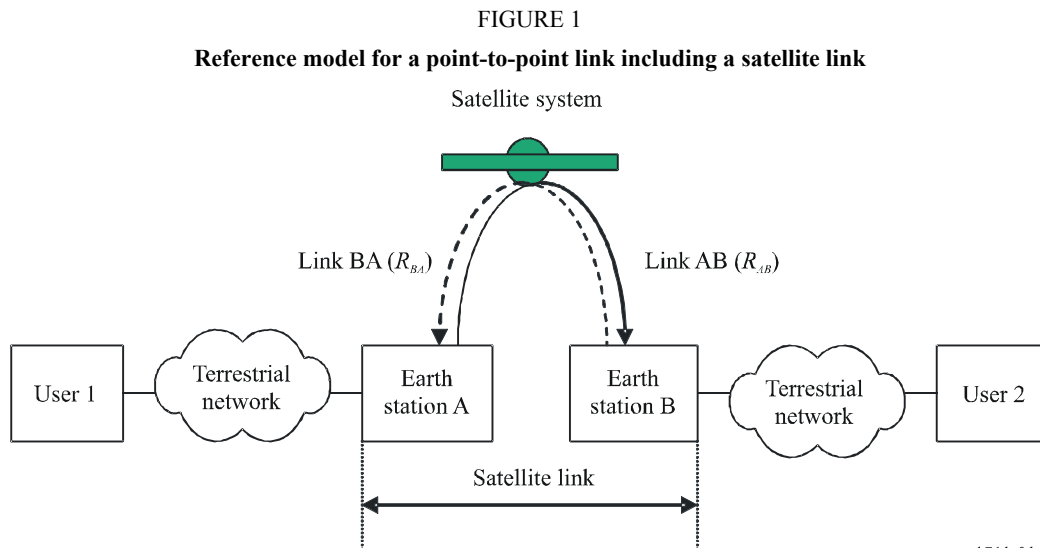
AAL	ATM adaptation layer
ACK	Acknowledgement
ATM	Asynchronous transfer mode
BDP	Bandwidth delay product
BER	Bit error ratio
BW	Bandwidth
CE	Congestion experience bit
CPU	Central processing unit
cwnd	Congestion window (variable in TCP)
DA	Dedicated access
DACK	Delayed acknowledgement
DAMA	Demand assignment multiple access
DVB-S	Digital video broadcast via satellite
ECN	Explicit congestion notification
EIRP	Equivalent isotropic radiated power
FEC	Forward error correction
FIN	Final segment (in a TCP connection)
FTP	File transfer protocol
G/T	Gain to equivalent system temperature ratio
GSO	Geostationary satellite orbit
GW	Gateway
HEO	Highly elliptical orbit
HPA	Hub power amplifier
HSP	Hub satellite processor
HTML	Hypertext markup language
HTTP	Hypertext transfer protocol

ICMP	Internet control message protocol
IETF	Internet engineering task force
I/O	Input/output
IP	Internet protocol
IPSEC	IP security protocol
ISP	Internet service provider
LAN	Local area network
LEO	Low earth orbit
LFN	Long fat network
MEO	Medium earth orbit
MF-TDMA	Multi-frequency time division multiple access
MPEG	Moving picture experts group
MPLS	Multiprotocol label switching
MSS	Maximum segment size
MTU	Maximum transmission unit
NNTP	Network news transport protocol
NTP	Network time protocol
OS	Operating system
PAD	Padding bytes
PAWS	Protect against wrapped sequence(s)
PC	Personal computer(s)
PDU	Protocol data unit
PEP	Performance enhancing proxy
RA	Random access
RAM	Random access memory
RBP	Rate-based pacing
rcvwnd	Receive window (variable in TCP)
RFC	Request for comments (issued by the IETF)
RPA	Remote page accelerator
RS	Reed-Solomon code
RTT	Round trip time
RTTM	RTT measurement
Rx	Receiver
SACK	Selective acknowledgment
SCPC	Single channel per carrier
SSPA	Solid state power amplifier

ssthres	Slow start threshold (variable in TCP)
SYN	Synchronous start segment (used to establish a TCP connection)
T/TCP	TCP for transactions
TBF	Token buffer filter
TCP	Transfer control protocol
TDMA	Time division multiple access
TWTA	Travelling wave tube amplifier
Tx	Transmitter
UDP	User datagram protocol
URL	Uniform/universal resource locator
VSAT	Very small aperture terminal
WAN	Wide area network

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. ATM, frame relay, MPLS).



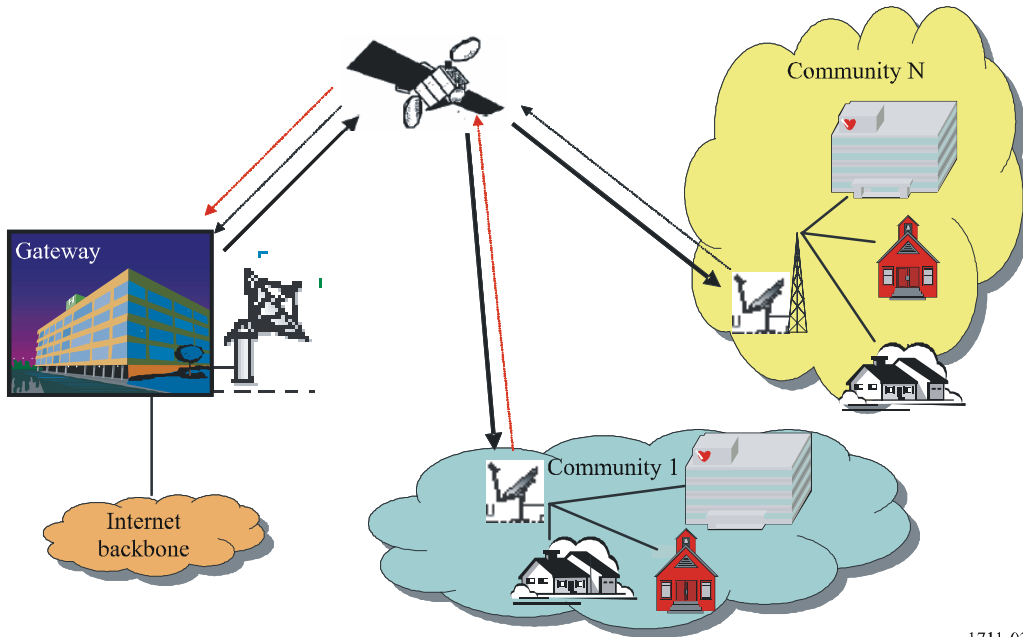
NOTE 1 – The reference model above considers only one satellite hop. Throughout this Recommendation, the techniques that segment the TCP connection to improve TCP performance over satellite links are described for one satellite hop. However an end-to-end connection may include several satellite hops. 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

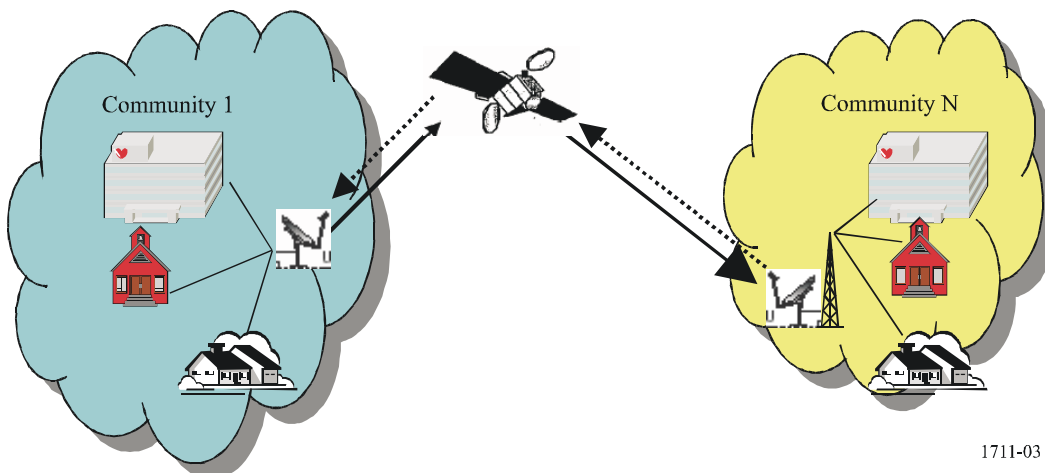


1711-02

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



1711-03

2.3 Broadband access

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

3 TCP limitations over satellite links

The TCP protocol 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 protocol (TCP Reno) specifies slow start, congestion avoidance, fast retransmit and fast recovery for congestion control. The TCP protocol 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 bytes, the amount of unacknowledged data that the sender can have in transit to the receiver.

3.1 BDP

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 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 bytes 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 kbytes and satellite links with variable RTT, the maximum throughput is as follows:

TABLE 1

Maximum throughput according to RTT values

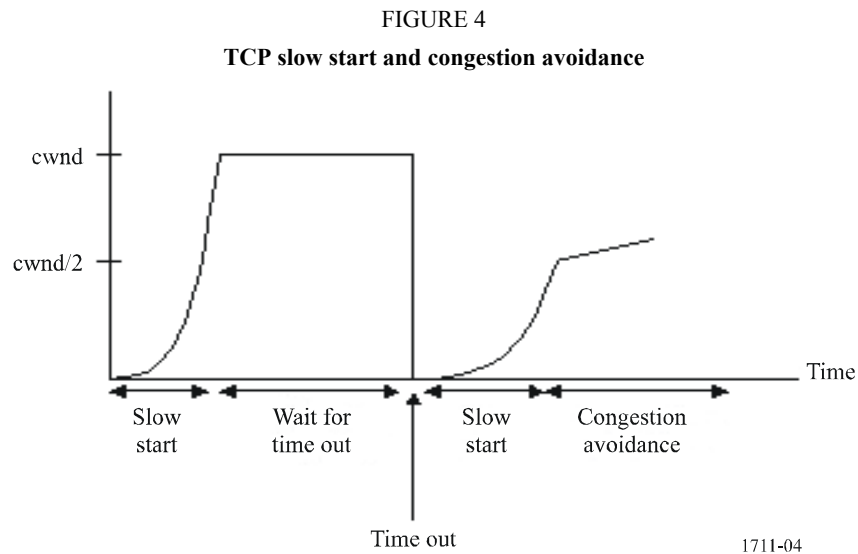
Satellite network type	RTT (ms)	Maximum throughput (kbyte/s)
LEO	~20	~3 200
MEO	~200	~320
HEO	~600	~110
GSO	~520	~120

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

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 kbytes. The sender maintains a retransmission timeout for the last unacknowledged packet. Congestion is detected by the expiration of the retransmission timeout. 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 acknowledgement 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 the Figure, cwnd stands for congestion window).



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

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

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.

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:

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

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 $1/(B \cdot RTT)$. 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 timeout. As a result, during congestion, the TCP connection loses time waiting for the timeout. In Fig. 1, the horizontal line (at the cwnd value) shows the time lost when waiting for a timeout to occur. During this time, TCP neither sends new packets nor retransmits lost packets. Moreover, once the timeout 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.

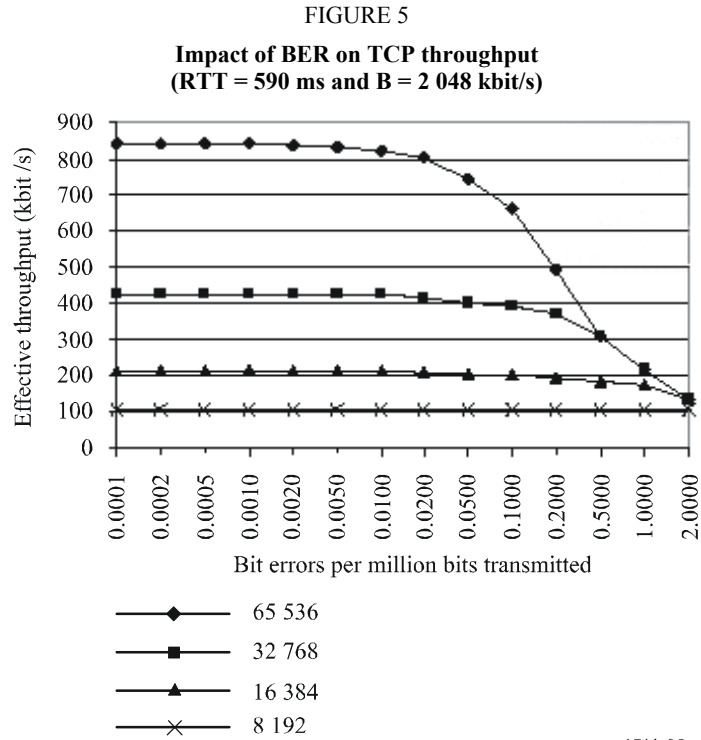
If the network drops a segment, the subsequent segments arriving at the receiver are out-of-order. For each of them, the TCP receiver immediately sends an acknowledgement to the sender indicating the sequence number of the missing segment. This acknowledgement is called a duplicate acknowledgement. When 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

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.



Annex 2

TCP enhancement methodologies

1 Scope

There are various enhancements to the baseline TCP that can be implemented in order to mitigate the degradation of TCP connections in networks such as those modelled in Annex 1. This Annex describes them and lists their respective advantages and drawbacks for use in satellite networks.

2 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). The Table also indicates what impairments caused by the satellite link (e.g. latency, large BDP or high BER) the enhancement can aid.

TABLE 3

TCP enhancements

TCP enhancement	RFC number ⁽¹⁾	TCP impairments over satellite links			Abstract
		Latency	Large BDP	Link errors	
Large initial window	2414	Yes	Yes	No	RFC 2414 "Increasing TCP's initial window" (1998) It specifies an increase in the permitted initial window for TCP from one segment to roughly 4 kbits. It also discusses the advantages and disadvantages of such a change.
	2581				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.
Byte counting	2414	Yes	No	No	RFC 2414 "Increasing TCP's initial window" (1998) Byte counting mechanism increases the congestion window based on the number of transmitted bytes 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.
Window scaling	1323	Yes	Yes	No	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.

TABLE 3 (cont.)

TCP enhancement	RFC number ⁽¹⁾	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 timeout and retransmission at the TCP sender side. The receiver should not delay the acknowledgement more than 0.5 s.

TABLE 3 (cont.)

TCP enhancement	RFC number ⁽¹⁾	TCP impairments over satellite links			Abstract
		Latency	Large BDP	Link errors	
ECN	2481	Yes	Yes	Partly	<p>RFC 2481 “A proposal to add Explicit Congestion Notification (ECN) to IP” (1999)</p> <p>It describes a proposed addition of ECN to IP by setting in the routers a congestion experienced bit. It also describes what modifications would be 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.</p>
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 bytes including the 2 bytes 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 bytes, than smaller segments.</p>

TABLE 3 (end)

TCP enhancement	RFC number ⁽¹⁾	TCP impairments over satellite links			Abstract
		Latency	Large BDP	Link errors	
Path MTU discovery (cont.)	2488	Yes	Yes	No	RFC 2488 “Enhancing TCP over satellite channels using standard mechanisms” (1999) 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.
T/TCP	1644	Yes	Not relevant	Not relevant	RFC 1644 “T/TCP – TCP extensions for transactions. Functional specifications” (1994) 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.
FEC	2488	Not relevant	Not relevant	Yes	RFC 2488 “Enhancing TCP over satellite channels using standard mechanisms” (1999) 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.

⁽¹⁾ RFC stands for Request For Comments and are documents issued by the Internet Engineering Task Force (IETF).

3 Segment splitting methodologies

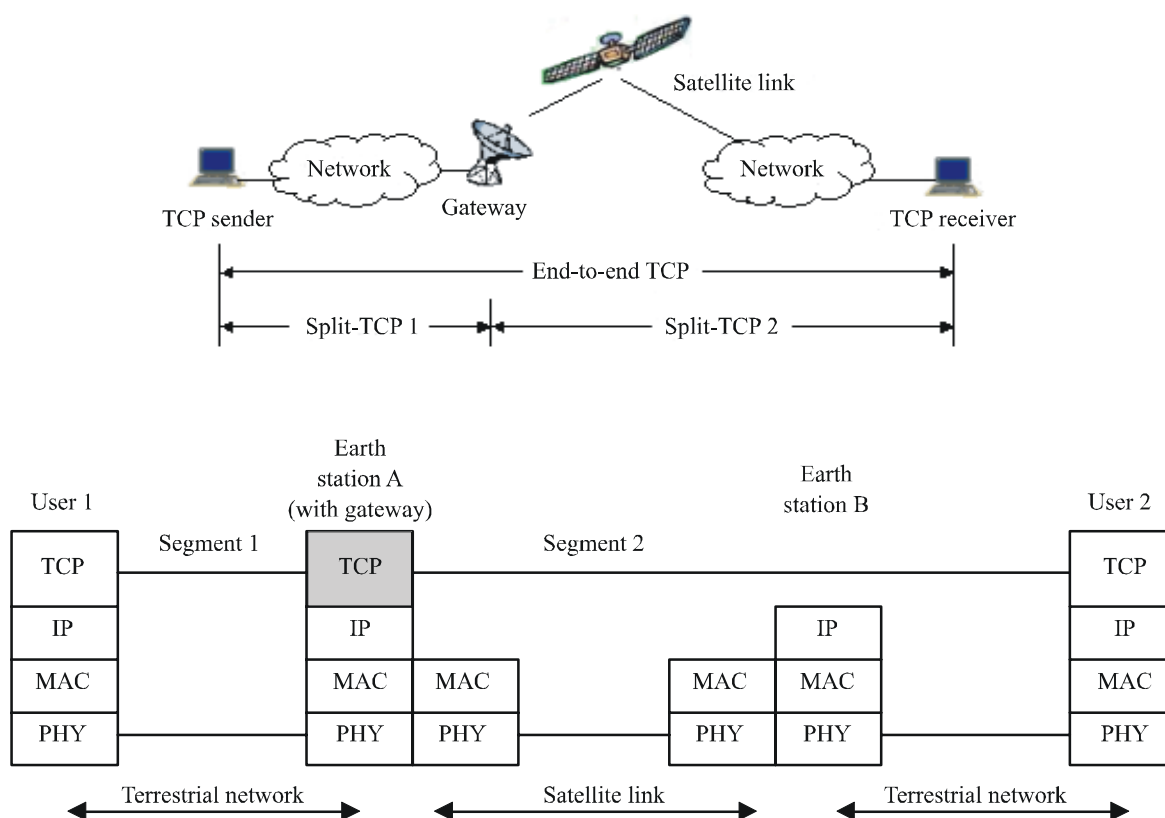
TCP segment splitting is a scheme where an end-to-end network connection is divided into multiple TCP connections or segments. Typically the segments are divided between terrestrial and satellite components. Moreover the TCP connection over the satellite segment can be modified in order to overcome the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

3.1 Two-segment splitting methodology

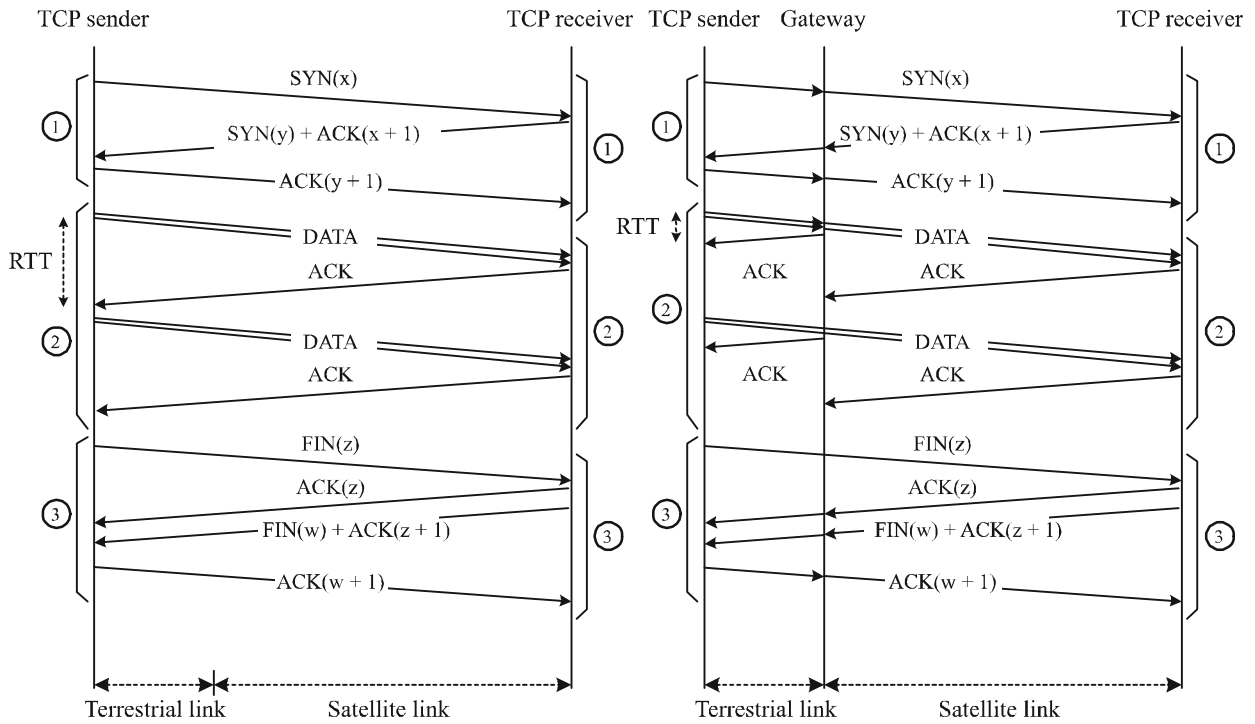
The two segment splitting technique divides end-to-end TCP connections into two segments. The network topology as well as the protocol stack associated with this method is depicted in Fig. 6. The comparison of the TCP sequence between the standard TCP and the two-segment splitting technique is depicted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

FIGURE 6
Two-segment splitting



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. 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.

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

3.2 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

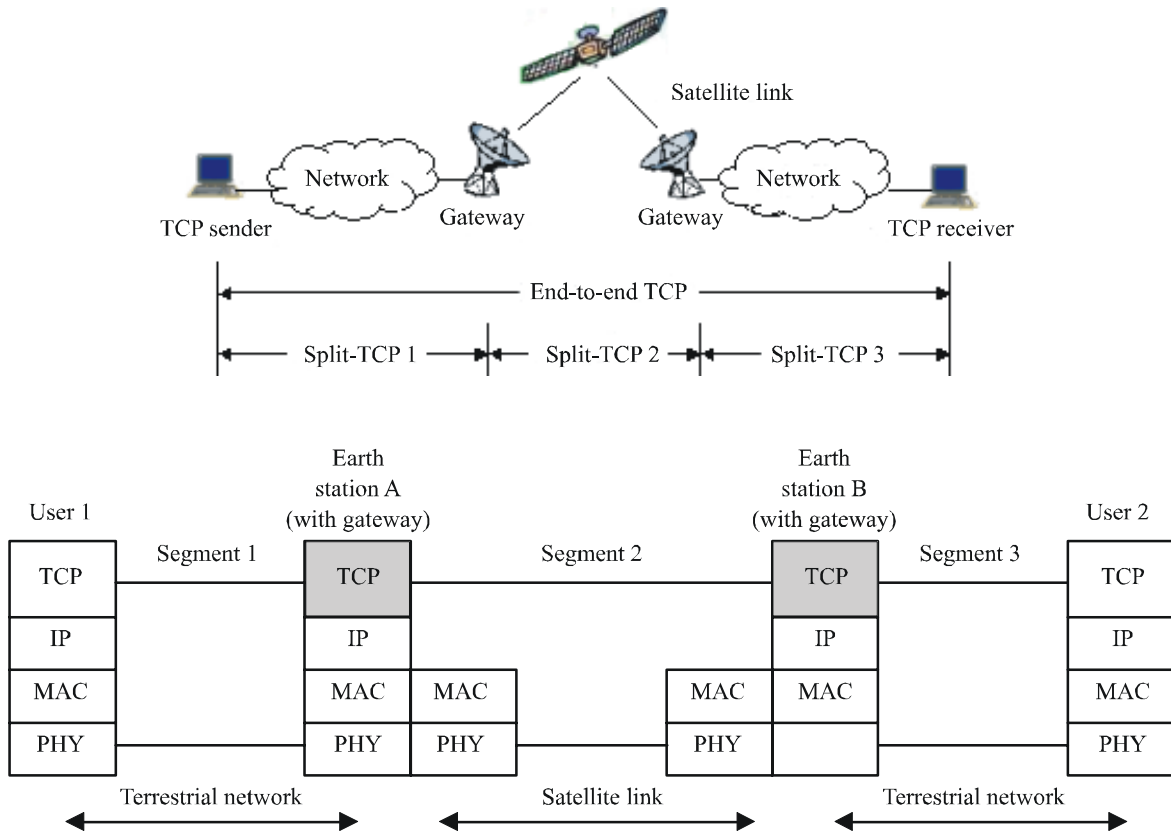
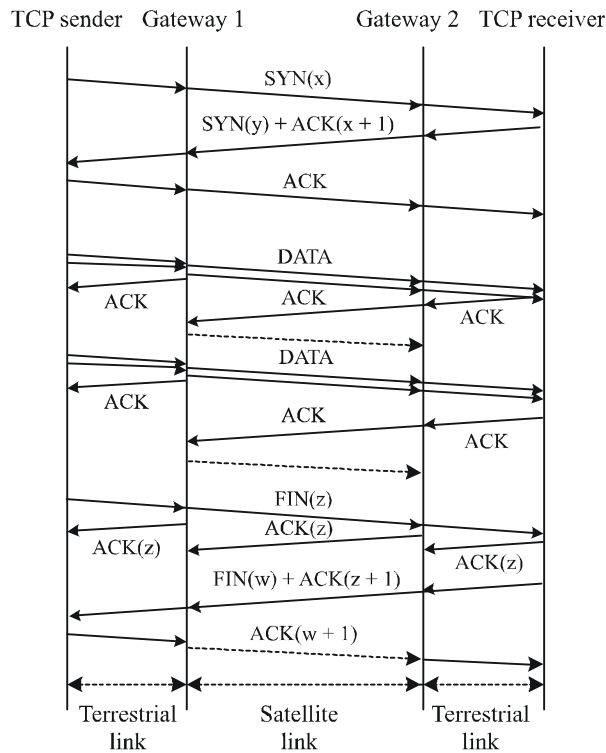


FIGURE 9

Three-segment splitting technique

Communication sequence with three-segment splitting

1711-09

3.3 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 throughputs¹ 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 Performance enhancing proxies

PEPs represent a *de facto* solution for TCP over satellite links (see IETF RFC 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 Recommendation. 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.

¹ The aggregation of TCP throughputs is defined as the sum of instantaneous throughputs of end-to-end TCP connections in the network.

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 strategies in PEP design: TCP spoofing and TCP splitting (see § 3). 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.

4.1 TCP spoofing

The principle of TCP spoofing is a router (gateway) near the source sending back acknowledgements for TCP segments in order to give 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.

4.2 PEP mechanisms

PEP mechanisms include acknowledgement spacing, acknowledgement regeneration, local acknowledgements, local retransmissions, tunnels to control routing of packets, header compression, payload compression and priority-based multiplexing.

4.3 Implications of using PEP

4.3.1 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.

4.3.2 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.

Annex 3

Tests and measured performance of TCP enhancements

1 Scope

Annex 3 presents the results of independent tests and measurements conducted to experiment some of the methods described in Annex 2 and provides valuable information for satellite system designers.

2 TCP performance with splitting enhancement

INTELSAT and KDDI have carried out measurement of TCP performance using segment splitting techniques (two-segment splitting and three-segment splitting) to verify the effectiveness of these techniques. This section provides the results of these measurements.

Section 2.1 presents the performance test results of a single TCP connection without any gateway. Section 2.2 contains the test results of the two-segment splitting technique. Section 2.3 gives the test results of the three-segment splitting technique.

2.1 Single TCP connection performance tested without any improvement

2.1.1 Single TCP connection performance testing

The first set of tests examined the throughput of a single TCP connection. Tests were run with and without protocol gateway enhancement for round trip delays of 200 ms, simulating a terrestrial WAN connection, and 700 ms, simulating a combined satellite link into the terrestrial backbone.

2.1.2 TCP without performance enhancement

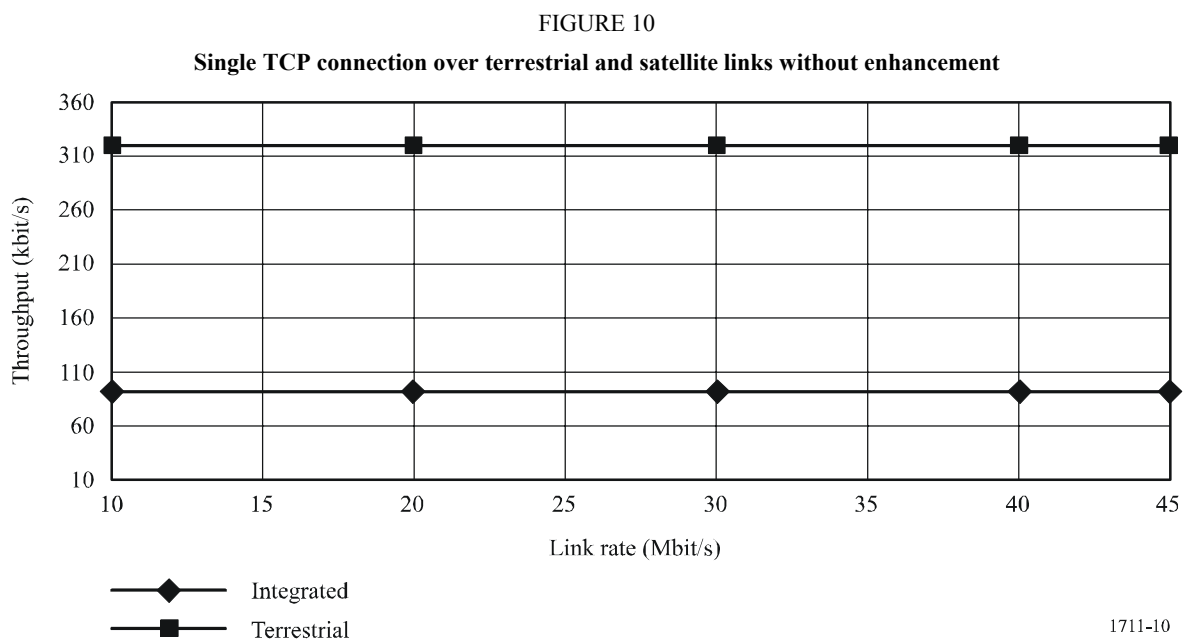
The first set of tests was for single TCP connections without performance enhancement. The client window size was set to 8 kbytes to match the default settings on Windows 95, Windows 98, Windows NT, and many other common operating systems. The terrestrial link RTT delay was set to 200 ms and the combined satellite/terrestrial link to 700 ms.

2.1.3 Performance of single TCP connection without performance enhancement

The maximum throughput without performance enhancement is 320 kbit/s for the terrestrial connections and 91 kbit/s for the satellite link (see Fig. 10). These results demonstrate that without performance enhancement, the maximum single-connection TCP throughput rate will be approximately equal to the window size,

$$\frac{8 \text{ kbytes} \cdot 8 \text{ bits}}{RTT = 200 \text{ ms}} = 320 \text{ kbit/s}$$

even if the link rate is increased.



2.2 Two-segment splitting

2.2.1 Configuration of satellite networks

In order to verify the effectiveness of two-segment splitting techniques under various conditions, the following tests were conducted:

TEST-A: Field measurement using a real VSAT system over a satellite link

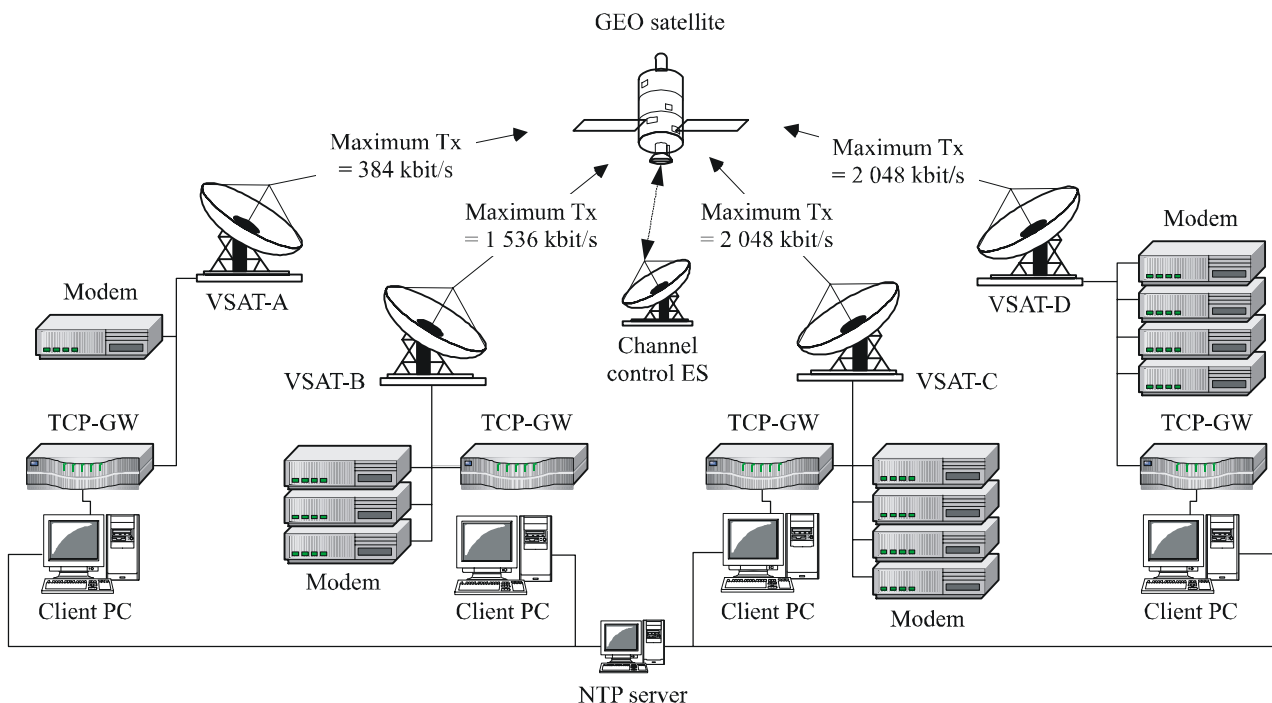
TEST-B: Measurement using a satellite link simulator

2.2.1.1 TEST-A: Field measurement using a real VSAT system over a satellite link

The TEST-A measurement was carried out using a DAMA-based satellite IP network with a mesh topology employing variable rate SCPC technology. The test consisted of throughput measurement of single TCP connection and multiple TCP connection for various link rates, UDP and mixture of UDP and TCP connections. The test was conducted using both an asymmetric and a symmetric satellite network whose link rates are 384 kbit/s, 1 536 kbit/s and 2 048 kbit/s.

The test network in TEST-A is shown in Fig. 11. The DAMA satellite network consists of an earth station for channel control and four VSAT. Each VSAT is equipped with gateway equipment enabled for two-segment splitting technique to enhance the throughput of the forward direction. The major specifications of each VSAT are shown in Table 4. Note that there are two types of gateway equipment (Type 1 and Type 2) from two different vendors were used in this test. The satellite link is designed with link availabilities of 99.9% between VSAT and HUB, and 99.85% between VSAT and VSAT.

FIGURE 11
Test network configuration for TEST-A



ES: earth station
TCP-GW: TCP gateway with 2-segment splitting technique

TABLE 4
Specifications of VSAT earth station

Earth station	Antenna size (m)	SSPA power (w)	Maximum Tx rate (kbit/s)	Provider of TCP gateway
Control E/S	7.6	N/A	N/A	N/A
VSAT-A	1.2	10	384	Type-1
VSAT-B	1.2	40	1 536	Type-2
VSAT-C	1.8	40	2 048	Type-2
VSAT-D	1.8	120	2 048	Type-2

Earth station	OS	CPU	Memory (Mbit)	Application
VSAT-A	FreeBSD 4.3	Pentium III 1 GHz	256	Iperf 1.1.1
VSAT-B	FreeBSD 4.3	Pentium III 1 GHz	512	Iperf 1.1.1
VSAT-C	FreeBSD 4.3	Pentium III 1 GHz	256	Iperf 1.1.1
VSAT-D	FreeBSD 4.3	Pentium III 1 GHz	512	Iperf 1.1.1
NTP server	Windows 2000	Pentium III 600 MHz	256	

The FreeBSD 4.5 operating system is installed on all client PCs, which are then individually connected to a NTP server in order to synchronize the timings between PCs. Each VSAT is equipped with Iperf² software to generate IP packets and measure throughput.

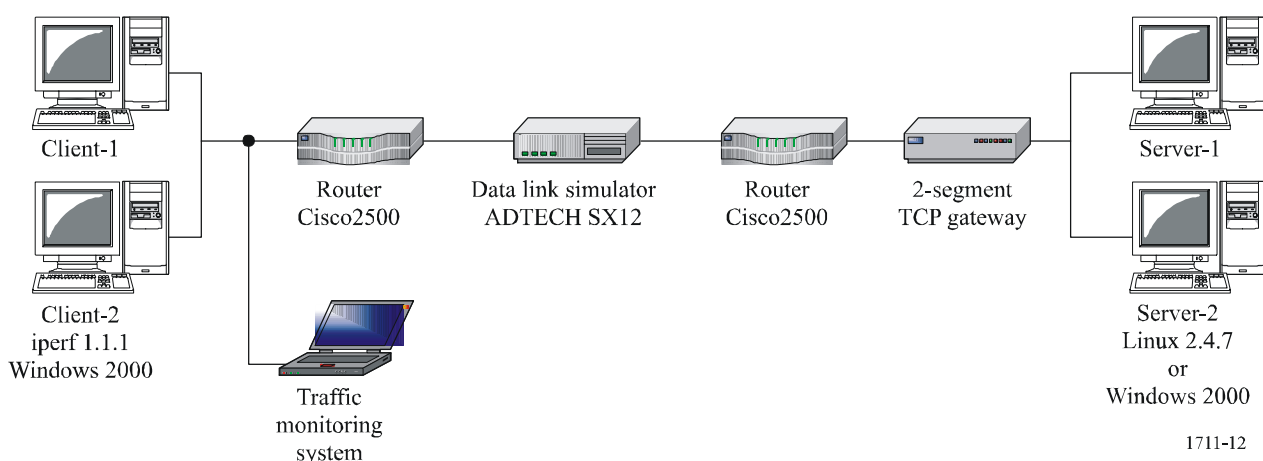
2.2.1.2 TEST-B: measurement using a satellite link simulator

The TEST-B: measurement was carried out with a point-to-point configuration using a data link simulator which can insert delay and channel errors. The test consisted of throughput measurements for 1, 2, 4 and 8 simultaneous TCP connections with various link rates (ranging from 384 kbit/s to 1 536 kbit/s) simulating various bit error conditions (error-free, 10^{-8} , 10^{-7} , 10^{-6} , 10^{-5} ; random error/burst error) and 250 ms of satellite delay (for one way). For the purpose of comparison, the throughputs were measured with and without the use of gateway equipment.

The test network is shown Fig. 12. The network contains the data link simulator to simulate the effect of the satellite link conditions in point-to-point networks. The data link simulator uses a serial interface, with two routers installed on both sides of the data link simulator to adapt the interface. The client PC run the Microsoft Windows 2000 (SP2) operating system, and the server PC runs either Microsoft Windows 2000 (SP2) or Linux Version 2.4.7.

² Iperf is an application to generate traffic and measure the throughput of IP packet. Iperf version 1.1.1 is a free software that can be downloaded at: <http://dast.nlanr.net/Projects/Iperf1.1.1/release.html>.

FIGURE 12
TEST network configuration for TEST-B



1711-12

2.2.2 Tests results

2.2.2.1 TEST-A

The four tests were conducted under clear-sky conditions. The test results are presented in the following sections:

- UDP connection (§ 2.2.2.1.1)
- Single TCP/IP connection (§ 2.2.2.1.2)
- Multiple TCP/IP connection (§ 2.2.2.1.3) (see Note 1)
- One TCP session (60% of link rate) and UDP session (40% of link rate) (§ 2.2.2.1.4) (see Note 2).

NOTE 1 – The number of session is four on one PC. Some VSAT could not establish four sessions of TCP/IP simultaneously.

NOTE 2 – The throughput measured in UDP connection test is assumed to be the maximum throughput of TCP session.

The effect of enhancement by the gateway for two-segment splitting technique was duly observed through all the tests as throughputs of more than 95% of the maximum throughput were obtained.

2.2.2.1.1 Results of UDP connection

TABLE 5

Link rate: 384 kbit/s, average: 360.2 kbit/s (93.8%) (without overhead data)

		Tx (set the link rate to 384 kbit/s)			
		VSAT-A (kbit/s)	VSAT-B (kbit/s)	VSAT-C (kbit/s)	VSAT-D (kbit/s)
Rx (384 kbit/s)	VSAT-A		365	365	365
	VSAT-B	345		365	365
	VSAT-C	346	365		365
	VSAT-D	346	365	365	

TABLE 6

Link rate: 1 536 kbit/s, average: 1 463 kbit/s (95.2%) (without overhead data)

		Tx (set the link rate to 1 536 kbit/s)			
		VSAT-A	VSAT-B	VSAT-C	VSAT-D
Rx (384 kbit/s)	VSAT-A			1 463 kbit/s	

TABLE 7

Link rate: 2 048 kbit/s, average: 1 947.5 kbit/s (95.1%)

		Tx (set the link rate to 2 048 kbit/s)			
		A (384 kbit/s)	B (1 536 kbit/s)	C (2 048 kbit/s)	D (2 048 kbit/s)
Rx (1 536 kbit/s)	VSAT-B			1 946 kbit/s	1 949 kbit/s

2.2.2.1.2 Results of single TCP connection

TABLE 8

Link rate: 384 kbit/s, average: 349.2 kbit/s (96.9%) (without overhead data)

		Tx (set the link rate to 384 kbit/s)			
		VSAT-A (kbit/s)	VSAT-B (kbit/s)	VSAT-C (kbit/s)	VSAT-D (kbit/s)
Rx (384 kbit/s)	VSAT-A		359.0	359.0	359.0
	VSAT-B	327.8		358.5	358.3
	VSAT-C	328.0	348.3		357.8
	VSAT-D	328.0	358.5	348.3	

Table 8 shows the average over four runs.

TABLE 9

Link rate: 1 536 kbit/s, average: 1 397.5 kbit/s (95.5%) (without overhead data)

		TX (set the link rate to 1 536 kbit/s)			
		VSAT-A (384 kbit/s)	VSAT-B (1 536 kbit/s)	VSAT-C (2 048 kbit/s)	VSAT-D (2 048 kbit/s)
Rx (384 kbit/s)	VSAT-A			1 397.5 kbit/s	

Table 9 shows the average over four runs.

TABLE 10

Link rate: 2 048 kbit/s, average: 1 890.1 kbit/s (97.1%) (without overhead data)

		Tx (set the link rate to 2 048 kbit/s)			
		VSAT-A (384 kbit/s)	VSAT-B (1 536 kbit/s)	VSAT-C (2 048 kbit/s)	VSAT-D (2 048 kbit/s)
Rx (1 536 kbit/s)	VSAT-B			1 888.3 kbit/s	1 891.8 kbit/s

2.2.2.1.3 Results of multiple TCP connection

TABLE 11

Link rate: 1536 kbit/s, average: 1 370.5 kbit/s (95.5%) (without overhead data)

		Tx (set the link rate to 1 536 kbit/s)			
		VSAT-A (384 kbit/s)	VSAT-B (1 536 kbit/s)	VSAT-C (2 048 kbit/s) (kbit/s)	VSAT-D (2 048 kbit/s) (kbit/s)
Rx	VSAT-A (384 kbit/s)			360	338
				358	337
				345	329
				345	329
	Total			1 408	1 333

TABLE 12

Link rate: 2 048 kbit/s, average: 1 910 kbit/s (98.1%) (without overhead data)

		Tx (set the link rate to 2 048 kbit/s)			
		VSAT-A (384 kbit/s)	VSAT-B (1 536 kbit/s)	VSAT-C (2 048 kbit/s) (kbit/s)	VSAT-D (2 048 kbit/s) (kbit/s)
Rx	VSAT-B (1 536 kbit/s)			759	680
				597	565
				562	657
	Total			1 918	1 902

2.2.2.1.4 Results of combination of UDP and TCP sessions

TABLE 13

Combination of one TCP (60%) connection and one UDP (40%) connection

		Tx (384 kbit/s)			
		VSAT-A (384 kbit/s)	VSAT-B (1 536 kbit/s)	VSAT-C (2 048 kbit/s) (kbit/s)	VSAT-D (2 048 kbit/s) (kbit/s)
Rx	VSAT-B			1241 687	1102 841

Upper row: TCP connection; lower row: UDP connection.

2.2.2.2 TEST-B

Test results of TEST-B are shown in Figs. 13 to 17. The throughput values in the graph of 1, 2, 8 TCP sessions are sum of throughputs of all TCP sessions. The effect of enhancement by the gateway for two-segment splitting technique was duly observed through all the tests.

As can be seen in Figs. 13 and 14, the maximum throughput for one TCP/IP session is limited to approximately 200 kbit/s when the gateway is not activated. For instance, in the case of two TCP sessions (see Fig. 14), the total throughput (“without TCP GW”) is limited to approximately 400 kbit/s. On the other hand, in the case of eight TCP sessions (see Fig. 15), the total throughput (“without TCP GW”) reaches approximately 1.5 Mbit/s, and no remarkable difference is observed between the cases of “without TCP GW” and “with TCP GW”.

With higher BERs (i.e. $BER = 10^{-6}$), the effect of enhancement is reduced as shown in Fig. 16.

FIGURE 13

Result of one TCP session ($BER = 10^{-8}$)

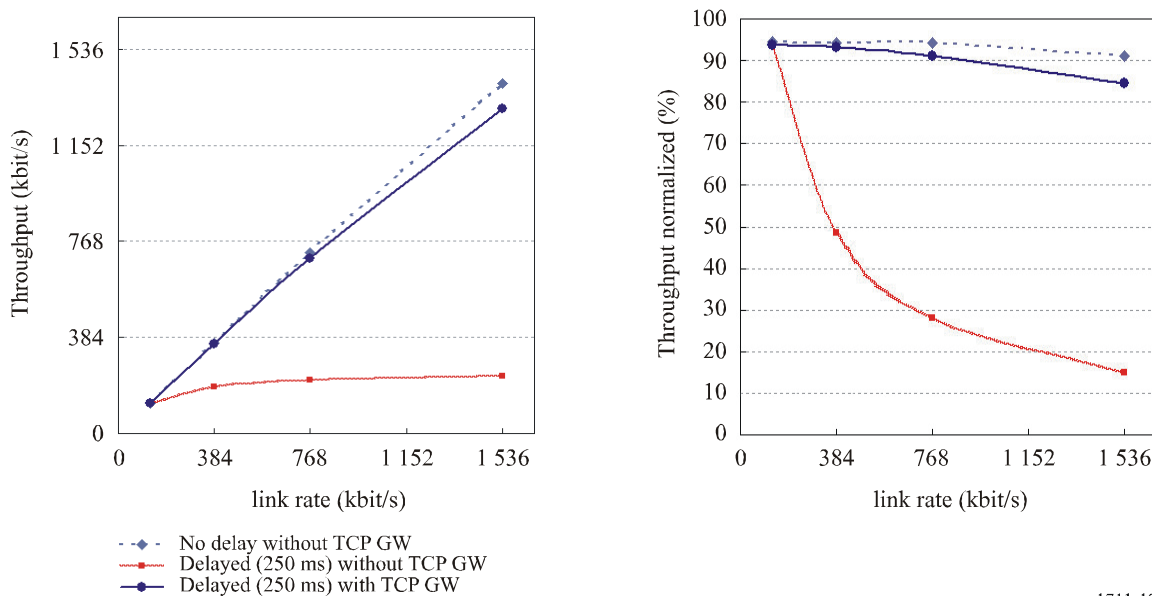
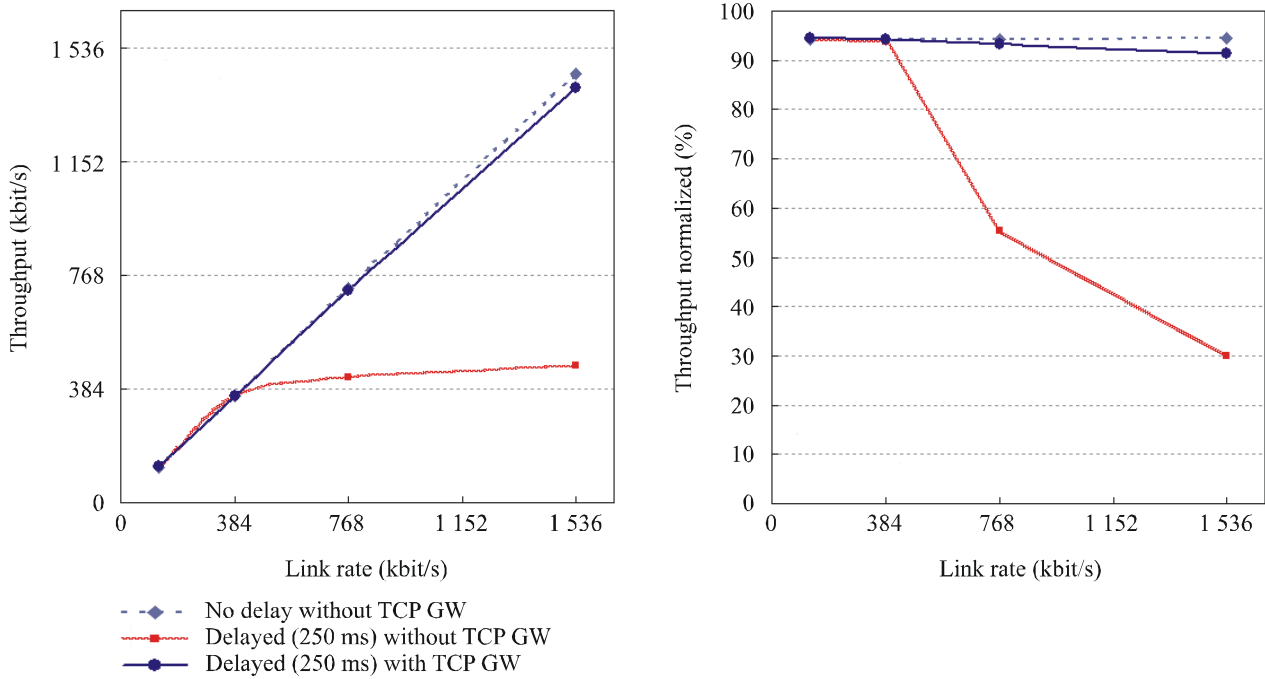
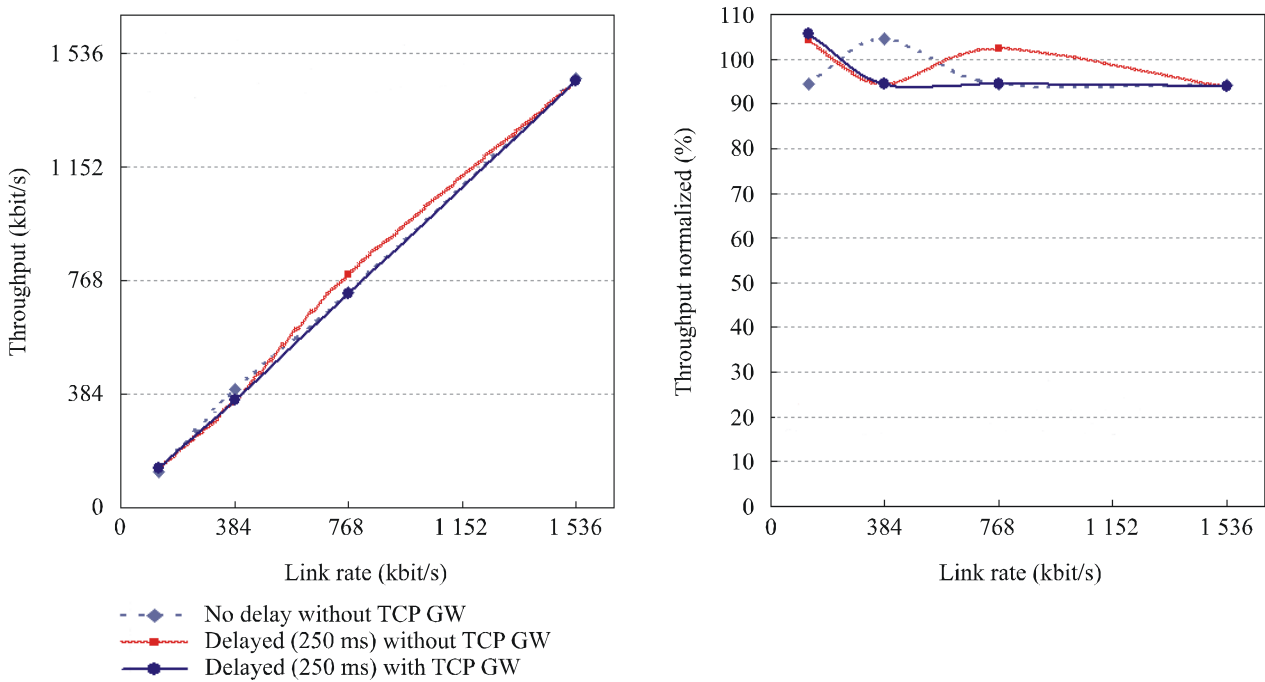


FIGURE 14
Result of two TCP sessions (BER = 10⁻⁸)



1711-14

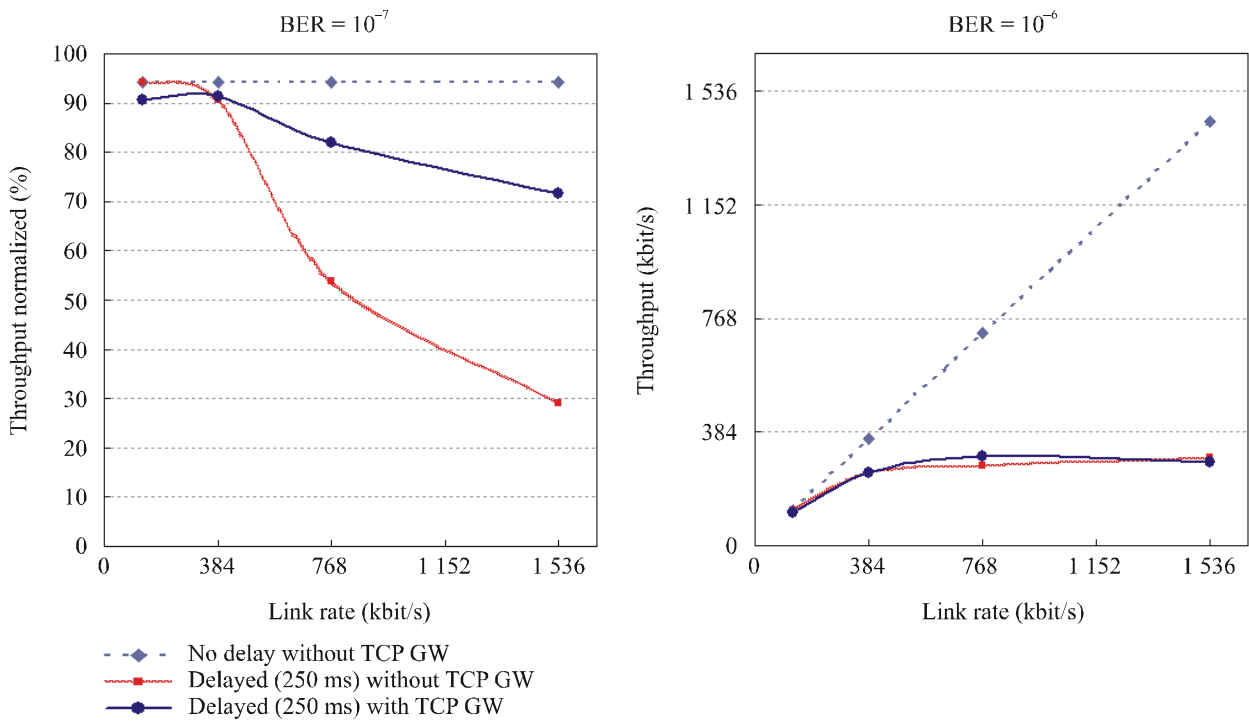
FIGURE 15
Result of eight TCP sessions (BER = 10⁻⁸)



1711-15

FIGURE 16

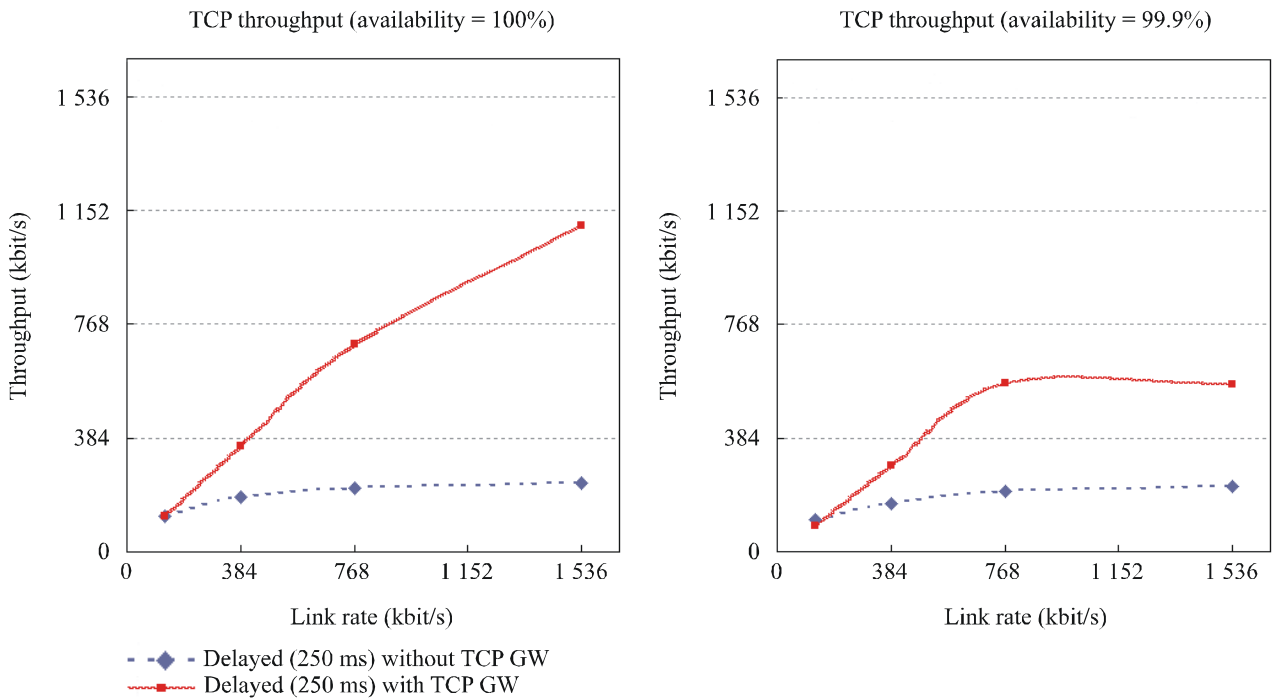
Results of two TCP sessions (BER = 10^{-7} , 10^{-6})



1711-16

FIGURE 17

Results of one TCP session in case of various burst errors



1711-17

2.2.3 Conclusions

The results show that the two-segment splitting method can significantly improve TCP throughput in the satellite link with a BER of 10^{-7} or better. The total throughput performance corresponding to 95% of capacity with 1, 2 and 8 simultaneous TCP connections was achieved in the presence of long delay such as satellite network. In addition, it was shown that the two-segment splitting technique is suitable for both asymmetric and symmetric satellite networks.

Segment-splitting techniques allow for an optimized load balancing scheme among connections in a satellite segment in order to avoid unwanted degradation of TCP transmission window by the congestion avoidance algorithms. This load balancing scheme is possible because the gateway equipment acts as an intermediary between the TCP end-users.

2.3 Three-segment splitting

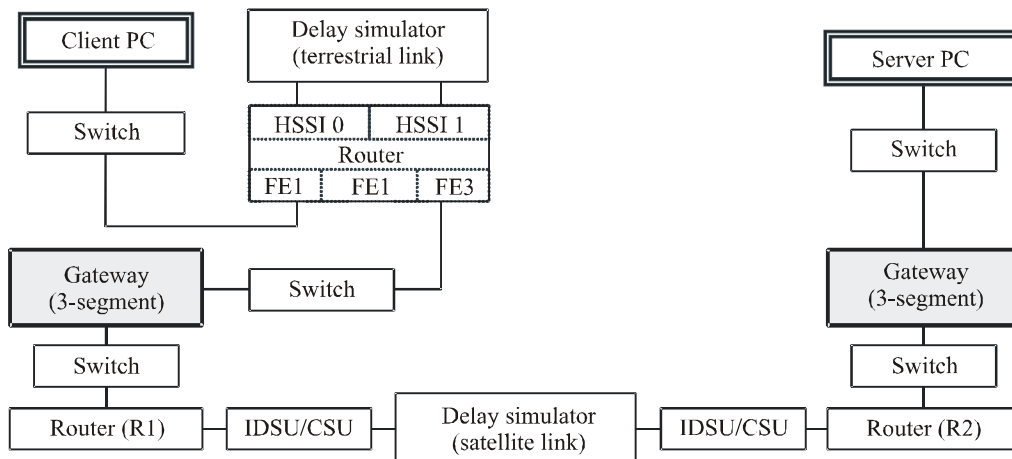
2.3.1 Configuration of satellite networks

The test network is shown in Fig. 18. The network contains two link simulators to simulate the effects of both the satellite link conditions and the terrestrial Internet backbone. For testing of only the satellite link conditions, the terrestrial link delay simulator and Cisco 7206 router were not included in the network. The client machine was connected directly to the Ethernet switch attached to the protocol gateway.

Both the client and server machines are Sun Enterprise 450 ($2 \times$ UltraSPARC-II 296 MHz) with 2 048 Mbytes memory running the Solaris 7 operating system. A client-server application was used for the load generator.

FIGURE 18

Test network configuration



Note 1 – “switch” indicates the 100 Mbit/s Ethernet switch

1711-18

2.3.2 Test procedures

The objectives of the tests were to conduct independent testing to investigate the effectiveness of protocol gateways under various loading conditions, TCP connection rates and error rates for typical satellite link conditions. In addition to simulating satellite conditions, testing also examined the effect of Internet congestion on end-to-end throughput with and without the protocol gateway.

Three types of tests were conducted:

Test 1: Single TCP connection throughput for various link bandwidths

These tests were designed to simulate high-speed LAN and Internet-2 applications where high speed transfers of large data files are common, comparing the performance with and without the protocol gateway.

Test 2: Multiple TCP connections with fixed per-connection bandwidth

These tests were designed to examine the performance benefit of the protocol gateway for ISP links supporting large numbers of small TCP connections. Tests were run for round trip delays of 200 ms to simulate a terrestrial WAN link, and 700 ms to simulate the combination of a 500 ms satellite hop from the user to the Internet backbone and 200 ms delay to reach the server. For simplicity, delay and bit errors were added at a single point, with the protocol gateways located on either side of the link simulator.

Test 3: Multiple TCP connections with terrestrial packet loss

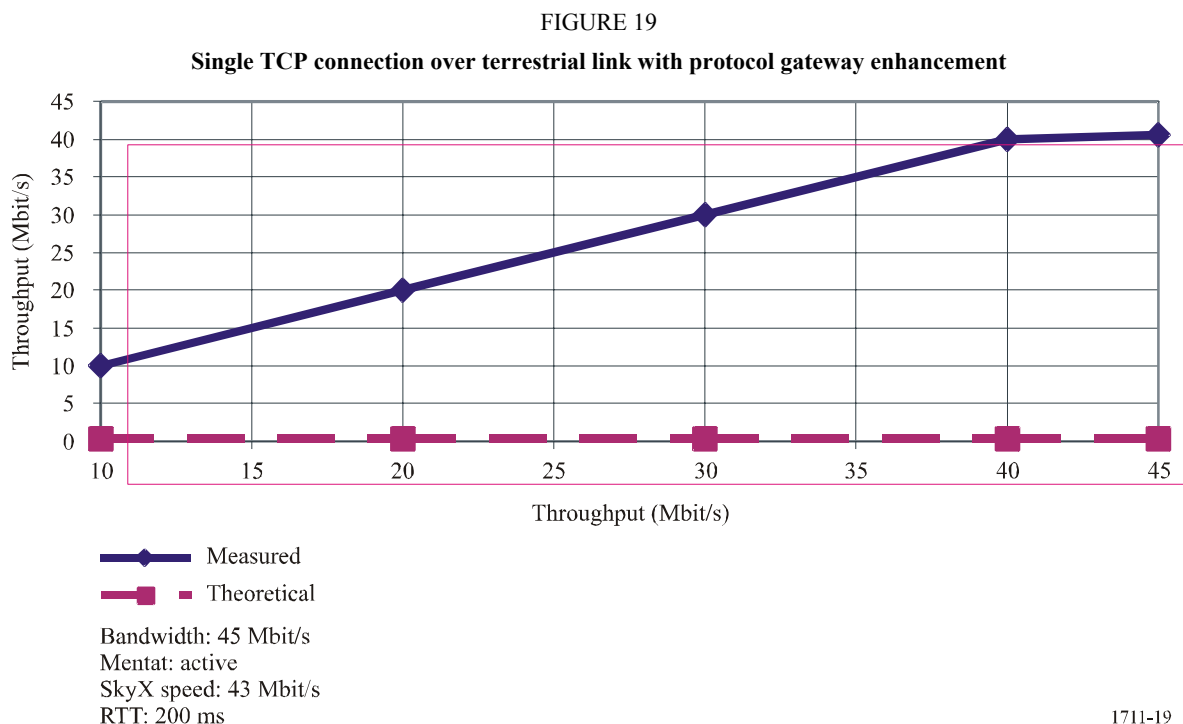
The previous testing was extended to more closely examine the effect of delay and error across the Internet backbone, independent of the satellite hop. A second link simulator was added to simulate the satellite and terrestrial conditions separately. For these tests, the satellite hop was set to a round trip delay of 500 ms with no errors, and the backbone link was set to a delay of 200 ms with various loss rates.

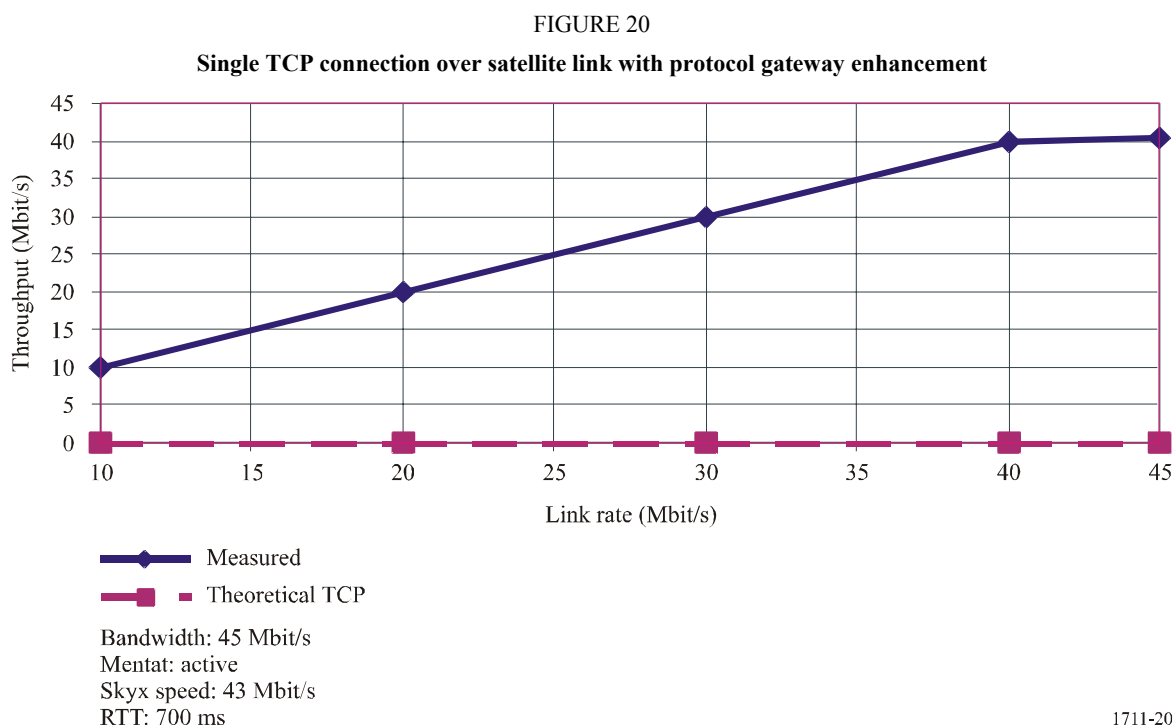
2.3.3 Test results

2.3.3.1 TCP with protocol gateway enhancement

The next set of tests examined the performance enhancement provided by the protocol gateway for a single TCP connection. Figure 19 shows the throughput as a function of the link bandwidth for a round trip delay of 200 ms, comparing the measured throughput using the protocol gateway to the theoretical un-enhanced TCP maximum throughput rate. Figure 20 shows test results for a round trip delay of 700 ms.

For both the 200 ms and 700 ms delay cases, the performance using the protocol gateway is clearly orders of magnitude better than the theoretical maximum TCP throughput. Even despite a 700 ms delay, the protocol gateway allows the connection to take advantage of the full bandwidth available.





2.3.3.2 Multiple TCP connection performance testing

Rather than a single large TCP connection, ISP servicing home users connecting to the Internet support large numbers of small connections on their networks. TCP connection rates are generally limited to the speed of the user's connection to the ISP. The next set of tests was therefore designed to examine the performance of TCP with and without protocol gateway enhancement for large numbers of TCP connections, with each connection limited to 128 kbit/s. Tests were run for both 200 ms and 700 ms round trip delays to compare terrestrial and satellite performance. Various bit error rates were tested for each condition.

2.3.3.3 Multiple TCP connections without enhancement

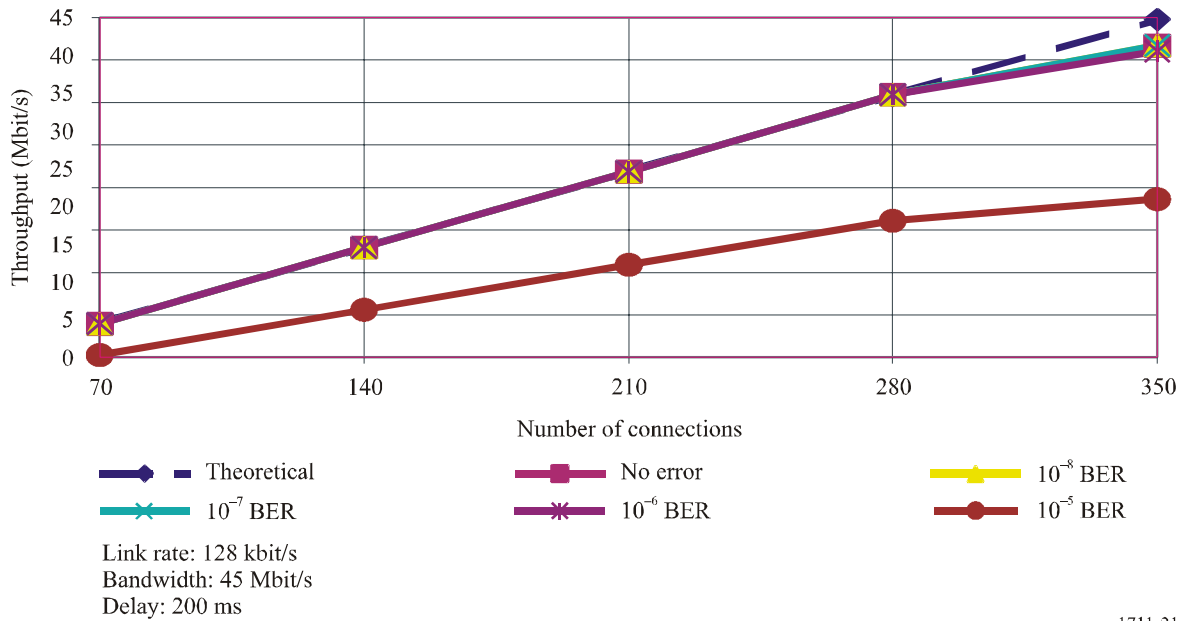
Figures 21 and 22 show the aggregate throughput between client and server without TCP enhancement for multiple 128 kbit/s TCP connections under various bit error rate conditions. With a limit of 128 kbit/s per-connection, 350 connections would be required to fill a 45 Mbit/s link.

Figure 21 shows the aggregate throughput for a terrestrial link. With the 200 ms delay, TCP is able to provide aggregate throughput rates near the theoretical maximum except at high bit error rates.

Figure 22 shows the aggregate throughput for the satellite-based connection. With a 700 ms delay, even without errors, TCP is limited to only 31 Mbit/s for 350 connections. At high error rates, TCP performance drops off rapidly.

FIGURE 21

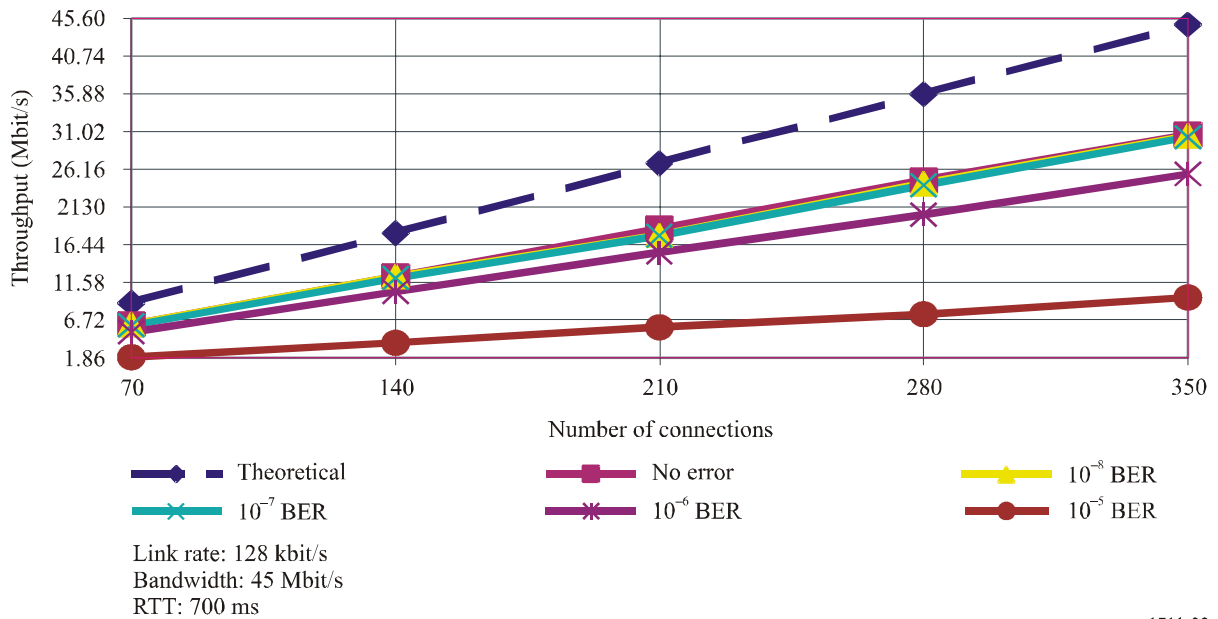
Multiple TCP connections over terrestrial link without enhancement



1711-21

FIGURE 22

Multiple TCP connections over satellite link without enhancement



1711-22

2.3.3.4 Multiple TCP connections with protocol gateway enhancement

Figures 23 and 24 illustrate the effects of adding the protocol gateway to the network. Figure 23 shows the aggregate throughput for a delay of 200 ms, while Fig. 24 shows the results for a delay of 700 ms.

For both the terrestrial and satellite conditions, the protocol gateway allows the connection to utilize the full bandwidth available. For both cases, the performance is essentially identical to the theoretical limit for up to 280 connections.

Compared to § 2.3.3.3, the protocol gateway provides almost 100% improvement in aggregate throughput at a packet loss rate of 10% (corresponding to a BER of 1×10^{-5} for 1 500-byte packets). For the satellite-based network, the protocol gateway provides a substantial increase in aggregate bandwidth at low bit error rates, and at a packet loss rate of 10%, the aggregate throughput for 350 connections with the protocol gateway is 33 Mbit/s compared to only 10 Mbit/s for enhanced TCP.

FIGURE 23

Multiple TCP connections over terrestrial link with protocol gateway enhancement

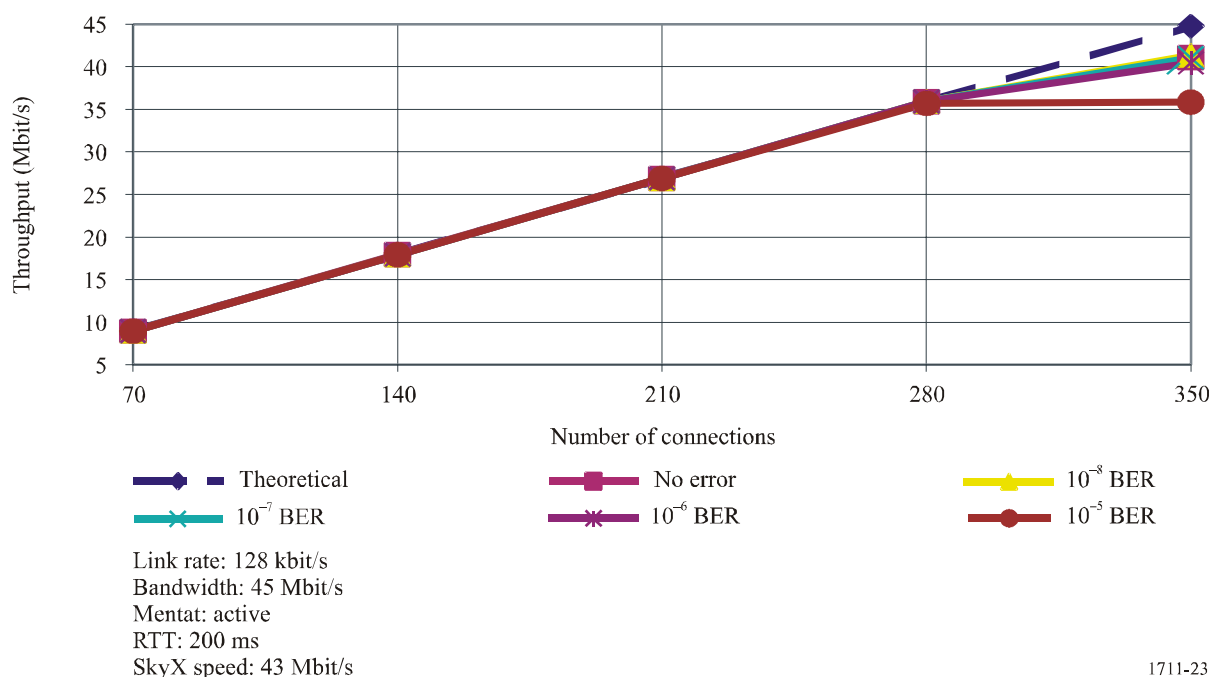
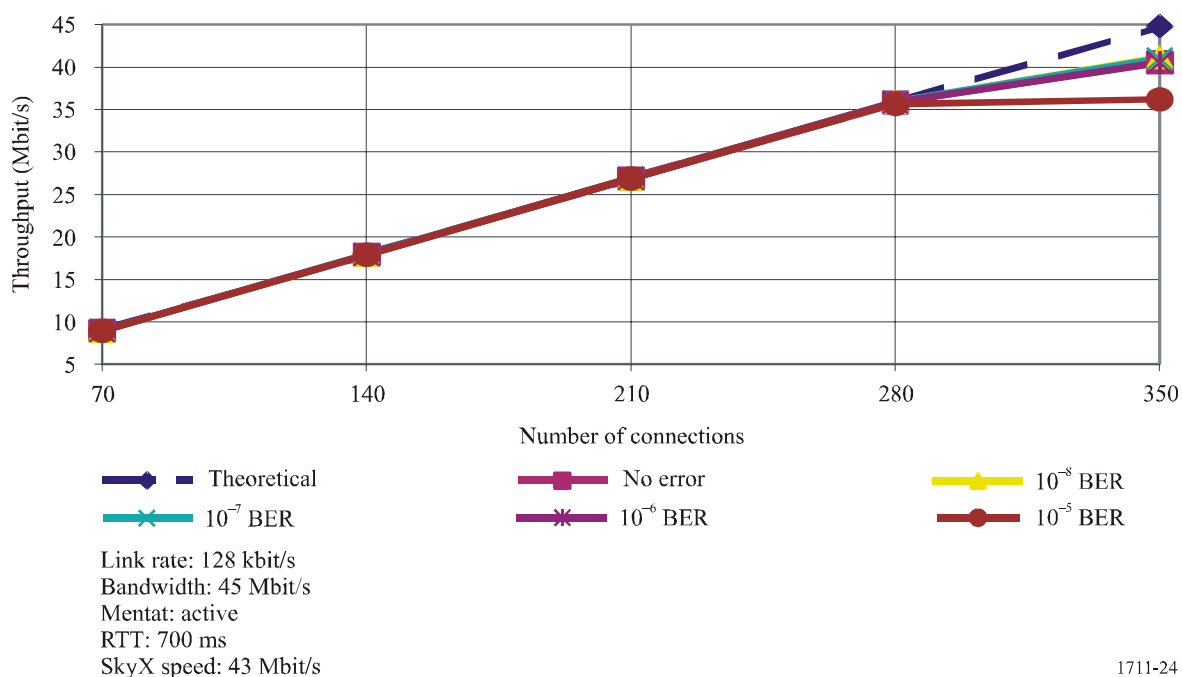


FIGURE 24

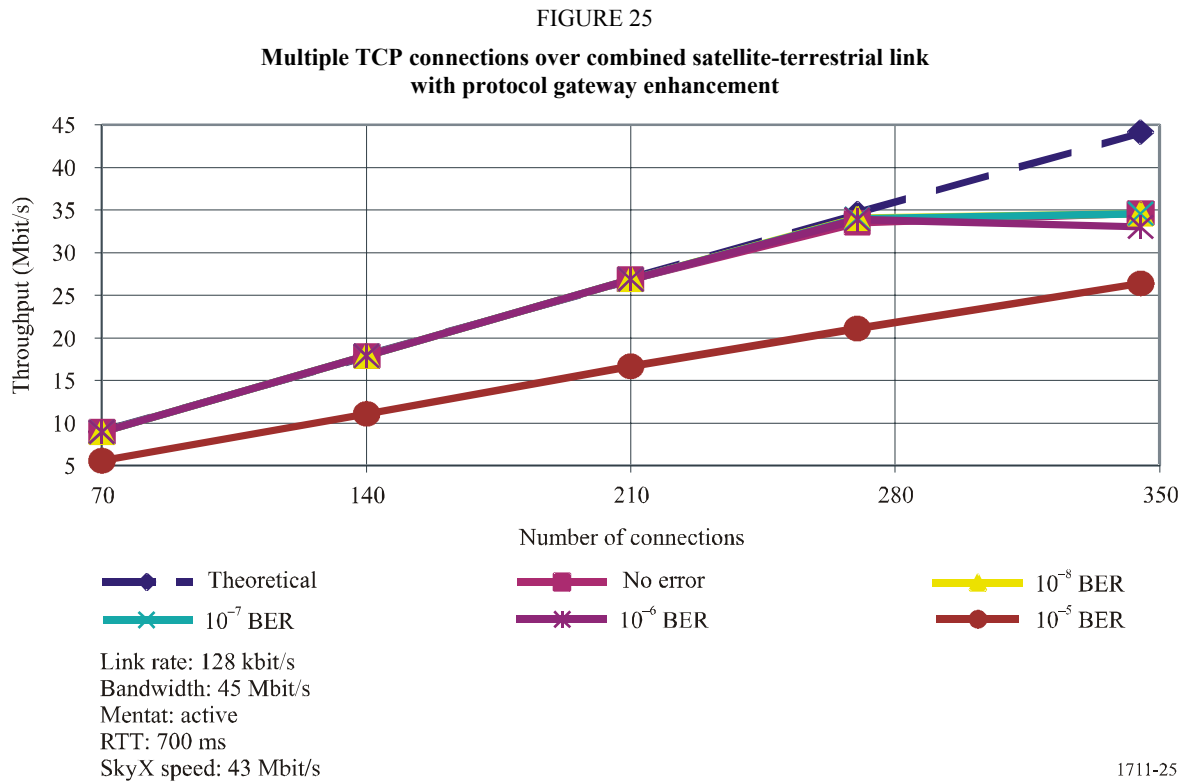
Multiple TCP connections over satellite link with protocol gateway enhancement



2.3.3.5 Multiple TCP connections with terrestrial data loss

In considering the performance of Internet users connected via a satellite-based ISP, the connection will traverse both the satellite hop and then cross the Internet backbone to reach the server. The connection may encounter data loss due to congestion over the Internet backbone. To more accurately model conditions that the end user experiences, the next set of tests divided the connection into a lossless satellite link with a 500 ms delay, combined with a backbone path of 200 ms delay and various error rates. In these tests, the protocol gateways are located on either side of the satellite link. Standard TCP is used for the portion of the connection over the backbone between the server and server-side protocol gateway.

As shown in Fig. 25, the protocol gateway allows the connection to maintain full speed with almost no degradation except at high packet loss rates. Comparing Fig. 25 with Figs. 22 and 24 illustrates that the protocol gateway is able to overcome the latency and errors on the satellite link as well as the delay and most of the packet loss on the backbone link.



2.3.4 Conclusions

The results of this testing show that protocol gateway/connection splitting devices can improve the throughput for carriers with TCP type traffic on satellite links with up to 700 ms in delay. The tests also show that the TCP throughput is not affected as long as the link BER is better than 10^{-7} .

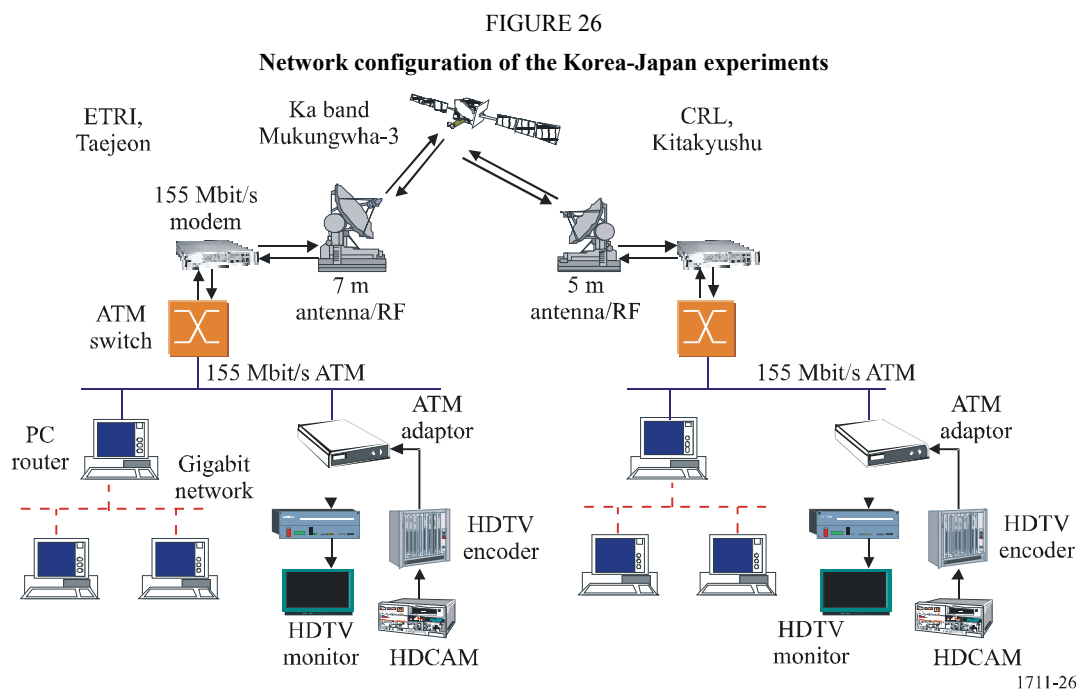
3 TCP performance over a satellite ATM network

Paragraph 3.1 describes the environment and configurations of experimental network. Paragraph 3.2 presents the results of TCP throughput when the throughput was measured in pure ATM-based satellite network. Paragraph 3.3 discusses the TCP behaviour when broadband ATM-based satellite network interconnects with high-speed terrestrial networks such as gigabit Ethernet. Paragraph 3.4 summarizes the conclusions reached through this experiment.

3.1 Network configuration

Figure 26 shows the configurations of Korea-Japan high-speed satellite ATM network. In this joint experimentation, the two ground stations with 7 m antenna at ETRI, Korea and 5 m antenna at CRL, Japan were installed respectively. The main specifications of the Korea-Japan 155 Mbit/s satellite ATM link are as follows:

- Satellite: Mukungwha-3
- Frequency band: uplink: 27.5-31 GHz, downlink 17.7-21.2 GHz
- Maximum TWTA power: 125 W
- Normal e.i.r.p. (Mukungwha-3): 71 dBW
- G/T (45° elevation): 32 dB/K (minimum)
- TC 8-PSK modulation/demodulation
- Coding: $K = 7$, 7/8 convolutional RS
- Bit rate: 155.52 Mbit/s
- Allocated bandwidth: 80 MHz two channels.

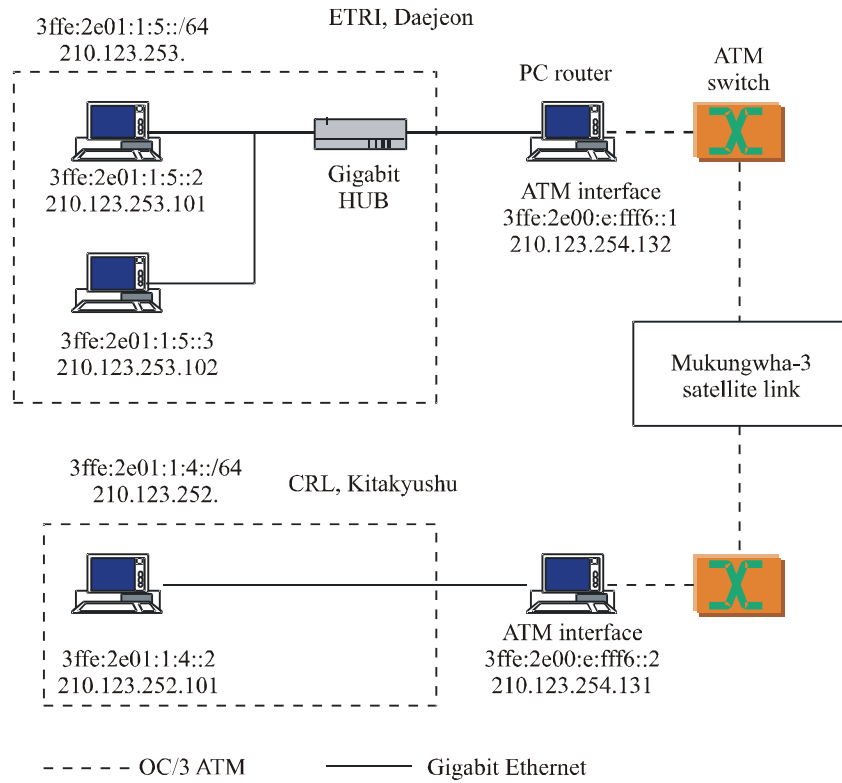


The whole network could be divided into two networks – gigabit subnet and OC-3 ATM backbone network including GSO satellite link. PC-based routers that have both gigabit and ATM network interfaces which interconnect two networks. The experimental network was configured with IPv4 and IPv6 addresses. The ATM-based satellite network was used for MPEG transmission experiment.

For two types of applications – Internet and real-time video transmission, an ATM-based network was built for the experiment. Figures 27 and 28 show the detailed configurations and protocol stacks of the experimental network.

FIGURE 27

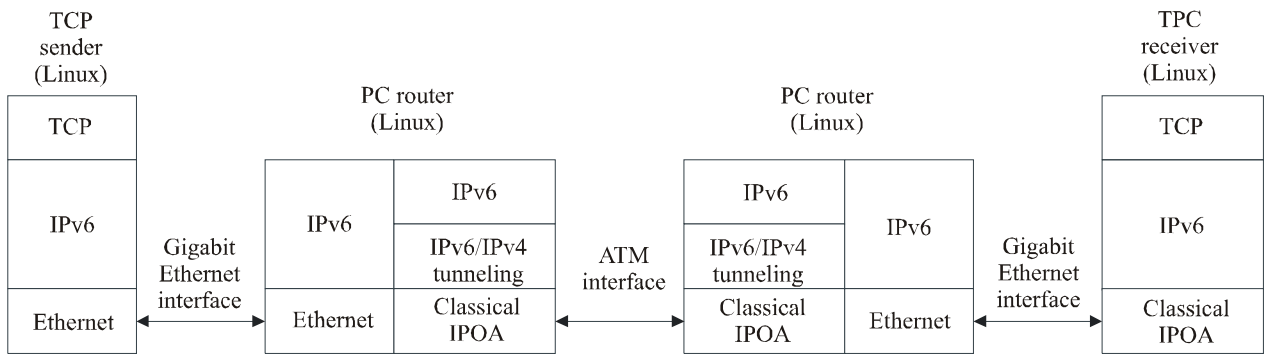
Experimental network configuration



1711-27

FIGURE 28

Protocol stacks of experimental network interconnected via PC router

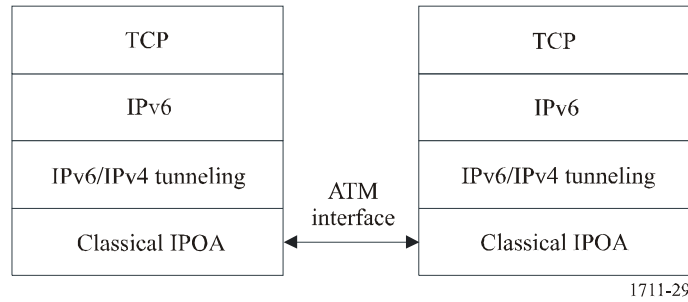


1711-28

3.2 TCP performance in ATM-based satellite-only network

The TCP throughput in pure ATM network with 540 ms GSO satellite round trip delay was first measured. IPv6/IPv4 tunnelling was used because IPv6 in IP over ATM was not completely implemented yet. The protocol stack of IPv6/IPv4 tunnelling over ATM-based satellite network is shown in Fig. 29.

FIGURE 29
Protocol stack of IPv6/IPv4 tunnelling on ATM



The MTU size in IP over ATM is usually 9 180 bytes. MTU size in IPv6/IPv4 tunnelling interface is 9 160 bytes because IPv4 header size is 20 bytes. Therefore the MSS is a maximum of 9 100 bytes (the sizes of IPv6 header and TCP header are 40 and 20 bytes respectively). However when TCP scale option is used to enlarge TCP window size in ATM-based satellite network, the MSS is reduced by the TCP option bytes. When IP packets are encapsulated into AAL5, the maximum theoretical bandwidth can be calculated. The number of bytes of padding necessary is:

$$PAD = 48 - ((8 + 20 + 40 + 20 + 9100 + 8) \text{MOD } 48) = 20$$

Then the PDU utilization is:

$$\eta = \frac{9100}{(8 + 20 + 40 + 20 + 9100 + 20 + 8)} = \frac{9100}{9224} = 0.986$$

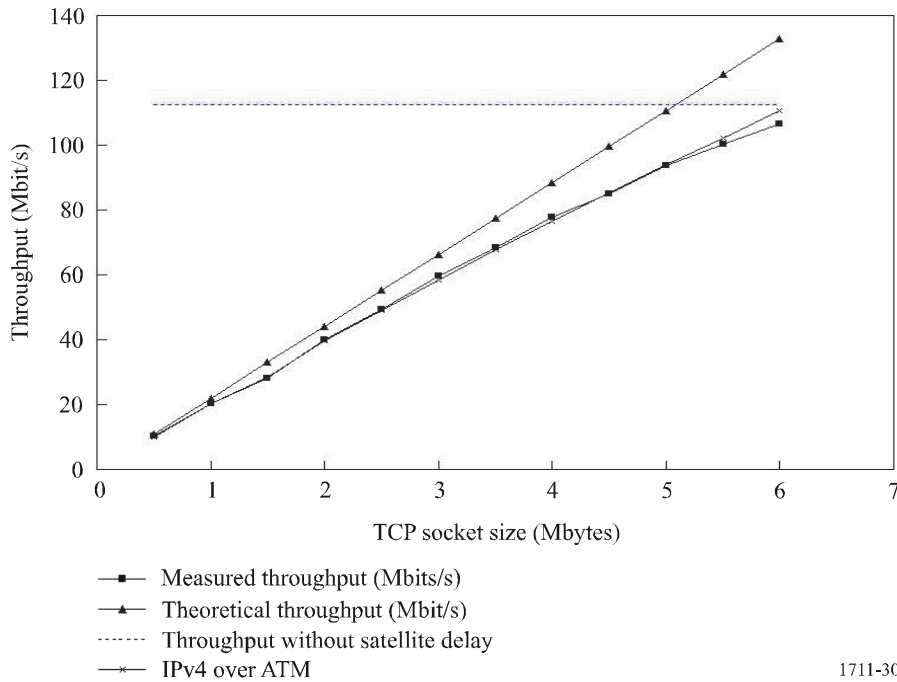
The pure cell rate of STM-1 is about 149 Mbit/s and the bandwidth of 48 bytes payload in ATM cells is about 134.94 Mbit/s. Therefore the theoretical maximum bandwidth of TCP in OC-3 network is given by:

$$BW_{TCP} = 134.94 \cdot \eta \cong 133.05 \text{ Mbit/s}$$

Figure 30 shows the TCP throughput with various TCP socket (or window) sizes. When TCP socket size was 6 Mbytes, throughputs of 113 Mbit/s using IPv4 and of 106 Mbit/s using IPv6 were obtained. The performance was 95% of throughputs without any satellite delay. The TCP throughput in a pure ATM network with 540 ms GSO satellite round trip delay was then measured.

FIGURE 30

TCP throughput of ATM-based satellite network with various TCP window sizes

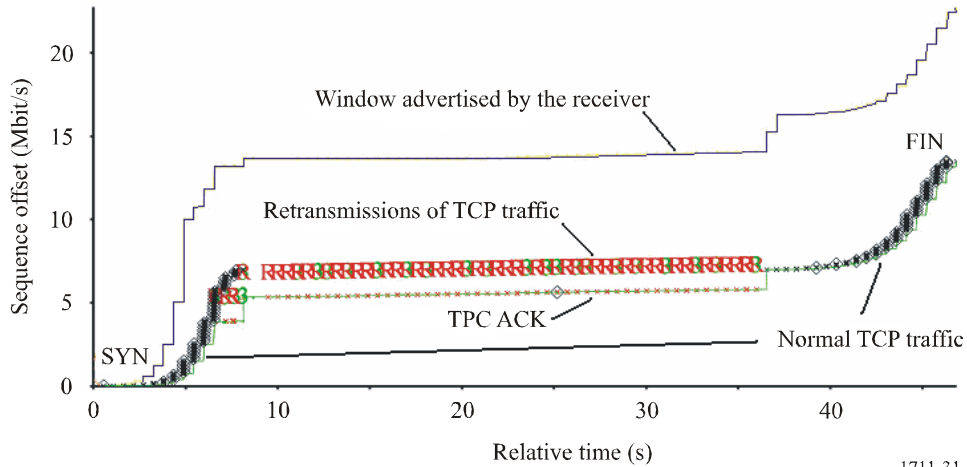


1711-30

3.3 TCP behaviours in heterogeneous networks including satellite link

Next the TCP throughput was measured when the source was located on a gigabit subnet and the destination located on another gigabit subnet. When the TCP socket size was set to about 6 Mbytes in 540 ms satellite delay network, the maximum throughput was only about 2 to 3 Mbit/s. This is due to the burstiness of the TCP traffic when large TCP window size is set for good throughput in LFNs. When the TCP window size is increased to use full bandwidth of the LFN, the TCP traffic is bursty because the physical interface of the sender (i.e. gigabit Ethernet) is faster than the ATM backbone. The intermediate router should have large buffer to prevent packet losses. Existing routers do not have large enough buffers to allow for large TCP window sizes. Therefore the packet loss of the intermediate router degrades the TCP throughput. Figure 31 shows a TCP sequence graph of the TCP sender. In the first 8 s, there is a TCP slow start. As the TCP window size increases, there are packet losses due to buffer overflow in the intermediate router. There are many retransmissions due to these packet losses. TCP restarts the slow start behaviour after long time (about 27 s later). This transmission pattern (loss and retransmit) repeats and the overall TCP throughput performance becomes poor.

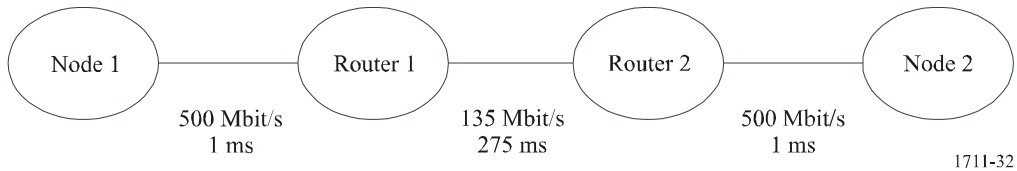
FIGURE 31
TCP sequence graph in the TCP traffic source



3.3.1 Simulation of buffer size

The buffer size of intermediate router can be assessed by calculations and simulation. The configuration of simulation model using a network simulator is shown in Fig. 32. For the simulation, the maximum available TCP bandwidth in gigabit and ATM networks is set to 500 Mbit/s and 135 Mbit/s respectively.

FIGURE 32
Configuration of simulation model



The delay of a gigabit subnet is negligible compared to the GSO satellite delay in the ATM link and MSS of the Ethernet which is 1 430 s in IPv6. Therefore the required TCP window size is:

$$W = \frac{BW \cdot RTT}{8} = \frac{135 \times 10^6 \times 0.540}{8 \times 1\,430} \cong 6\,374 \text{ packets}$$

R_1 and R_2 being the packet rates of the gigabit network and the ATM network respectively ($R_1 > R_2$), the rate of TCP acknowledgement is set up by R_2 (the lowest link in the end-to-end connection). The packet rate in the slow start phase is $R_1 = 2 \times R_2$ because the TCP sender transmits two packets when it receives one ACK. Assuming W_{max} is the maximum TCP window size during slow start, the number of packets in the queue of the intermediate router is given by:

$$P(t) = [R_1 - R_2] \cdot t = [2 \cdot R_2 - R_2] \cdot t = R_2 \cdot t$$

$$t = \frac{T}{R_2}, 2 \cdot \frac{T}{R_2}, 3 \cdot \frac{T}{R_2} \dots \leq \frac{W_{max}}{2 \cdot R_2}$$

where T is the start time of the TCP window during slow start.

Therefore the maximum packet size in the queue is:

$$P\left(\frac{W_{max}}{2 \cdot R2}\right) = R2 \cdot \frac{W_{max}}{2 \cdot R2} = \frac{W_{max}}{2}$$

That is, during slow start sender sends twice as much as the number of ACK it received during one RTT. The required queue buffer of intermediate node is the half of the maximum TCP window size because in first of the RTT router receives as twice the rate as it sends and so it drains the queue in the next half of RTT. When the maximum available bandwidth of TCP in gigabit network and ATM network are 500 Mbit/s and 135 Mbit/s respectively, theoretically the required buffer size of the experimental network is about 3 187 packets. Figures 33 and 34 are the result of simulation using network simulator-2. Figure 33 shows TCP sequence, ACK and the number of packets in the intermediate queue. Figure 34 shows TCP throughput during first 10 s in sender and receiver. The maximum throughput is obtained when TCP buffer size was about 6 374 packets and at that time the maximum of intermediate router queue was 2 279 packets.

FIGURE 33

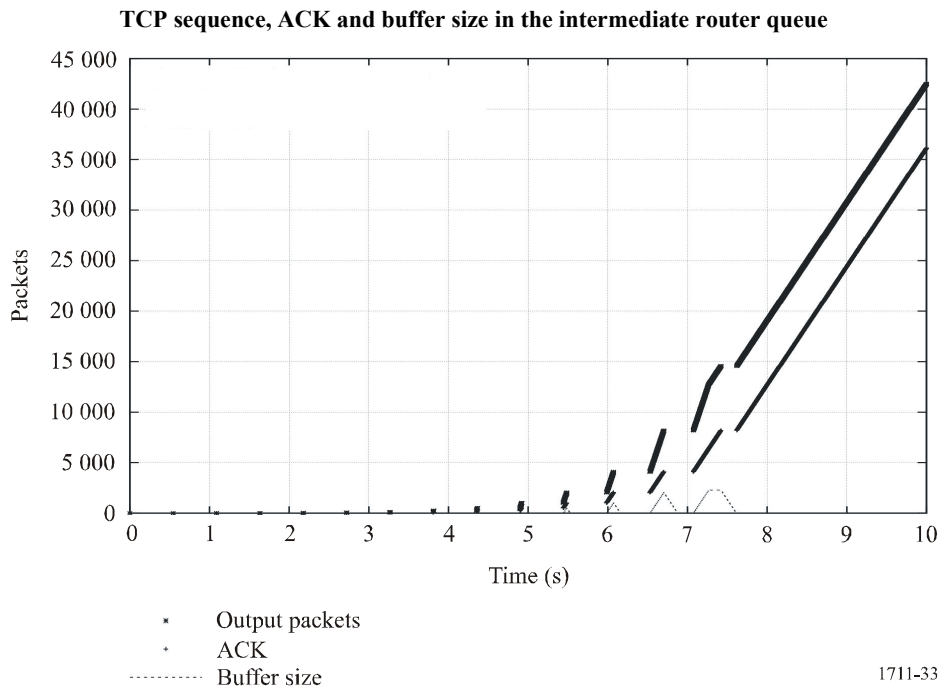
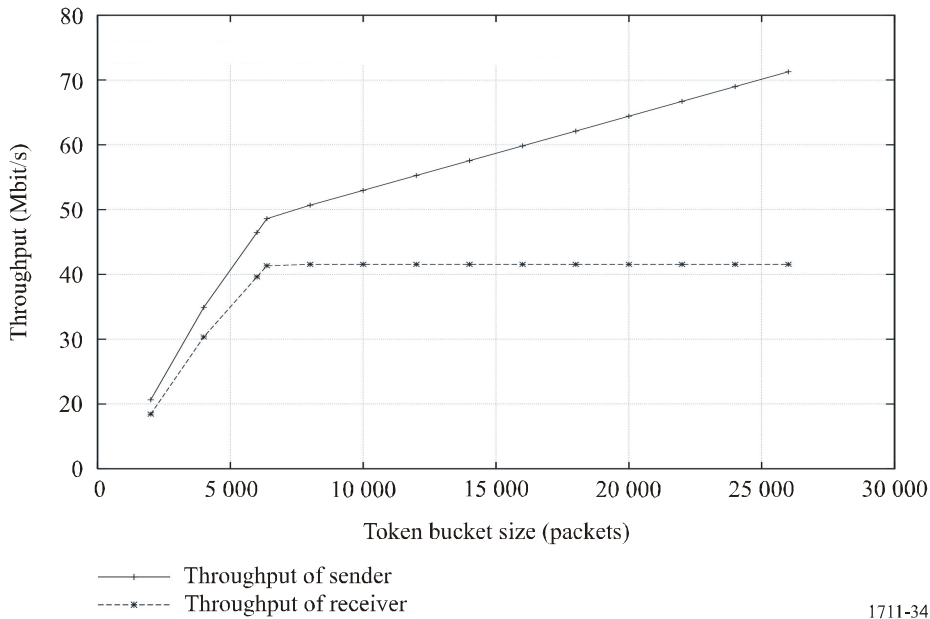


FIGURE 34

TCP throughputs with various buffer sizes



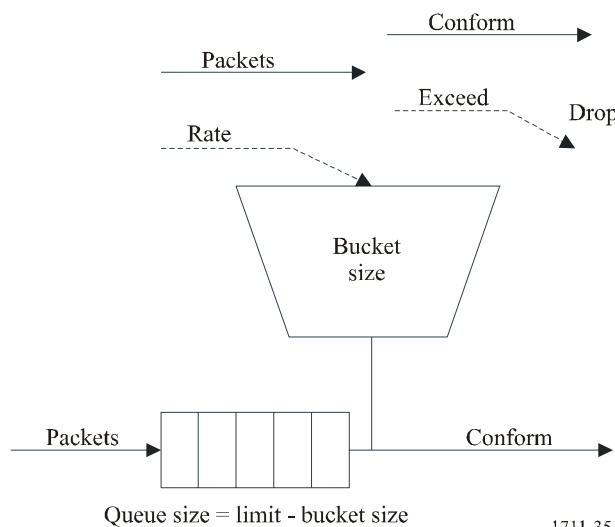
1711-34

3.3.2 TCP performance with traffic control in TCP sender

A traffic shaping mechanism called TBF (supported by Linux advanced networking option) was then used. TBF regulates the rate of TCP burst traffic. Figure 35 shows the basic parameters for TBF queuing discipline. *Rate* is the rate the bucket refills with tokens – which represents the average transmission rate of a traffic flow. The *bucket size* or *burst size* is the number of tokens the token bucket can store. The *limit* parameter is the sum of the bucket size and the size of the queue. If limit is equal to the bucket size and the queue size is zero, non-conforming packets are dropped. Thus the stream is policed. If the limit is greater than the bucket size some non-conforming packets are queued.

FIGURE 35

Basic parameters for TBF queuing discipline



1711-35

Figure 36 shows the TCP throughput with various TCP socket size when TBF was used in TCP sender with 6 Mbytes fixed TCP window size (for comparison, the case where there is no satellite delay is also shown). Figure 37 shows the TCP throughput with various token bucket sizes. A token bucket of 120-130 kbytes results in the best throughput. Above 130 Mbytes of bucket size, there are packet losses in the intermediate router due to the burstiness of the TCP traffic.

FIGURE 36

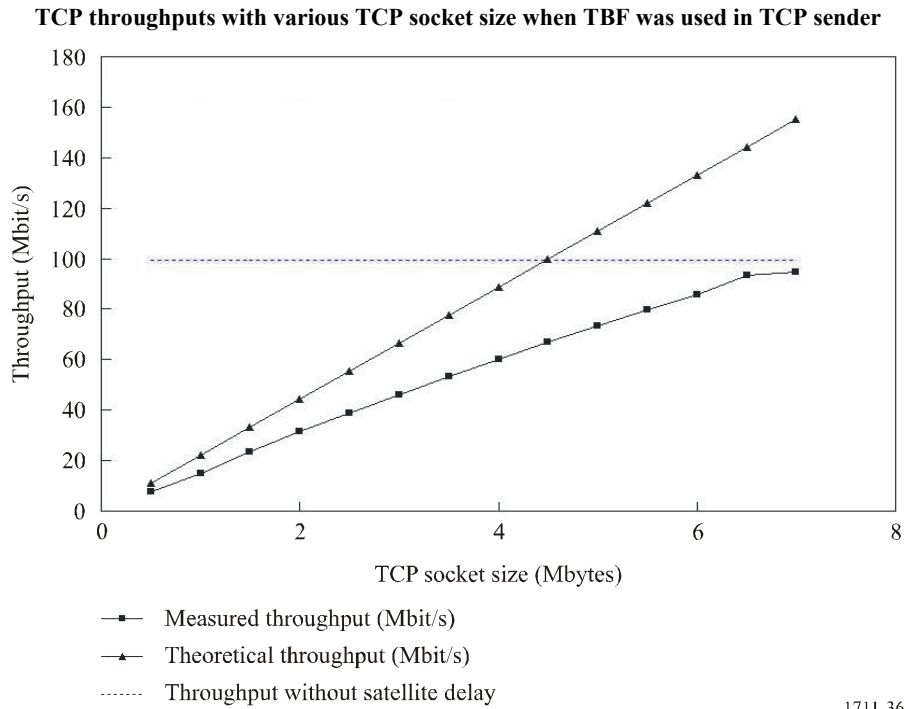
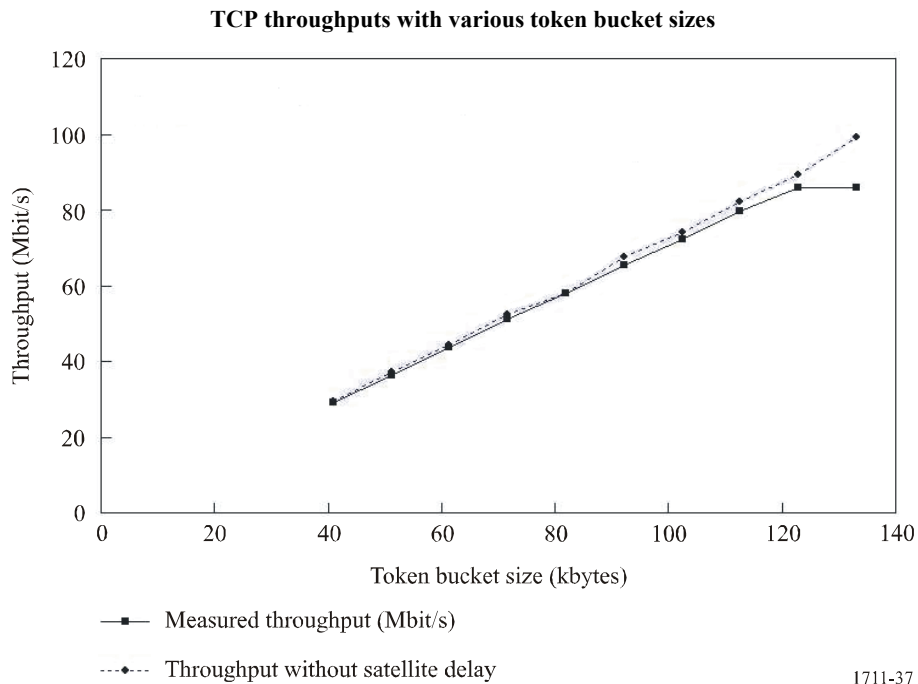


FIGURE 37



3.4 Conclusions

TCP throughput was measured and the TCP internal behaviours in the heterogeneous high-speed network including GSO satellite link were analysed. The experiment results are summarized below:

1 For a maximum TCP throughput of 155 Mbit/s over a satellite network, about 6Mbytes TCP socket size is needed.

2 When TCP source is on another network media that is faster than ATM-based satellite backbone, mechanisms to lessen burst traffic due to large TCP window are needed. Two solutions can be considered: large buffer in the intermediate routers or traffic control at the TCP source. For large buffer in intermediate routers, the queue buffer size to prevent packet losses due to the burst TCP traffic was estimated and verified through simulation. Alternatively, traffic control mechanism at the TCP source was considered: a throughput of about 95 Mbit/s (i.e. 95% of the maximum throughput without satellite delay) was achieved.

3 TCP window scale option for large TCP windows is one of the solutions to improve TCP performance in the long delay networks that include GSO satellite link. The use of large TCP windows in long delay network improves the throughput up to about 90% of theoretical maximum. But the large TCP window generates the burst traffic in short time. Especially in the case of the heterogeneous network that is composed of different physical media and different link-layer control protocols, there are serious packet losses in the intermediate router due to burst traffic. To avoid this situation, the network designer should consider possible solutions and verify them through experiments.

4 TCP performance in satellite access networks

This section presents the results of tests conducted by Star One, a satellite company from Brazil. Section 4.1 describes the network architecture used in the tests. Section 4.2 provides the system performance measurements using three-segment splitting and includes performance results.

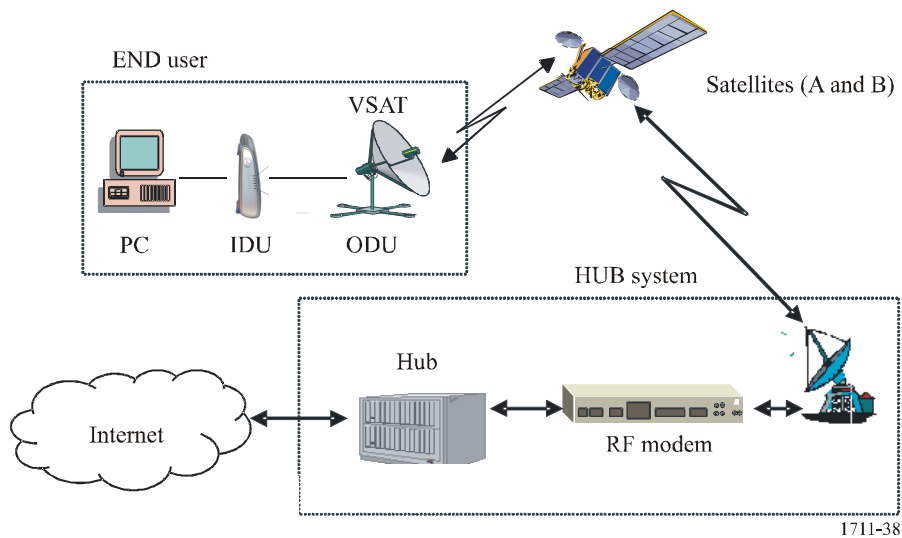
The Hub and VSAT local acknowledgements improve TCP performance by eliminating windowing limitation. The local acknowledgements eliminate slow start mechanism and improve TCP throughput efficiency. The IP headers compression used in this test reduces the system overhead and increases efficiency.

4.1 Network architecture and configuration

4.1.1 Network architecture

The system architecture encompasses a Hub with a DVB-S stream (outbound) and a proprietary reverse link (inbound). The central Hub is connected to an Internet backbone. Figure 38 shows the architecture of the system used to reach the performance data.

FIGURE 38
Network architecture



1711-38

4.1.2 Configuration

4.1.2.1 VSAT and satellite characteristics

The two geostationary satellites are operating in the range of 14 068 kHz to 14 214 kHz (uplink) and 11 774 kHz to 11 919 kHz (downlink).

Each platform in use has a forward link (outbound) of 48 Mbit/s and 140 return channels (inbound) of 76.8 kbit/s.

This forms three clusters (Cluster 1, Cluster 2 and Cluster 3) using two different satellites (satellite A and satellite B) in the 14/11 GHz frequency band with a total of 8 000 VSATs installed on the Brazil territory. The RTT considered is 800 ms.

The cluster characteristics are showed below:

Cluster 1 (Satellite A): divided in 3 HSPs with around 1 200 VSATs

Cluster 2 (Satellite A): divided in 3 HSPs with around 3 200 VSATs

Cluster 3 (Satellite B): divided in 4 HSPs with around 3 600 VSATs

The VSAT characteristics are shown in Table 14 below. The various necessary E_b/N_0 ratios on the outbound link are shown in Table 15.

TABLE 14
VSAT characteristics

Clusters in use	Antenna size (m)	SSPA power (W)	Maximum inbound rate (kbit/s)	Maximum outbound rate (kbit/s)
1, 2 and 3	0.96	1	50	320

TABLE 15

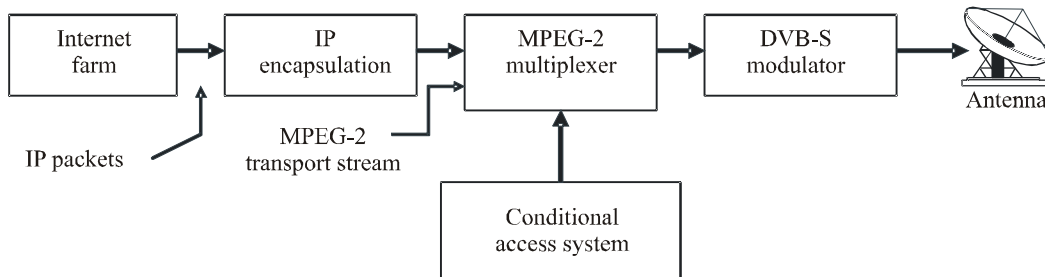
Characteristics of the outbound link

FEC ratios	DVB-S mode
	Minimal E_b/N_0 (dB)
1/2	4.5
2/3	5.0
3/4	5.5
5/6	6.0
7/8	6.4

4.1.2.2 Protocols

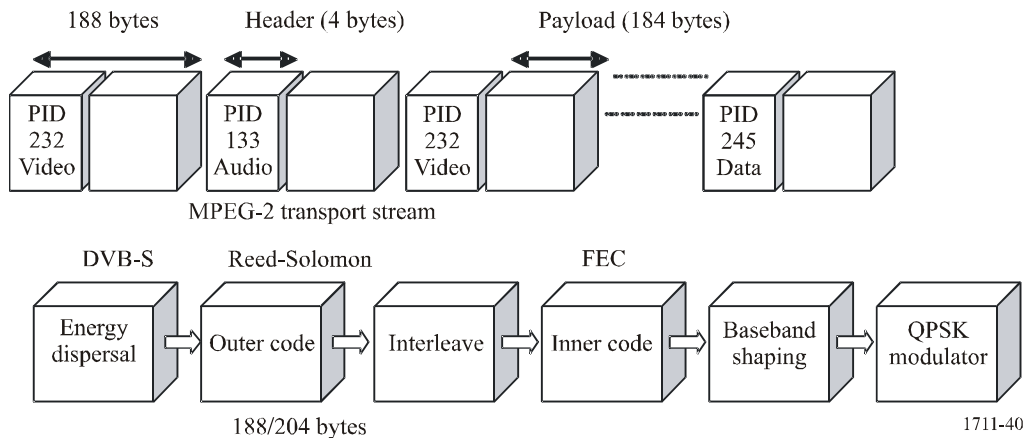
The protocol utilized in the uplink from the Hub to the VSAT (outbound) is a MPEG-2 transport stream over DVB-S. The IP packets are encapsulated, multiplexed and modulated in the Hub as described in Figs. 39 and 40.

FIGURE 39
MPEG over DVB-S



1711-39

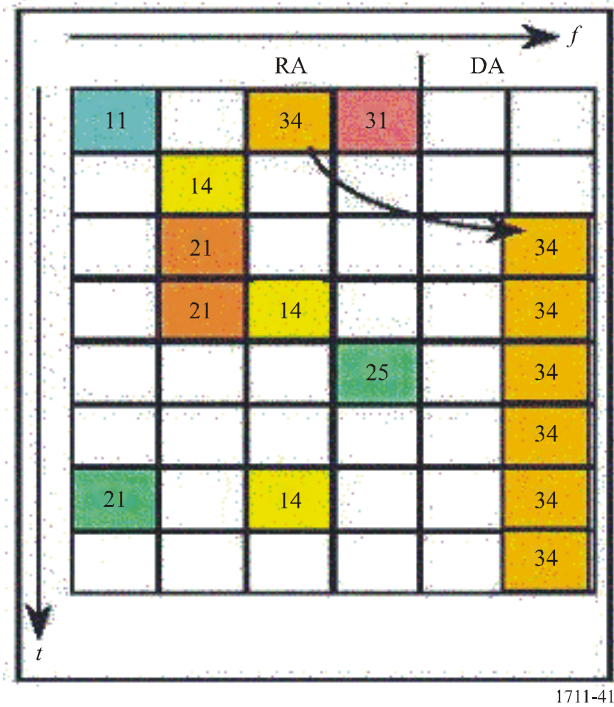
FIGURE 40
MPEG-transport stream



1711-40

The reverse channels (inbound link) are MF-TDMA channels, divided in RA channels and DA channels as described in Fig. 41.

FIGURE 41
Reverse channels



4.2 Results of performance measurement

The measurement of availability included Clusters 1, 2 and 3. The measurement of throughput and traffic only included Clusters 2 and 3.

4.2.1 Availability

Table 16 shows the availability of each different cluster and the availability and unavailability of the system caused by rain and others atmospheric conditions in each month of 2003.

TABLE 16
Availability of the outbound system

Month	Clusters			System			
	Cluster 1	Cluster 2	Cluster 3	Availability (rain, scintillation, etc.)	Unavailability (rain, scintillation, etc.)	Unavailability (rain, scintillation, etc.)/h (month to month)	Availability (rain, scintillation, etc.)/h (month to month)
January	100.0	99.0	N/A	98.6	1.4	10.416	733.584
February	100.0	100.0	N/A	98.9	1.1	7.392	664.608
March	99.9	100.0	N/A	99.8	0.2	1.488	742.512
April	100.0	100.0	N/A	100.0	0.0	0.000	744.000
May	100.0	100.0	99.9	100.0	0.0	0.000	720.000
June	100.0	100.0	99.9	100.0	0.0	0.000	744.000
July	100.0	100.0	100.0	100.0	0.0	0.000	720.000
August	100.0	100.0	100.0	100.0	0.0	0.000	744.000
September	100.0	100.0	100.0	100.0	0.0	0.000	720.000
October	99.9	99.9	99.9	100.0	0.0	0.000	744.000
November	100.0	99.9	100.0	99.9	0.1	0.720	719.280
December	100.0	100.0	100.0	99.9	0.1	0.744	743.256
Year average	100.0	99.9	100.0	99.8	0.2	1.730	728.270

Figure 42 shows the availability of Cluster 2, which uses 3 200 VSATs and satellite A, as a consequence of atmospheric conditions in each month of the year of 2003.

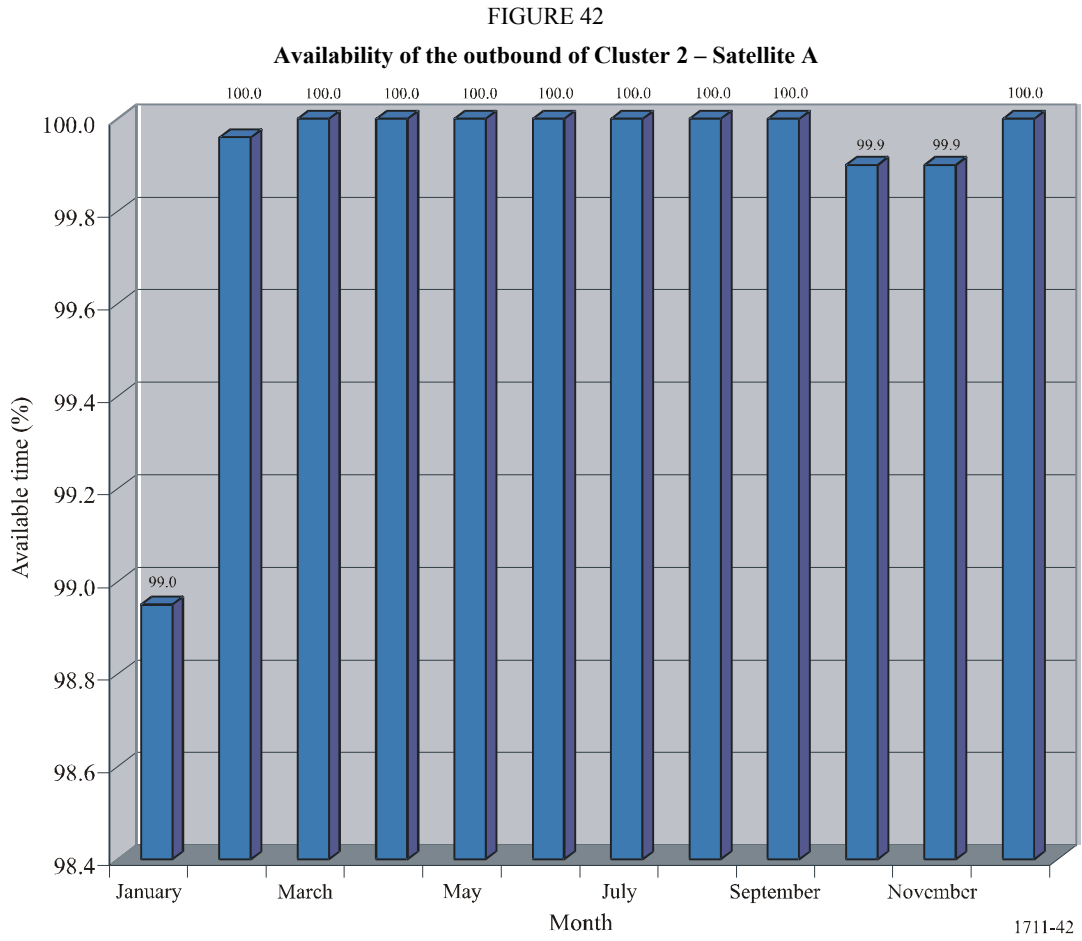


Figure 43 shows the availability of Cluster 3, which uses 3 600 VSATs and satellite B, as a consequence of rain and other atmospheric conditions from May until December 2003.

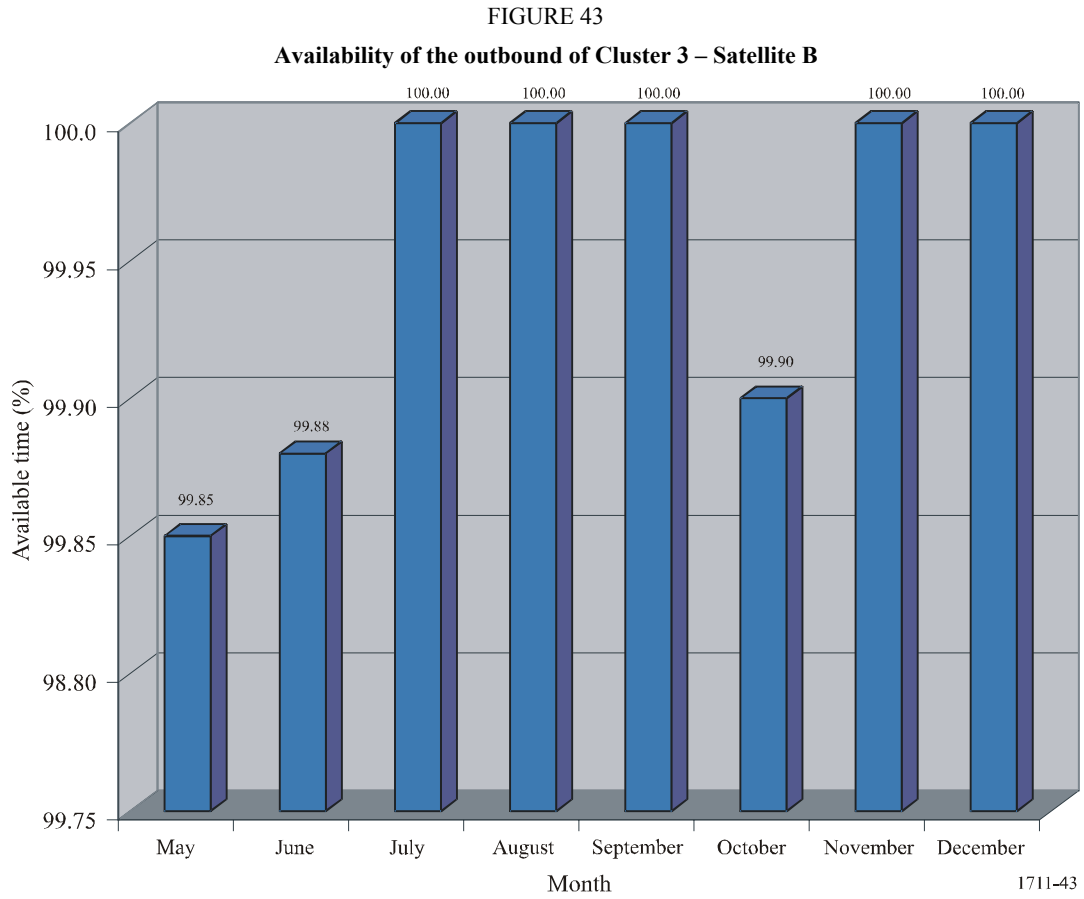
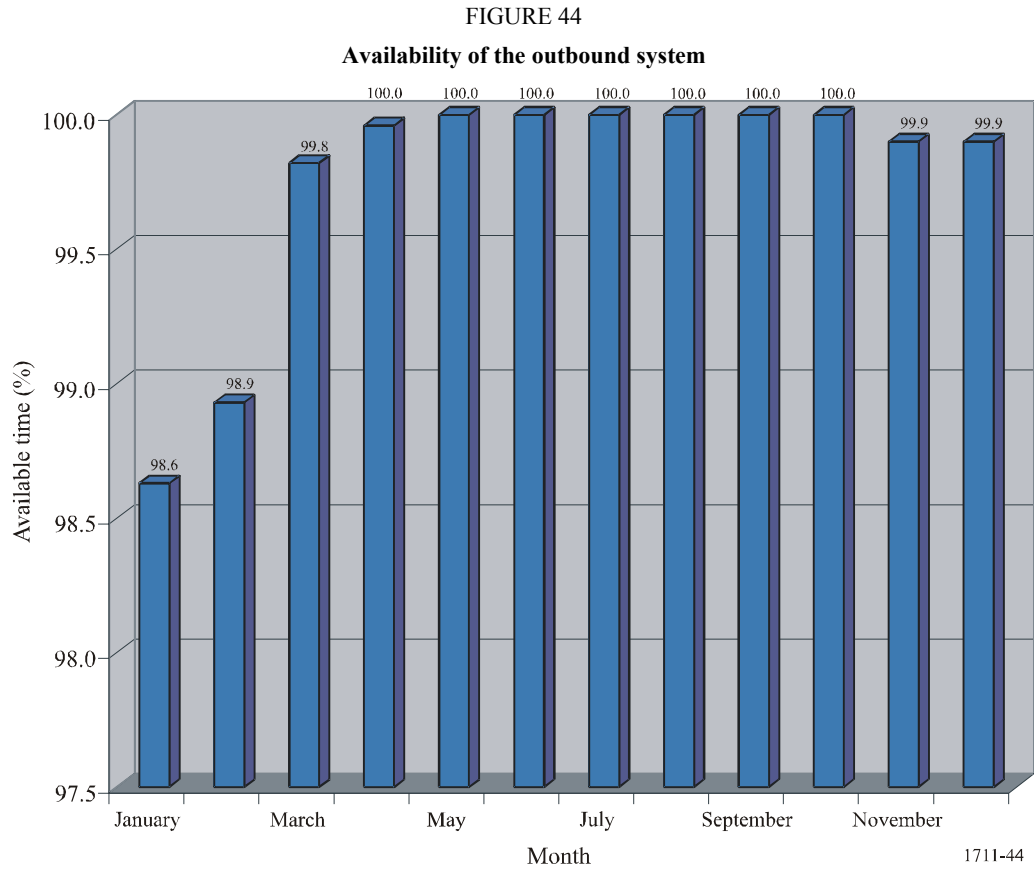


Figure 44 shows the availability of the outbound system (8 000 VSATs), which uses satellites A and B, as a consequence of rain and other atmospheric conditions in each month of 2003.



4.2.2 Throughput

Table 17 shows throughputs of Cluster 2 and its respective HSP. About 200 measurements/day were performed during each day of February 2004. The maximum throughput is 319.11 kbit/s.

TABLE 17
Average throughput of Cluster 2

Reference date	Average value (kbit/s)			
	Cluster 2	HSP 1	HSP 2	HSP 3
02/01/04	314.14	314.96	313.13	314.32
02/02/04	307.66	314.31	303.18	305.49
02/03/04	306.35	312.83	303.72	302.51
02/04/04	306.85	313.92	303.61	303.02
02/05/04	304.75	308.57	303.19	302.48
02/06/04	303.58	306.76	305.28	298.70
02/07/04	310.87	311.00	309.50	312.10
02/08/04	314.78	316.82	311.92	315.60
02/09/04	303.58	305.23	304.28	301.23
02/10/04	302.13	304.34	301.11	300.93
02/11/04	302.33	304.77	299.06	303.16
02/12/04	313.03	312.16	316.88	310.04
02/13/04	306.89	304.71	312.99	302.97
02/14/04	311.65	311.76	316.49	306.69
02/15/04	319.11	317.44	321.36	318.52
02/16/04	305.80	307.72	309.28	300.40
02/17/04	313.06	313.62	316.66	308.90
02/18/04	306.48	305.78	310.73	302.93
02/19/04	312.25	304.46	308.19	324.09
02/20/04	303.92	304.96	305.54	301.26
02/21/04	311.64	313.07	311.59	310.25
02/22/04	314.45	316.07	312.51	314.77
02/23/04	311.23	312.86	310.76	310.08
02/24/04	312.89	313.44	313.04	312.19
02/25/04	306.45	307.98	304.75	306.63
02/26/04	303.26	304.03	301.08	304.68
02/27/04	304.36	306.54	304.20	302.35
02/28/04	312.95	314.71	313.99	310.15
02/29/04	313.78	314.66	313.43	313.26

Table 18 shows throughputs of Cluster 3 and its respective HSP. About 200 measurements/day were taken during each day of February 2004. The maximum throughput was 262.35 kbit/s.

TABLE 18
Throughput of Cluster 3

Reference date	Average value (kbit/s)				
	Cluster 3	HSP 1	HSP 2	HSP 3	HSP 4
02/01/04	255.55	250.85	256.56	263.72	263.72
02/02/04	253.18	244.96	252.73	267.17	267.17
02/03/04	248.84	242.61	248.14	261.28	261.28
02/04/04	247.09	241.08	245.35	255.85	255.85
02/05/04	250.08	245.11	247.89	263.84	263.84
02/06/04	247.32	240.75	244.80	259.46	259.46
02/07/04	249.72	243.19	247.17	267.40	267.40
02/08/04	249.80	242.75	249.98	266.99	266.99
02/09/04	253.24	244.22	248.74	274.18	274.18
02/10/04	247.89	237.55	245.33	267.71	267.71
02/11/04	253.98	247.96	252.15	266.42	266.42
02/12/04	254.68	246.75	253.03	272.60	272.60
02/13/04	254.59	248.29	252.55	266.69	266.69
02/14/04	262.35	256.94	259.45	282.42	282.42
02/15/04	256.41	250.58	256.43	270.07	270.07
02/16/04	256.46	248.50	256.95	265.66	265.66
02/17/04	256.73	247.34	256.32	270.59	270.59
02/18/04	252.14	244.07	249.90	266.22	266.22
02/19/04	253.13	246.87	253.71	264.87	264.87
02/20/04	250.83	243.26	244.38	270.81	270.81
02/21/04	256.81	256.27	252.30	264.39	264.39
02/22/04	254.83	253.47	254.08	263.59	263.59
02/23/04	258.27	251.77	256.24	275.64	275.64
02/24/04	259.50	256.27	254.82	276.89	276.89
02/25/04	252.52	246.91	247.29	267.84	267.84
02/26/04	246.89	246.56	245.30	250.36	250.36
02/27/04	243.98	240.79	241.94	249.46	249.46
02/28/04	254.07	246.89	249.99	270.75	270.75
02/29/04	257.48	252.96	256.60	272.69	272.69

4.2.3 Traffic

Table 19 shows the total inbound traffic of Cluster 2 measured during each day in February 2004.

TABLE 19
Traffic of Cluster 2

Reference date	Total traffic/day (Gbyte)			
	Cluster 2	HSP 1	HSP 2	HSP 3
02/01/04	4.158	3.787	4.198	4.488
02/02/04	7.566	7.052	8.095	7.552
02/03/04	7.346	7.568	7.059	7.412
02/04/04	7.241	7.302	6.575	7.844
02/05/04	7.160	6.925	7.050	7.505
02/06/04	7.483	7.031	7.978	7.440
02/07/04	4.582	4.001	4.269	5.476
02/08/04	3.648	3.303	3.801	3.840
02/09/04	7.243	7.187	7.635	6.906
02/10/04	7.418	6.823	7.754	7.676
02/11/04	6.979	7.322	6.619	6.996
02/12/04	7.650	7.267	7.972	7.712
02/13/04	7.050	6.759	6.995	7.396
02/14/04	4.660	4.348	4.131	5.500
02/15/04	3.729	3.339	3.849	3.998
02/16/04	7.438	6.639	7.681	7.993
02/17/04	7.295	6.600	7.337	7.947
02/18/04	6.911	6.977	6.817	6.939
02/19/04	7.137	6.618	7.192	7.600
02/20/04	7.158	6.240	7.570	7.666
02/21/04	4.698	4.521	5.250	4.324
02/22/04	3.558	3.146	3.899	3.629
02/23/04	4.873	4.442	5.420	4.758
02/24/04	4.212	3.920	4.059	4.658
02/25/04	6.351	5.895	7.072	6.086
02/26/04	7.020	6.205	7.200	7.654
02/27/04	7.452	7.069	7.971	7.317
02/28/04	4.309	3.999	4.045	4.882
02/29/04	3.961	3.638	4.733	3.513

Table 20 shows the total inbound traffic of Cluster 3 measured each day of February 2004.

TABLE 20
Traffic of Cluster 3

Reference date	Total traffic/day (Gbyte)				
	Cluster 3	HSP 1	HSP 2	HSP 3	HSP 4
02/01/04	3.106	3.822	2.920	3.677	2.004
02/02/04	6.159	7.339	5.982	7.229	4.084
02/03/04	5.411	6.684	4.441	7.009	3.510
02/04/04	6.334	7.370	6.105	7.530	4.330
02/05/04	5.926	7.060	6.002	6.903	3.739
02/06/04	3.738	3.769	4.280	4.239	2.663
02/07/04	2.642	3.374	2.677	2.633	1.885
02/08/04	4.649	5.895	4.378	5.769	2.555
02/09/04	6.317	7.440	5.840	7.537	4.452
02/10/04	5.403	6.229	5.175	6.431	3.777
02/11/04	6.042	6.971	5.171	7.459	4.566
02/12/04	5.797	6.449	5.532	6.927	4.279
02/13/04	3.757	3.637	4.035	4.730	2.624
02/14/04	3.052	2.790	2.984	4.229	2.208
02/15/04	5.865	6.723	5.203	7.058	4.477
02/16/04	5.052	6.445	3.839	5.812	4.113
02/17/04	5.983	6.939	5.309	7.486	4.199
02/18/04	5.730	6.297	4.864	7.332	4.428
02/19/04	5.996	7.087	5.427	7.231	4.238
02/20/04	3.948	3.699	3.919	4.534	3.640
02/21/04	2.792	3.269	2.391	2.935	2.574
02/22/04	4.155	4.041	4.269	4.330	3.978
02/23/04	3.146	3.163	3.172	3.530	2.721
02/24/04	5.264	5.228	5.360	6.264	4.204
02/25/04	6.081	6.896	5.658	6.630	5.142
02/26/04	6.202	6.964	6.342	7.218	4.282
02/27/04	3.707	3.528	4.133	4.308	2.860
02/28/04	3.167	3.193	2.918	3.857	2.702

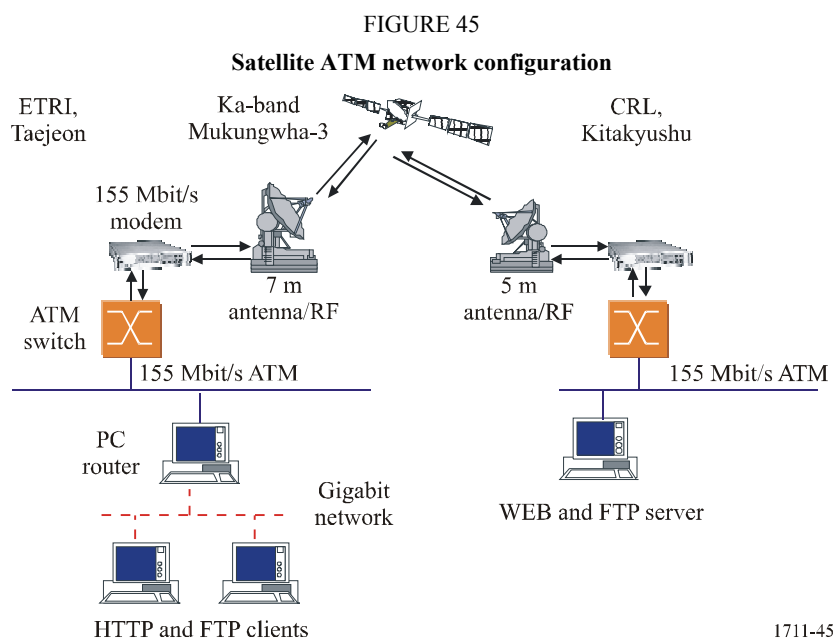
5 Application protocol measurements (FTP and HTTP)

ETRI in Korea and CRL in Japan started the Korea-Japan joint high data rate Satellite Communication Experiment in 2000. Two typical TCP applications – FTP and HTTP – were tested over ATM-based satellite network. The transmission of FTP and HTTP was done via Ka-band MUKUNGWHA-3 satellite.

Section 5.1 describes the network configuration for the experiment. Section 5.2 presents the experimental results of the tests conducted with FTP. Section 5.3 briefly introduces HTTP and presents the experimental results of the tests conducted with HTTP. Section 5.4 summarizes the conclusions.

5.1 Satellite ATM network configuration

Figure 45 shows the configurations of Korea-Japan high-speed satellite ATM network. For this joint experiment, two earth stations with 7 m antenna at ETRI, Korea and 5 m antenna at CRL, Japan were installed respectively. For main specifications of the the Korea-Japan satellite ATM network, see section 3.2.



The FTP and HTTP server was installed at CRL using a Linux-based PC. The server was directly connected to the ATM network. At ETRI, two client PCs were installed using Windows 2000 and Linux respectively. They were connected to a PC router that had two network interfaces: ATM and gigabit Ethernet. A gigabit subnet and a server were interconnected through the ATM-based satellite network.

For the FTP experiment, NCFTP 3.0 was used as FTP client and WUFTPD 2.6.1 as server. WUFTPD enables to set the maximum TCP window size to the value of the operating system. NCFTP implements the TCP window scaling option.

For the HTTP experiment, Apache 1.3.12 was used as a web server running on Linux. The TCP window size was set to 10 Mbytes. In order to monitor the internal operation and the performance of HTTP 1.0 and HTTP 1.1, two web browsers were used: Netscape 4.77 Linux version for HTTP 1.0, W3C's Webbot 5.2.8 for HTTP 1.1. When web pages were retrieved by the client's

request, all transferred packets were captured at the client side using tcpdump and post-processed using a tcptrace HTTP module. Five typical web pages were used in the HTTP experiment and described in Table 21.

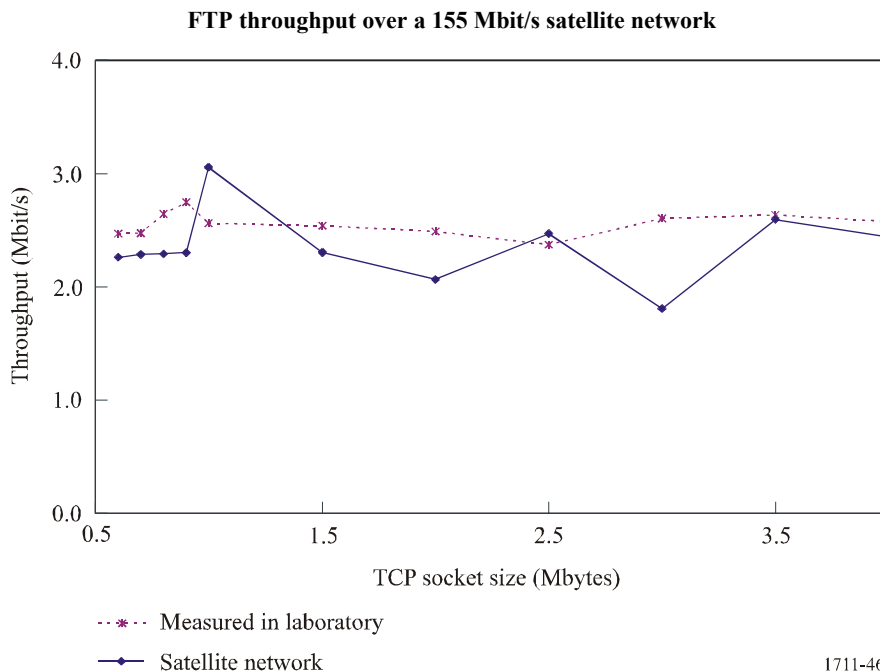
TABLE 21
Details of web pages

Web page	Number of elements	Page size (bytes)
China2008	30	212 207
CRL	21	80 333
FIFA	33	176 105
LionKing	16	393 672
RBLAB	8	72 103

5.2 FTP throughput over OC-3 satellite link

The goal was to measure the throughput of an FTP connection over an ATM-based satellite link. For the ease of comparison, the FTP throughput over a 155 Mbit/s link without satellite delay was also measured: a throughput of 118.32 Mbit/s was obtained with a TCP socket size of 64 kbytes (i.e. 87.5% of the theoretical throughput). Assuming a file size of about 92.1 Mbytes, the FTP throughput changing TCP socket buffer size was measured. Figure 46 shows FTP throughput using both a simulated and a real satellite link.

FIGURE 46



In the case of FTP, many factors affect the throughput performance such as CPU utilization, disk I/O and internal memory allocation for the network drivers and disk drivers. Figures 47 and 48 show TCP time sequence and TCP congestion window graphs respectively for a TCP socket buffer size of 1 Mbyte. For the first 38 s, the file transfer runs normally, afterwards there were some data

losses requiring retransmissions (since TCP recognizes data losses as due to network congestion). In Fig. 48, TCP congestion mechanism reduces the window size to half. Figure 47 shows another slow start after 38 s. As a result the overall throughput is severely degraded. Therefore for the normal operation of FTP with large TCP socket buffer, other system parameters and resources such as memory allocation for disk I/O and network driver interruption should be configured.

FIGURE 47
TCP time sequence graph
 (TCP buffer size = 1 Mbyte)

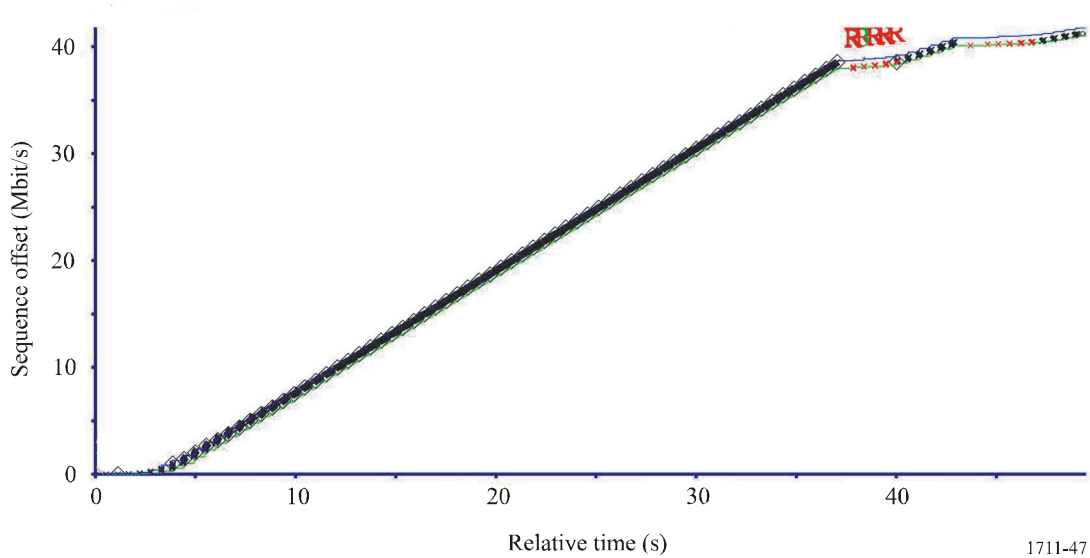
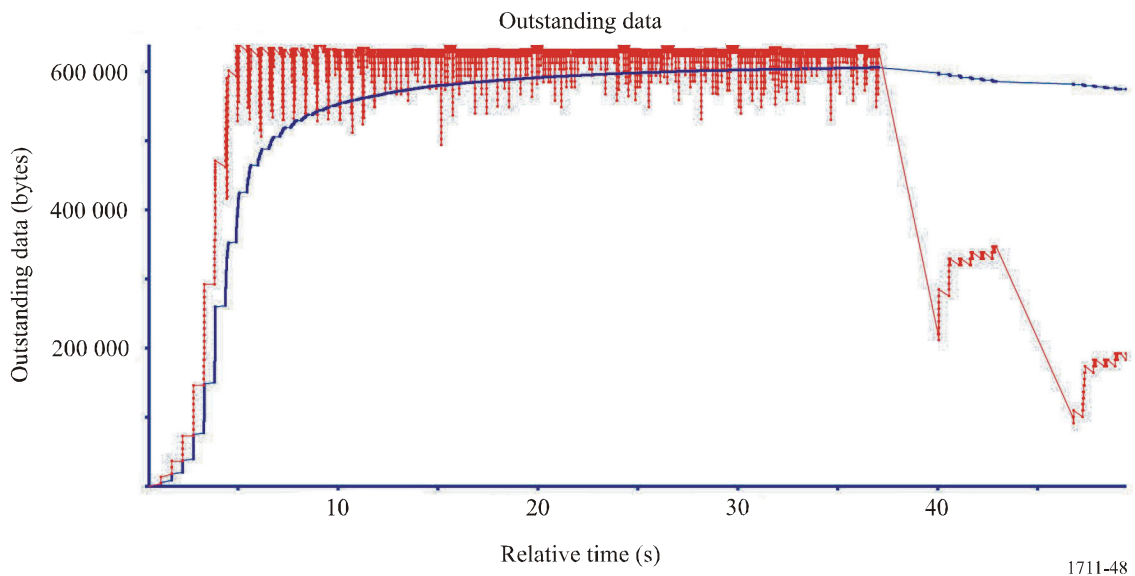


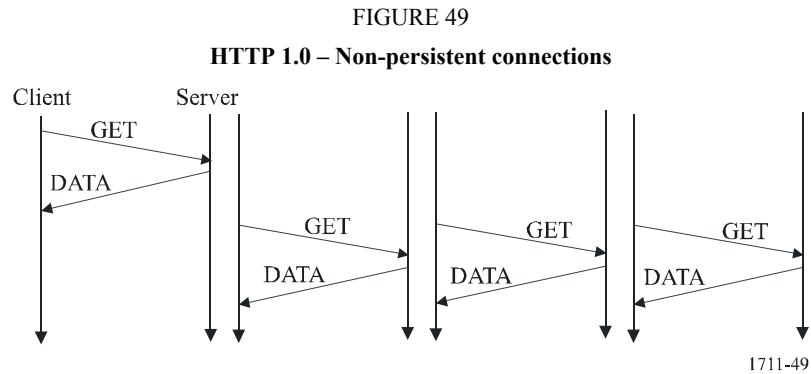
FIGURE 48
TCP congestion window graph
 (TCP buffer size = 1 Mbyte)



5.3 HTTP throughput over OC-3 satellite link

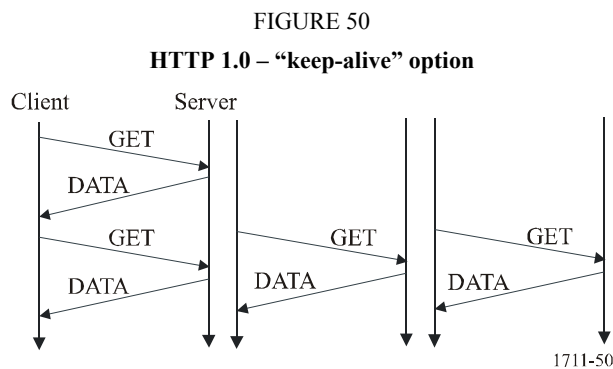
5.3.1 HTTP 1.0 with non-persistent connections

In HTTP 1.0, to download a complete web page, a separate TCP connection is required to retrieve each HTTP objects contained in a web page. Figure 49 shows the interactions between HTTP 1.0 client and server when a web page includes three objects. The base HTML page is first transferred via a TCP connection. Afterwards the TCP connection is closed and three new TCP connections are simultaneously established for the parallel download of the linked three objects. This may be inefficient as multiple simultaneous TCP connections burden the network.



5.3.2 HTTP 1.0 with “keep-alive” option

Some browsers and servers using HTTP 1.0 support the “keep-alive” option to overcome the inefficiency described above. This method uses one TCP connection to carry multiple HTTP requests. However browsers implementing this option can still establish multiple TCP connections. Figure 50 shows the operation of HTTP connection with the “keep-alive” option. The base document and one of the three objects are transferred through the first TCP connection. The other two objects are then transferred via two new TCP connections.

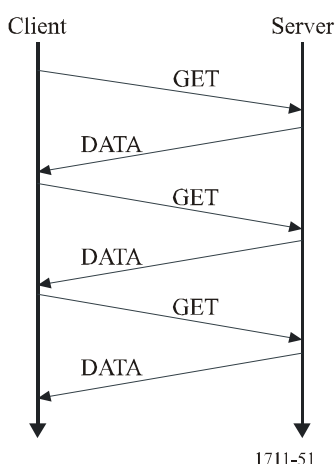


5.3.3 HTTP 1.1 without pipelining

The “keep-alive” extension, a form of persistent connection, was formally defined in HTTP 1.1. Persistent connections allow multiple requests. Responses can be contained in a single TCP connection and do not require multiple TCP connections. The performance of HTTP with persistent connections is improved because it bypasses the multiple slow start phases that would otherwise occur. Figure 51 shows the mechanism of HTTP 1.1 with persistent connection. The different objects are transferred in series. In the case of a base HTML document and three objects, it only takes four RTTs without pipelining.

FIGURE 51

HTTP 1.1 without pipelining



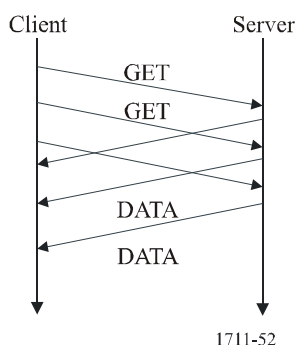
5.3.4 HTTP 1.1 with pipelining

HTTP 1.1 with pipelining allows multiple requests to be sent without waiting for a response. The pipelining can be used to avoid many round trip delays and improve performance because it eliminates the idle time between consecutive object retrievals.

Figure 52 shows the interactions between server and client using HTTP 1.1 with pipelining. A base document and three objects are transferred through a single TCP connection.

FIGURE 52

HTTP 1.1 with pipelining



5.3.5 Test results

The main goal was to measure the performance of Web page retrieval over satellite network using several HTTP versions. Table 22 summarizes the results of HTTP transfer over the satellite network for the five reference web pages. When webbot is used, only one TCP connection was established (HTTP 1.1 only needs one TCP connection). When Netscape was used, the number of TCP connections that were established corresponds to the number of elements linked to the web page. In the case of HTTP 1.0, each TCP connection is independent of the other ones. That is, each TCP connection performs slow start and congestion avoidance mechanism. When HTTP 1.0 was used, more packets were generated to transfer web page and linked elements. The total response time was less than with HTTP 1.1 without pipelining option. This means that, in long delay network, if there is no network congestion, multiple simultaneous TCP connections may be more effective than a

single one (especially when the size of the elements is small). However there are many negative aspects (e.g. burdens to server, network congestion due to more packets) of using multiple concurrent connections.

TABLE 22
HTTP transfer performance

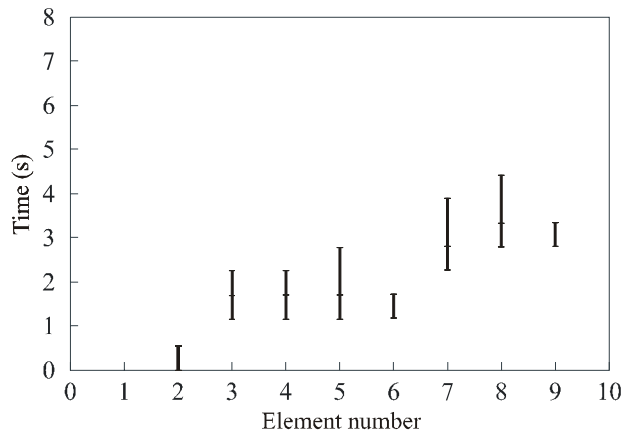
Web page	Web browser	Number of TCP connections	Number of packets	Total response time (s)	Average throughput (bit/s)
China2008 (30 objects, 212 207 bytes)	Netscape 4.77	41	655	14.764	14 373
	Webbot without pipeline	1	306	21.158	10 030
	Webbot with pipeline	1	318	4.363	48 638
CRL (21 objects, 80 333 bytes)	Netscape 4.77	22	307	8.642	9 296
	Webbot without pipeline	1	133	13.547	5 930
	Webbot with pipeline	1	137	3.247	24 741
FIFA (33 objects, 176 105 bytes)	Netscape 4.77	34	551	13.054	13 491
	Webbot without pipeline	1	282	21.682	8 122
	Webbot with pipeline	1	285	4.328	40 690
LionKing (16 objects, 393 672 bytes)	Netscape 4.77	14	660	8.277	47 562
	Webbot without pipeline	1	514	12.529	31 421
	Webbot with pipeline	1	564	4.882	80 637
RBLAB (8 objects, 72 103 bytes)	Netscape 4.77	8	166	4.365	16 518
	Webbot without pipeline	1	104	6.540	11 025
	Webbot with pipeline	1	119	3.822	18 865

When the request for a web page is made, the browser issues an HTTP GET command for the base HTML document. One RTT later, the base document will be received. Then the browser issues further GET commands for each element linked in the base document. With pipelining option of HTTP 1.1, these GET commands can be generated as soon as the reference is received by the browser without waiting for the current data transfer from the server to be completed. In the case of HTTP 1.0, separate TCP connections are established for the transfer of each element.

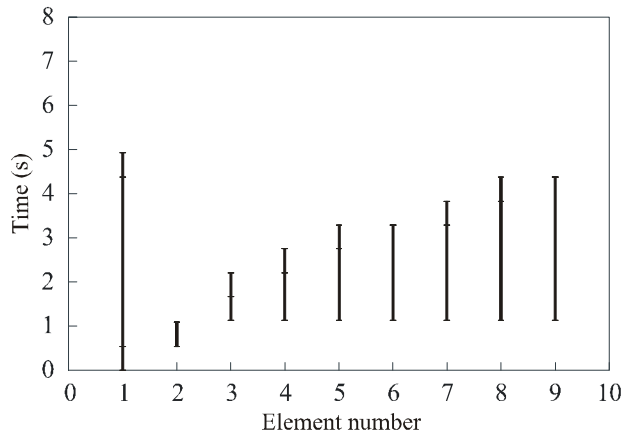
Figure 53 shows the sequence of element retrieval request and transfer for RBLAB page (seven elements). Item 1 in Figs. 53 b) and 53 c) represent the time for the entire transfer of the base page and linked objects. The other items depict the time for the transfer of each object. Item 2 is the first document from the web server at the request of the browser. Its transfer duration is the same regardless of HTTP version or options. However the following items have different transfer start times and durations depending on HTTP version and options. In the case of HTTP 1.0 (see Fig. 53 a)), when the base documents are received, the browser requests multiple GET for the objects linked to the base page. Therefore a number of TCP connections are established through three-way handshaking with a different connection request for each element. When the RBLAB page is loaded through HTTP 1.1 with pipelining option, the transfer of the following objects starts as soon as the base element is received. Without the pipelining option, the transfer of other objects can not start until the transfer of previous objects is completed. HTTP 1.1 establishes only one TCP connection hence triggering only one slow start.

FIGURE 53

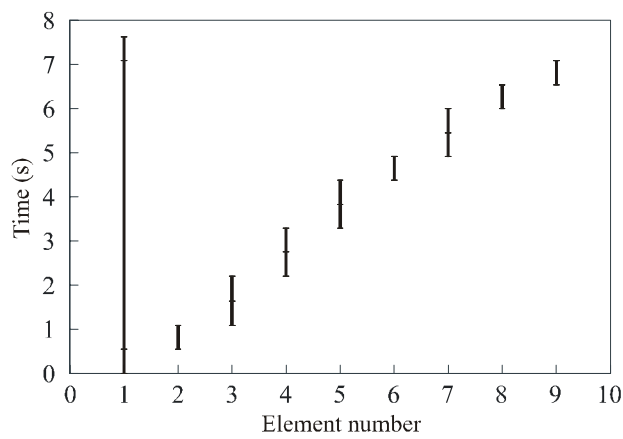
RBLAB web page and its elements transfer sequence graph



a) HTTP 1.0 no "keep-alive" (netscape)



b) HTTP 1.1 with pipelining (webbot)



c) HTTP 1.1 without pipelining (webbot)

When the pipelining option is active, several elements are transferred in the same connection appearing as a single bulk transfer. Many experiments show that bulk data transfers have good performance in LFN such as with GSO satellite networks. Therefore it seems that HTTP 1.1 with the pipelining option provides the best performance.

5.4 Conclusions

The maximum FTP throughput was about 3 Mbit/s with a TCP socket size of 1 Mbytes. With a TCP socket size of more than 1 Mbytes, the throughput is degraded. In the case of memory-to-memory transfer over satellite channel, the throughput mainly depends on the TCP window size. Increasing the TCP buffer size to improve TCP throughput may degrade the FTP performance by affecting disk I/O or system memory allocations.

As a result of several HTTP throughput measurements, it was found that HTTP 1.1 with pipelining option provided the best performance.
