Delta frames can be decoded after decoding the preceding frame; if such a frame is also a delta frame, then the other preceding frame must also be decoded first, and so on. A key frame is necessary to decode the succeeding delta frames one by one. For example, in Figure 4.19, frame 8 is a delta frame; to decode it, it is necessary to first decode frame 7, but frame 7 is a delta frame too, and to decode it it is necessary to first decode frame 6, and so on; frames 1–7 must be decoded before decoding frame 8. To avoid decoding many delta frames, it is possible to insert a higher number of key frames, but this Increases the total bit rate.

4.4 TCP/IP-based Protocols over Satellite Systems: A Telecommunication Issue

Mario Marchese

4.4.1 Introduction

This section presents the problems, metrics and the solutions of TCP/IP-based applications over satellite networks. The topic will be introduced with examples and descriptions that should help and simplify understanding. The problem is approached simply, to make the issue easy to understand also for non-experts in the field. The characteristics of the channels, algorithms and control schemes are described. The reference technology is represented by Geostationary Orbit (GEO) satellites, because many tests have been performed by the author using these, and many real measures can be reported to show the actual effect of decisions taken. It will be also shown that the issue is not limited to GEO satellites. The issue, as well as the methodology and metrics introduced, can also be applied in other environments, each characterized by a peculiar characteristic: the large delay per bandwidth product.

Unfortunately, a popular transport layer protocol such as TCP, and in particular its flow control, does not match the characteristics of a network where the product between the available bandwidth and the time that information requires to get to the destination, and to have confirmation of arrival, is large. Actually, the problem is hidden inside the TCP flow control. On the other hand, a large delay-bandwidth product characterizes not only GEO satellites, but also many other environments, such as broadband radio networks.

The section divided into five parts to introduce readers to the problems of TCP over GEO satellites (or, more generally, over large delay per bandwidth networks), to provide a framework to classify the solutions, and to give a methodology with which to approach the problem within this environment. A large, if not exhaustive, list of references is given to allow a deeper investigation by the reader. In reading this section it should be remembered that much of the material included in the book is part of work in progress, so the author will be grateful for any suggestions from readers – both strictly related to the material in the book and, more generally concerning the research topic, which is very hot, and also from an industrial viewpoint – that may be suitable for a research theme within the framework of a thesis for a masters degree or a Ph-D.

Section 4.4.2 states the motivations for using the TCP/IP protocol stack over satellite channels. It answers the simple questions: why should TCP be used? Why are communications over satellite increasing in importance? Why should TCP/IP protocols be applied over satellite channels? The section summarizes the aim of the TCP transport protocol, and the great number of applications that use it to ensure a reliable transfer of information; then it describes the advantages of using satellite links, including some examples of commercial enterprises, and utilizing TCP/IP, which is so widespread over the Internet. The importance of the performance metrics, related to the applications to be used is highlighted.

The problems that arise if TCP is used at the transport layer are presented in section 4.4.3. They are mainly linked to TCP flow control, which is explained in detail. Simple examples to make the concepts understood are reported, as well as some data really measured in the field. The aim of the section is to identify the problem and to understand the motivations that cause the low performance of TCP.

Section 4.4.4 presents a possible framework to describe the solutions. It is aimed at providing a classification of the state-of-the-art. Many solutions are reported in the literature, and it is not so simple to navigate among them, in particular at the start of research activity. The introduction of a general framework and a simple classification may be of help, both for beginners and for expert scientists taking their first steps in the field and looking for a specific solution. The section introduces two approaches: the 'black box' approach, whose description is the object of Section 4.4.5, and the 'complete knowledge approach', contained in Section 4.4.6.

Section 4.4.5 contains a study within the 'black box' approach, where the strong link with the metrics used is demonstrated. The presentation is a practical, 'in-the-field' description: the personal experience of the author within a three-year-long project dedicated to the issue is reported. It allows us to describe the growth of knowledge about the problem by using a real satellite test-bed, and to isolate the role of the parameters and the algorithm that are important in improving the performance.

The approach presented in Section 4.4.5 has inherent limitations. The 'complete knowledge' approach described in Section 4.4.6 allows us to design a new solution which overcomes the difficulties of the black box approach, using only its positive aspects. The section contains a description of new protocol architecture, and some preliminary results to envisage the real possibilities offered by the new protocol stack and by its implementation. The reader is introduced to a current research topic in strict connection with the ETSI (European Telecommunication Standard Institute).

4.4.2 Motivation

4.4.2.1 The TCP Protocol

The Transmission Control Protocol (TCP) is a connection-oriented, end-to-end reliable transport protocol working between hosts in packet-switched networks, and between interconnected systems of such networks. The motivations, philosophy and functional specification of the protocol are contained in RFC 793 [43]. Some of the material contained in it is summarized in the following to identify the aims and the scope of the TCP.

The reference protocol stack where TCP fits is shown in Figure 4.20. TCP assumes it can obtain a simple, potentially unreliable service from the lower level protocols and, in principle, TCP should be able to operate above a wide spectrum of communication systems, ranging from hard-wired connections to wireless and satellite networks. Nevertheless, using it within a satellite environment, even if it does not affect the correct working of the protocol, affects the performance, as will be shown. It is set just above the Internet

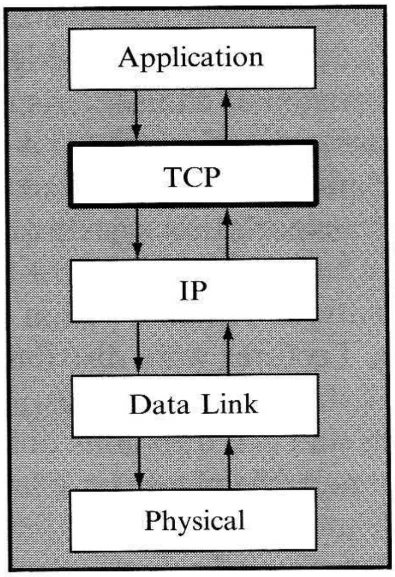

Figure 4.20 Reference protocol stack

Protocol (IP) [greatly], which offers a service to the TCP to send and receive information segments of variable lengths, called datagrams in IP terminology. Being a network layer, the IP also manages issues such as addressing and routing of the information. It may fragment TCP segments, which have to traverse portions of networks with different characteristics at the data link layer.

As noted above, the primary purpose of the TCP is to provide a reliable service between pairs of host computers (actually, their processes running on the hosts, because TCP is supposed to be a module within an operating system, but the term 'host' simplifies comprehension). To match the requirements on top of a less reliable Internet communication, it uses facilities in the following areas: basic data transfer, reliability, flow control, multiplexing, connections, precedence and security. A detailed description of each operation, which can be found in RFC 793 [43], is out side the scope of this section. Nevertheless, it is important to focus on two areas whose implementation has a strong impact on the metrics used to measure performance when TCP is applied over satellite links: reliability and flow control. TCP provides a means to rule the amount of data sent by the sender, applying an Automatic Repeat Request (ARQ) system. It assigns a sequence number to each segment, and uses an acknowledgement mechanism that flows from the destination to the source to be sure the information has arrived at the destination (more exactly, that it has arrived at the TCP process by the destination host computer). Conceptually, each segment is assigned a sequence number. The sequence number of the segment is transmitted with the segment itself. When the TCP transmits a segment, it puts a copy on a retransmission queue and starts a timer, called a Retransmission Timeout (RTO). When the acknowledgment for that segment is received, the segment is deleted from the queue. If the acknowledgment is not received before the timer runs out, the segment is retransmitted. The flow control mechanism employed is explained in detail in the next section. In general, TCP uses a 'window', which specifies the amount of data that can be sent.

4.4.2.2 *Applications*

Which applications, identified with the corresponding block in Figure 4.20 use TCP? The answer is simple: all the applications that require a secure and reliable non-delay-sensitive transport service. Anyway, this answer does not give us any idea about the number of applications that use TCP. Some of them are: Web navigation, database access to retrieve information, tele-medicine (transmission of clinical tests, x-rays, electrocardiograms, magnetic resonance), tele-control (remote control of robots in hazardous environments, remote sensors, systems for tele-manipulation), bank and financial operations, e-commerce for home business, e-commerce for transportation systems, goods movements, purchase and delivery, and tele-learning.

The latter deserves special attention. The first vision of tele-education was essentially based on non-interactive services. Lectures were distributed through videocassettes, CDs and also special TV channels. The students had no possibility to interact with the teacher; they could not make interventions and ask questions online. The possibility of asking for explanations was relegated to the mail, the phone or, when available, e-mail. The few tele-learning systems which included interaction, even if limited, presented too many drawbacks, such as the low number of sites, the high cost of the bandwidth and a service composed only of audio and video (voice and video of the teacher, along with a video camera showing a blackboard). The reference technology was ISDN, for private networks, or television. Dedicated technology such as video cameras and coders/decoders was used.

A first step in evolution was due to the diffusion of the Internet. Local Area Networks (LANs), located remotely (e.g. in different universities), can be connected through the Internet. The great advantage is the possibility of using TCP/IP-based applications, often already available on the market, to send audio/video, to prepare support material, documentation and presentations. The students may receive the material directly in their computer, both at the campus and at home. The teacher can give the lecture directly from his office, without using a special tele-teaching classroom, even if the presence of this type of classroom is always a guarantee of audio/video quality because it uses special tools as mixers. The concept of tele-education is not limited to audio and video, which are delay-sensitive non-TCP based services (they apply UDP, a protocol of the TCP/IP family, which offers an unreliable transport service). It also includes the presence of data, which can be utilized both online (e.g. the presentation of the teacher, explanations, support material that can be sent as a file) and offline (e.g. access to an educational data-base where books, papers and other material that can integrate the content of the lectures may be stored). The drawback of this approach is represented by the Internet technology available, which, on the one hand, offers the opportunities mentioned, but on the other, has a limited bandwidth and does not implement any algorithm to reserve bandwidth and guarantee a fixed level of quality to users. The effect is the limited possibility to have actual interactive services and the spreading of audio/video streaming services. In practice, the opportunities given by full TCP/IP internetworking cannot be taken because of the lack of QoS guarantees.

The last step of this evolution is the integration of different technological platforms, including satellites, along with the wide range of services offered. The integration of LANs located by the sites interested, ISDN, ATM and satellite networks, along with the introduction of new technologies and algorithms, has allowed a real interaction, a high number of sites involved and the application of a new vision of tele-education. Among the new technologies and algorithms, it is worth mentioning: the large availability of bandwidth, the implementation of new bandwidth reservation mechanisms in IP networks (integrated and differentiated services), modification of the transport layer (the object of this section), multicast protocols, multicast applications, and high quality data/audio/video applications. They allow an improvement of the performance of file transfers, and to reserve bandwidth for specific flows, to protect the 'most important' information from congestion, such as the teacher's audio and data. The result is a tele-education system that can efficiently reach remote users, can traverse portions of networks based on different technologies, and that it is based on audio-video interaction, guaranteed QoS and utilization of didactic Webs containing text, audio, video, images, online and off-line lectures.

The tele-teaching application is of special relevance because it allows us, on the one hand, to show a new environment where TCP is applied and, on the other, to introduce the importance of satellite networks.

4.4.2.3 Advantage of Satellite Networks

Satellites offer clear advantages with respect to cable networks [44]:

- *The architecture is scalable*: a new user can join a satellite communication by acquiring the necessary technical instrument, and no area has to be cabled to get high-speed services. Cabling is not a simple job, and adding a remote customer to the network is not always possible without technical complications. If a new customer wants to join a satellite network, he only needs to acquire the necessary tools. An example from personal experience may help. The Italian National Consortium for Telecommunications (CNIT), the association composed of universities and scientific laboratories where the author works, has a small private satellite network, mainly dedicated to offering videoconferencing and tele-education services to universities and research laboratories, members of the CNIT. The satellite network is a portion of the overall CNIT network, composed also of high-speed terrestrial links. The problem of adding a new university to the network is simply solved when the connection is performed through satellite by installing an antenna, and a base station, along with a router to guarantee the interconnection with LANs. There is no problem of scalability.
- *Diffusion throughout the land is wide*: a satellite network overcomes geographical obstacles which would make the installation of a cable network of equivalent quality difficult; moreover, satellites can cover isolated areas. It is sufficient to consider huge continents such as Australia, Africa, America or countries in Asia and South America, characterized by areas where the population density is so low that cabling could not be sustained economically. Other geographic obstacles characterize some regions: mountains, valleys and rivers, where either it is extremely difficult to offer a cable service, or it is in convenient from an economic viewpoint. The installation of a telecommunication network is made difficult in many regions of the world by natural disasters such as floods and hurricanes, or by wars. In these cases, the only possibility to guarantee telecommunications is represented by satellites.
- *The bandwidth availability*: satellite bandwidth, in particular in the Ka-band, which is the object of many experiments and also the object of the tests reported in this section is less affected by congestion than terrestrial networks, where, even if the bandwidth availability is high, the number of potential users is huge.
- *The multicast service is very simple*, satellite inherently being a broadcasting tool.
- *Satellite links are often private lines*: the Internet is characterized by heterogeneity from the point of view both of algorithms and management, which is performed by many organizations and providers. A completely private network has the advantage of being managed by few people, thus avoiding many problems regarding the property and management of different portions of the network.

4.4.2.4 TCP versus Satellite Networks

Matching the applications that use TCP with the advantages offered by satellites, it is natural to think of TCP/IP-based applications over satellite networks. However, the TCP/IP protocol family is not so suited for the satellite environment (the motivations should be made clear in the following), but on the other hand, the diffusion of TCP/IP applications makes it difficult to think of another protocol architecture, non-transparent to the user, dedicated to satellite links. A solution that allows us to efficiently transport TCP/IP applications through satellite networks transparently to the final user should be the clue.

Some more information about the commercial interest of using satellites (and, in particular, Geostationary Orbit (GEO) satellites) in modern telecommunications should help to clarify the situation.

Some satellite operators invest spatial resources to operate in markets with a great growth potential, such as Latin America. A GEO satellite may also have a key position to guarantee an intercontinental backbone network, and a satellite may cover from the East Coast of the USA and Latin America, to Europe, Middle East and Central Asia.

The Internet is not sufficient to support a telecommunications service in the case of problems, and also in countries where there is a large telecommunication infrastructure and many applications are particularly important. For instance, people may require news on the Web just in difficult moments, and the traffic bottlenecks of terrestrial networks and of traditional links could make the service highly inefficient and, in some cases, no longer useful. Web and downloading services over satellite are thought to be an important issue for the future, and TCP is the transport layer used for these applications. To build a broadband digital transmission system for video and data oriented to business and home users is very promising, in particular, but only for geographical areas not covered by a widespread cable telecommunication network.

Applications over future satellites should not be limited to the Internet (or to the present Internet), but should also include new environments. The competitive advantage is both the interactivity and the possibility of building networks and services adapted to different needs. From tele-learning to managing the activity of public administration, from bank and financial services to industrial activity located remotely. The latter is very relevant. Many industries have peripheral offices in East Europe and the Far East, and have the need to guarantee the continuity of the production processes, i.e. they have to create a direct line among the headquarters and the remote offices, where the telecommunication infrastructure is often limited or not so reliable. As a consequence, small 'light', specifically dedicated networks may be built to join the main site with the peripheral units. The connection speed and their duration may be determined by the different needs. Some applications may require a 24-hour connection, others only a limited time-window-based connection.

So, commercial offering include broadcasting, backbone access, international connection, tele-medicine, tele-education, data recovery, video-surveillance, connectivity through rural networks, consulting, control of the land, Earth observation, as well as satellite applications applied to car technology.

Tele-education deserves special attention. Many students (71% of American students aged between 12 and 17 years, for example) download material from the Internet for their studies. Many students live in remote areas and need a long time to get to school. Others would prefer to stay at home to prepare for exams and tests. From the technological point of view, small but powerful and less expensive antennas (90 cm) can be installed on the roofs, and Ka-band (20–30 GHz) satellites can be dedicated to broadband services, for home users. A possible satellite architecture, derived from a project to provide a tele-learning service, is described in the following.

The project is aimed at designing, implementing and experimenting with interactive and non-interactive services based on Web access towards remote sites connected through a satellite network. Protocols, topologies and applications will characterize the satellite network, properly designed so as to support Quality of Service (QoS) guaranteed data, audio and video services.

The objectives are:

- Creation of an international network to store and broadcast data and information mainly dedicated to tele-learning.
- Creation of a multimedia library concerning tele-learning, which can be accessed through a satellite network by a Web interface.
- Creation and utilization of data, audio and video transmission tools, which allow remote utilization for
  - tele-learning
  - video-conferencing.

Including applications such as

- Tele-control and access to remote sensors,
- Remote technical assistance to be applied in difficult to reach areas (tele-operability),
- Tele-medicine.

The telecommunication infrastructure shown in Figure 4.21 is composed of a fully connected network, which is the core of the telecommunication system, and by a star shaped network, which guarantees the low cost interconnection of a large number of users. The aim is the implementation of a telecommunication system covering a wide area, dedicated to the applications mentioned above. The most advanced telecommunication devices available on the market will be used, along with components, protocols and tools designed to improve the system performance and the quality perceived by the users.

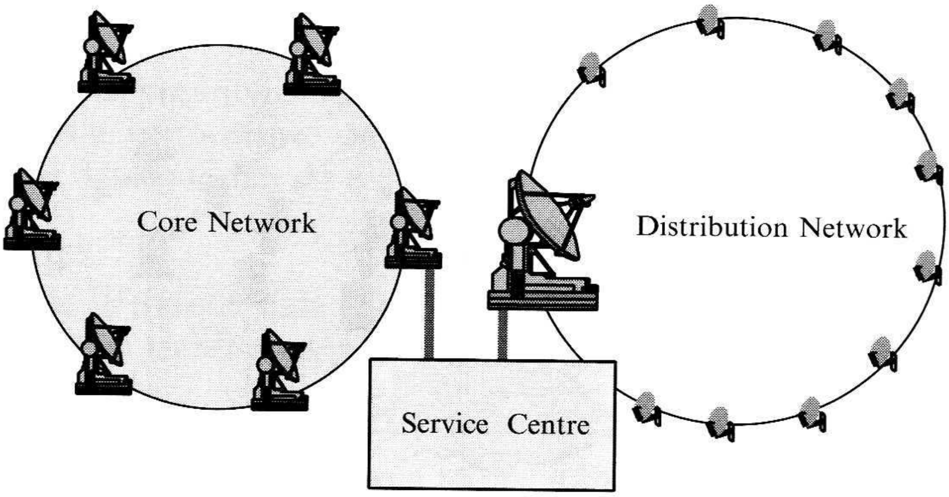

Figure 4.21 Configuration of the network

In more detail, the technological network is formed by three components:

- A fully connected satellite network (core Network) in the Ka-band (20–30 GHz), to connect the main sites, characterized by high interoperability, using tools such as videoconferencing, tele-working and multimedia data exchange at high speed.
- A distribution network (satellite or heterogeneous) with a star topology, composed only of the service users, who may be residential users, companies, public and private institutions such as schools, teaching centers and so on. The user station is bidirectional connected to the core network.
- A Service Center, which is an interconnection node between the two networks, and is aimed at storing and distributing the multimedia data and contents flowing from the core network to the distribution network.

The contents may be stored in each of the main sites, or localized in the service center. Each main site can collect the information flowing from the other main sites, send data to the terminal users, and provide a service through a Web interface both to the main sites and the terminal users.

The terminals receive the data via satellite and assume an interactive role through a satellite network, if devices for the satellite transmission are located by the user terminals, or through, a terrestrial network through ISDN (or ADSL) technology. The architecture is reported in Figure 4.22.

On the basis of the descriptions reported previously, the services offered by the technical support may be summarized as follows:

- *Real-time audio-video transmission*: the service may be represented by an audio-video application followed by the other sites and/or by the terminal users, who may interact (for example, students may ask questions to the teacher or take part to a discussion

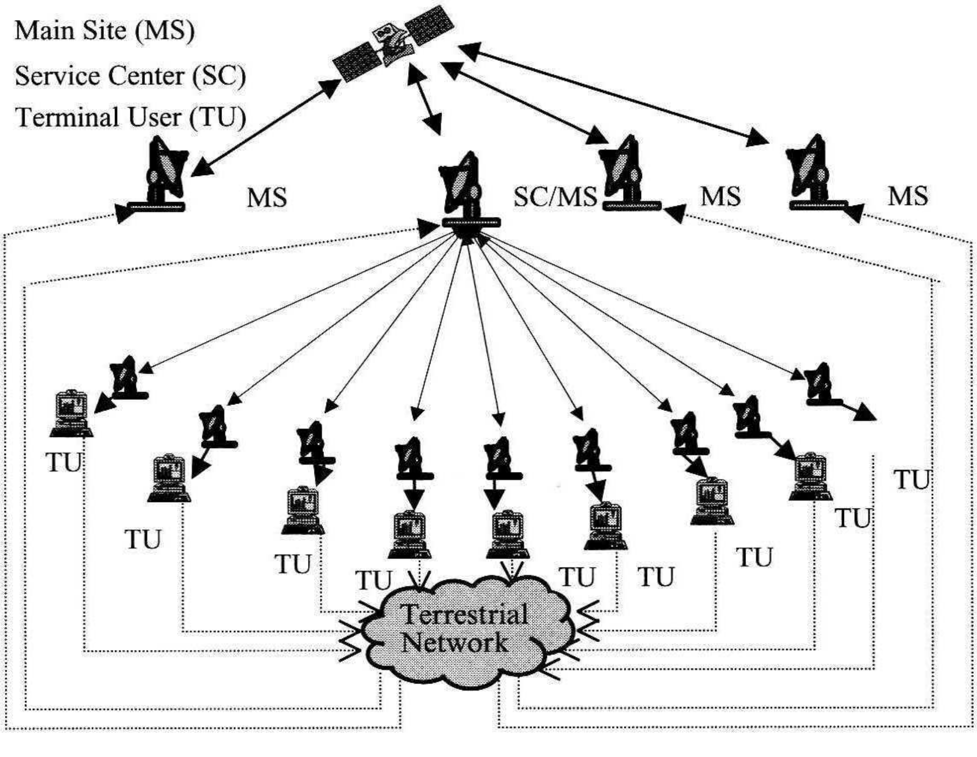

Figure 4.22 Network topological architecture

phase). Another possible application may be a sort of videoconference where the various partners may express opinions concerning a specific topic.

Referring to the topology shown in Figure 4.22, the service mentioned would concern the connection both among main sites (MS< - >MS) and among main sites and terminal users via the service center (MS< - >SC< - > TU).

- *Non real-time audio-video transmission*: the service is similar to the previous one, except for the real-time interaction. The main sites may, for instance, follow a previously recorded lecture, the teacher not being directly present. Some questions or comments might be sent, recorded and answered after a certain amount of time, e.g. with no real-time interaction.

In this case, the direct connection among main sites is useless. The service is provided both by one of the main sites, or by the service center which, for example, may store audio and video, operate distribution and, on the other side, record questions, comments and so on. From the technical point of view, the service is an asynchronous multicast transmission. Even if it is technically different, there is a strict connection with the following proposed service.

- *Web service*: the object is the provision to all users of an amount of information of interest via a Web form. Similarly, as it is possible to do using the Internet, users may find information about various topics, whose detailed content definition would be one of the objects of the project, in any form: written data, audio, video. The aim is to create the already mentioned distributed library, where, depending on the knowledge level, all the components of the project may find useful information about their work in a simple and user-friendly form. Moreover, the Web should make possible co-operative work among all the components, which could also fill some part of the Web site, thus having an active part in the creation.

The technology already available in the commercial world is sufficient to install the network and to offer the services mentioned above, but the performance is greatly affected by the inefficiency of the transport layer protocol.

4.4.2.5 Metrics

On the basis of the applications mentioned, it is important to define some metrics with which to measure the performance. Most applications have a common root: they need a quick and reliable data transfer. As a consequence, the reference metrics are, essentially, two:

- The overall time to transfer data (e.g. file), measured in seconds (s).
- The amount of data transferred in each time unit (the throughput), measured in bytes per second (bytes/s).

Actually, the two metrics may be reduced to one if the final throughput is considered, as they depend upon each other, but the distinction should be maintained because it helps to understand the difference among the different schemes that will be presented. For instance, even if a file transfer requires a similar time to be concluded by using two different algorithms (and, as a consequence, the final throughput for the two alternatives is almost the same), it may have a different throughput in some instances of the transmission. It may help us to understand the behavior and give information useful for specific applications. For example, if one algorithm is more efficient than the other in the initial phase of the connection, it may be applied in applications that require a short file transfer, such as telecontrol.

The metrics mentioned are 'objective' metrics of Quality of Service (QoS). They do not depend up on the user's opinion. It is very important to know, for modern applications, which is the subjective effect of the 'objective' performance on the real users, i.e. it is important 'to measure' the opinion of the users. An example may be of help. There are two alternative algorithms, as in the previous case: one of the them allows us to transfer a file in 15 seconds, while the other provides a transfer in 10 seconds, but it costs more than the first one. The latter is surely the best, but is the large cost convenient? Do the users really appreciate the difference? It is really important to know their impression and to define a metric to measure their opinion. It is called Perceived-Quality of Service (P-QoS). A technique such as the Mean Opinion Score (MOS) is used to assign a numerical value to the users' opinions. Operatively, each user is asked to express an opinion by giving a mark ranging from 1 to 5. The lowest (1) corresponds to a very bad perception, while the highest (5) to a very good quality.

The results reported in this section concern objective metrics. Nevertheless, their correspondence with user perception should not be neglected.

4.4.3 Problems with TCP at the Transport Layer

4.4.3.1 TCP Congestion Control

Most of this subsection is taken from RFC 2581 [45], which describes the algorithms ruling TCP behavior in the presence of congestion. In detail, it specifies four algorithms: slow start, congestion avoidance, fast retransmit and fast recovery. The definitions contained in Table 4.7 have been used.

A segment is considered lost either after the special timer (Retransmission Timeout – RTO) expires, as mentioned in the previous section and in RFC793[43], or after three duplicated acknowledgements (four ACKs indicating the same sequence number), as explained in the following.

The slow start and congestion avoidance algorithms are used by a TCP sender to control the amount of outstanding data being injected into the network. The minimum of cwnd and the minimum between the source buffer and rwnd governs data transmission (the variable TW identifies the real transmission window). Another state variable, the slow start threshold (ssthresh), is used to determine whether the slow start or congestion avoidance algorithm is used to control data transmission, as discussed below.

Some more indications about generating acknowledgements. A TCP receiver should use the delayed ACK algorithm, which means that an ACK is not generated every full-sized received segment, but in most implementations, every second full-sized segment, and in any case, it must be generated within 500 ms of the arrival of the first unacknowledged packet.

Out-of-order data segments should be acknowledged immediately, so as to accelerate loss recovery.

Table 4.7 Definition of TCP parameters

Parameter	Definition
Segment	Any TCP/IP data or acknowledgment packet (or both)
Sender Maximum Segment Size (SMSS)	Size of the largest segment that the sender can transmit. It depends on the type of network used, and on other factors
Receiver Maximum Segment Size (RMSS)	Size of the largest segment the receiver can accept
Full-sized segment	A segment that contains the maximum number of data bytes permitted (i.e. SMSS bytes of data)
Receiver window (rwnd)	The most recently advertised receiver window, and it is a receiver-side limit on the amount of outstanding data. It corresponds, at least in the implementations checked, to half of the receiver buffer length at the beginning of the transmission
Congestion window (cwnd)	A TCP state variable that limits the amount of data a TCP can send. It is a limit on the amount of data the sender can transmit into the network before receiving an acknowledgment (ACK). Some implementations maintain cwnd in units of bytes, while others use units of full-sized segments
Initial Window (IW)	Size of the sender's congestion window after the three-way handshake is completed
Loss Window (LW)	Size of the congestion window after a TCP sender detects loss using its retransmission timer (see below)
Restart Window (RW)	Size of the congestion window after a TCP restarts transmission after an idle period
Flight size	The amount of data that has been sent but not yet acknowledged. Actually it identifies the segments still 'in flight' inside the network

Slow Start

The slow start algorithm aims to probe the network to check the available capacity and thus avoid congestion. IW, the initial value of cwnd, must be less than or equal to 2(SMSS bytes. A non-standard, experimental TCP extension allows the use of a larger window whose value is limited by Equation (4.48):

$$\text{IW} = \min(4 \cdot \text{SMSS},\ \max(2 \cdot \text{SMSS},\ 4380\ \text{bytes}) \tag{4.48}$$

The initial value of ssthresh may be arbitrarily high, and it is reduced in response to congestion. The slow start algorithm is used when cwnd < ssthresh.

During slow start, a TCP increases cwnd by, at most, SMSS bytes for each ACK received that acknowledges new data. The slow start phase ends when cwnd exceeds ssthresh, or when congestion is observed.

Congestion Avoidance

The congestion avoidance algorithm is used when cwnd > ssthresh. When cwnd and ssthresh are equal, the sender may use either slow start or congestion avoidance.

Congestion avoidance continues until congestion is detected. Within the congestion avoidance phase, cwnd is increased by one full-size segment after a number of ACKs corresponding to the value of cwnd/SMSS has arrived (each (cwnd/SMSS) ACKs → cwnd = cwnd + 1 · SMSS). One formula commonly used to update cwnd during congestion avoidance is given in Equation (4.49). It contains the adjustment executed on every incoming non-duplicate ACK:

$$\text{cwnd} = \text{cwnd} + \text{SMSS} \cdot \text{SMSS}/\ \text{cwnd} \tag{4.49}$$

The implementations that maintain cwnd in units of full-sized segments will find Equation (4.49) difficult to use, and should use another method to implement the general principle. Actually, the general principle of the congestion avoidance algorithm is that cwnd should be incremented by one full-sized segment per Round-Trip Time (RTT), which is defined as the time to get to the destination and back. Equation (4.49) allows approximation of the general indication. When a TCP sender detects segment loss using the retransmission timer, the value of ssthresh must be set to no more than the value given in Equation (4.50):

$$\text{ssthresh} = \max(\text{FligthSize}/2,\ 2 \cdot \text{SMSS}) \tag{4.50}$$

FlightSize is the amount of not yet acknowledged data in the network. It is important not to use cwnd rather than FlightSize. This mistake has characterized some TCP implementations in the past.

Furthermore, upon a timeout, RTO cwnd must be set to no more than the loss window, LW, which equals one full-sized segment (regardless of the value of IW). Therefore, after re-transmitting the dropped segment, the TCP sender uses the slow start algorithm to increase the window from one full-sized segment to the new value of ssthresh, at which point congestion avoidance again takes over.

Fast Retransmit/Fast Recovery

A TCP receiver should send an immediate duplicate ACK when an out-of-order segment arrives. The purpose of this ACK is to inform the sender that a segment was received out-of-order, and which sequence number is expected. From the sender's perspective, duplicate ACKs can be caused by a number of network problems (e.g. dropped segments, re-ordering of data, replication of data). Obviously, a TCP receiver will send an immediate ACK when the incoming segment fills in all or part of a gap in the sequence space. The TCP sender uses the 'fast retransmit' algorithm to detect and repair loss, based on incoming duplicate ACKs. The fast retransmit algorithm uses the arrival of three duplicate ACKs (four identical ACKs without the arrival of any other intervening packets) as an indication that a segment has been lost. After receiving three duplicate ACKs, TCP performs a retransmission of what appears to be the missing segment, without waiting for the retransmission timer to expire.

After the fast retransmit algorithm sends what appears to be the missing segment, the 'fast recovery' algorithm governs the transmission of new data until a non-duplicate ACK arrives. The reason for not performing slow start is that the receipt of the duplicate ACKs not only indicates that a segment has been lost, but also that other segments are most likely leaving the network. For instance, if three duplicated ACKs reach the source, it means that three segments have reached the destination.

The fast retransmit and fast recovery algorithms are usually implemented together as follows:

- When the third duplicate ACK is received, the ssthresh value is set to the value given in Equation (4.50)
- The lost segment is retransmitted and cwnd set to ssthresh plus 3(SMSS (as already said, three duplicated acknowledgements means that three segments have left the network).
- For each additional duplicate ACK received, cwnd is increased by SMSS.
- A segment is transmitted, if allowed by the new value of cwnd and the receiver's advertised window.
- When the next ACK arrives that acknowledges new data, cwnd is set to ssthresh (the value set in step 1).

TCP researchers have suggested a number of loss recovery algorithms improving fast retransmit and recovery. Some of them are based on the TCP selective acknowledgment (SACK) option [46], which allows exact specification of the sequence number of the missing segment.

Operatively and referring to a specific TCP implementation (a NewReno TCP under the 2.2.1 version of the Linux kernel), the TCP transmission begins with the slow start phase, where the congestion window (cwnd) is set to one segment ($IW = 1 \cdot SMSS$, in this implementation), and the slow start threshold (ssthresh) is set to a very high value (infinite). With each received acknowledgement (ACK), cwnd is increased by $1 \cdot SMSS$. If the value of cwnd is less than ssthresh, the system uses slow start. Otherwise, the congestion avoidance phase is entered. More precisely, cwnd is increased by $1 \cdot SMSS$ after receiving a number 'cwnd' of acknowledgements. If there is a loss, a packet is considered lost after three ACKs that carry the same acknowledgement number (duplicated ACKs); the system enters the fast retransmit/fast recovery algorithm, and performs a retransmission of the missing segment, without waiting for the retransmission timer to expire. In some TCP versions, as already said, ssthresh was erroneously set to cwnd/2. In the implementation used here, ssthresh has been set to the maximum between FlightSize/2 and $2 \cdot SMSS$, where FlightSize is the measure (in bytes) of the amount of data sent but not yet acknowledged, i.e. the packets still in flight. The cwnd is set to ($ssthresh + 3 \cdot SMSS$). When the error is recovered (i.e. when the lost packets have been successfully retransmitted), the value of cwnd is set to ssthresh. The real transmission window (TW) is set, in any case, to the minimum between cwnd and the minimum between the TCP buffer dimension at the source and the receiver's advertised window (rwnd), which is half of the receiver TCP buffer length ($TW = \min\{cwnd, \min(sourcebuff, rwnd)\}$). The receiver window rwnd has been measured to be 32 kbytes at the beginning of the transmission. It corresponds to half of the receiver buffer space that is automatically set by the TCP to 64 kbytes. This numerical value is due to the TCP header (see [43] and Comer [47], for a detailed description) which uses a 16 bit field to report the receiver window size to the sender. Therefore, the largest window that can be used is 2^{16} bytes. To circumvent this problem, RFC 1323 [48] has defined a new TCP option, 'Window Scale', to allow the use of larger windows. This option defines an implicit scale factor, which is used to multiply the window size value found in a TCP header to obtain the true window size. The 'Window Scale' option is considered to be allowable in all the examples and results reported in the rest of this section.

In a more schematic way, the procedures listed above may be summarized as in Table 4.8; a C-like language is used for the description.

Table 4.8 TCP parameters

TW=min {cwnd, min(source buff, rwnd)}	
Slow Start	cwnd $= 1 \cdot$ SMSS; ssthr $= \infty$ ACK $\rightarrow$ cwnd $=$ cwnd $+ 1 \cdot$ SMSS
Congestion Avoidance	$<$ cwnd/SMSS $>$ ACKs $\rightarrow$ cwnd $=$ cwnd $+ 1 \cdot$ SMSS

Fast Retransmit/Recovery	ssthr $= \max\{$FlightSize$/2$, $2 \cdot$ SMSS$\}$;
	cwnd $=$ ssthr $+ 3 \cdot$ SMSS;
	Duplicated ACK $\rightarrow$ cwnd $=$ cwnd $+ 1 \cdot$ SMSS;
	cwnd $=$ ssthr

4.4.3.2 An example

An important quantity is the 'delay per bandwidth' product. As indicated in RFC 1323 [48], the TCP performance does not depend upon the transfer rate itself, but rather upon the product of the transfer rate and the Round-Trip Time (RTT). The 'bandwidth · delay product' measures the amount of data that would 'fill the pipe', i.e. the amount of unacknowledged data that TCP must handle to keep the pipeline full. TCP performance problems arise when the bandwidth · delay product is large. In more detail, within a geostationary large delay per bandwidth product environment, the acknowledgement mechanism described takes a long time to recover errors. The propagation delay makes the acknowledgement arrival slow, and cwnd needs more time than in cable networks to grow. If, for example, just one segment was sent, it takes at least one RTT to be confirmed. The throughput is very low, even in the slow start phase. This greatly affects the performance of the applications based on TCP. A simple example shows this.

The control mechanism is simplified. The Delayed ACK mechanism is not used, and an ACK is sent each full-size segment. No segment is lost and the slow start phase is never abandoned. In this simple example, the Transmission Window is considered limited not by the formula appearing in Table 4.8, but only by the value of cwnd. The example in the following is didactic, and no real measures have been taken. The behavior of the TCP (simplified as described above) when there is a Round Trip Time (RTT) of 100 ms is compared with the behavior when the RTT is 500 ms (the average RTT of a GEO satellite network). The value of the Transmission Window contains the number of SMSS [bytes] actually sent at the instant indicated in the first row. For example, if SMSS=1500 bytes, the real number of bytes sent at the beginning (instant 0) is $1 \cdot 1500$, for both cases. In the RTT=100 ms case, after receiving the first ACK (i.e. after 0.1 s), cwnd is augmented by one and two segments may be sent; after 1·RTT, two ACKs arrives substantially in the same time, and four segments are allowed to leave the source (instant 0.2), and so on. The behavior if RTT=500 ms is exactly the same, but at the time 0.2 the first ACK has not yet arrived and the second segment has not left the source. The result after 1 s is that 1024 SMSS may leave the source if the RTT equals 100 ms and only 4 SMSS if RTT=500 ms (Table 4.9).

The problem is the delay of the network or, in more detail, the delay per bandwidth product of the network, that has a devastating effect on the acknowledgement mechanism used by the TCP.

Table 4.9 Simplified TCP behavior

Time (s)	Transmission Window (SMSS) RTT=100 ms	Transmission Window (SMSS) RTT=500 ms
0	1	1
0.1	2	1
0.2	4	1
0.3	8	1
0.4	16	1
0.5	32	2
0.6	64	2
0.7	128	2
0.8	256	2
0.9	512	2
1	1024	4

Figure 4.23 contains the throughput (bytes/s) really measured on the field for a file transfer of 675 kbytes and an available bandwidth of 2 Mbits/s. The transfer is performed by using a standard TCP with a 64 kbytes buffer length both at the receiver and at the source. Two RTT values have been set: 100 ms and 500 ms. The difference is outstanding. While the first case requires only three seconds, the other case needs more than 12 seconds. Table 4.10 contains the exact values of the overall transmission time and of the throughput measured in the final phase of the connection (a movable window is used to measure the throughput).

The implications on the different applications mentioned in the previous section are simple to imagine. It is sufficient to think of a tele-learning system where the student is waiting for the material (an image or a graph) on the screen, or a remote control system aimed at activating an alarm or a robot.

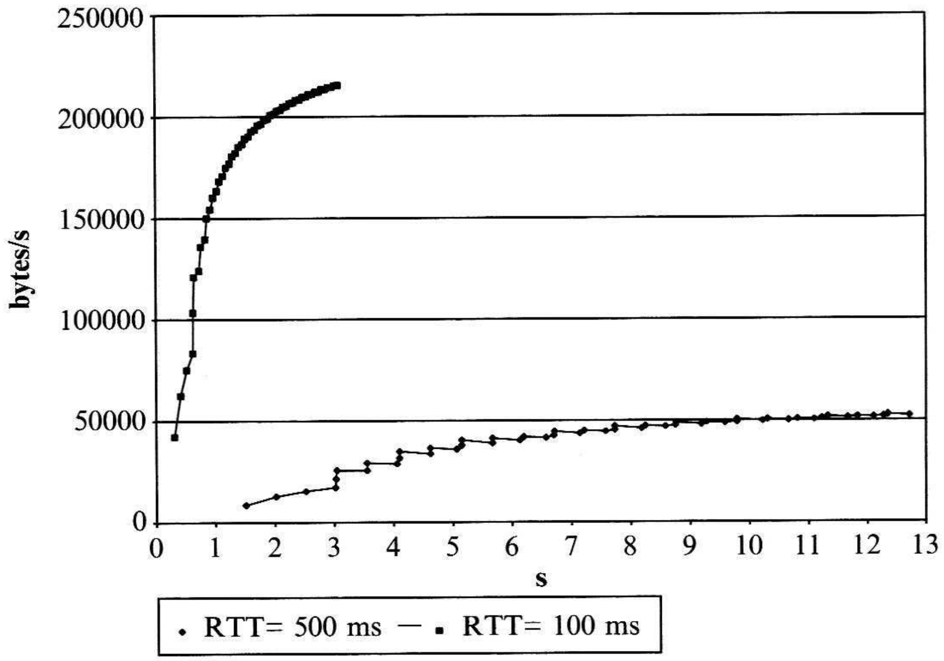

Figure 4.23 Throughput versus time, 675 kbytes file transfer

Table 4.10 Overall transmission time and throughput in the final phase of the connection

RTT (ms)	Time (s)	Throughput (bytes/s)
100	3.1	215154
500	12.7	52278

Moreover, within a satellite environment, the problems are different if a LEO (Low Earth Orbit), MEO (Medium Earth Orbit) or GEO (Geostationary Orbit) satellite system is used. Issues related to each environment are listed in RFC2488[49]. This subsection mainly concerns GEO satellites.

4.4.3.3 State-of-the-art

The following subsection is written for students and scientists who wish to know more about the topic and to start a research activity in it. Actually, it is a list of references with a few explanations. It should be considered as a guide for further studies. Its reading may be also postponed until the end of the section.

The problem of improving TCP over satellite has been investigated in the literature for some years: see Partridge and shepard [50] for a first overview on the topic, and Lakshman [51] and medhow [51] for a more specific study in TCP/IP networks with high delay per bandwidth product and random loss. More recently, Ghani and Ditil [52] provided a summary about improved TCP versions, as well as issues and challenges in satellite TCP, and possible enhancements at the link layer; Henderson and katz [53] highlight the ways in which latency and asymmetry impair TCP performance; RFC2760 [54] lists the main limitations of the TCP over satellite, and proposes many possible methods to help. Barakat *et al.* [55] represents, to the best of the author's knowledge, the most recent tutorial paper on the topic: various possible improvements both at the transport level and at the application and network levels are summarized and referenced; the paper also focuses on large delay per bandwidth product networks, and suggests possible modifications to the TCP, such as the buffer size. A recent issue of the *International Journal of Satellite Communications* is entirely dedicated to IP over satellite [73]. In more detail, Bharadwaj *et al.* [56] propose a TCP splitting architecture for hybrid environments (see also Zhang *et al.* [57]); Kruse *et al.* [58] analyse the performance of web retrievals over satellite, and Marchese [59] gives an extensive analysis of TCP behavior by varying parameters such as the buffer size and the initial congestion window. Goyal *et al.* [60] also focus on buffer management, but in an ATM environment. Also, international standardization groups such as the Consultative Committee for Space Data Systems (CCSDS), which has already issued a recommendation [61], and the European Telecommunications Standards Institute (ETSI), which is running its activity within the framework of the SES BSM (Satellite Earth Station – Broadband Satellite Multimedia) working group, are active on these issues.

The concept that the satellite portion of a network might be isolated and receive a different treatment and attention with respect to the cabled parts of the network is also investigated, as already mentioned above concerning Bharadwaj *et al.* [56]; methodologies such as TCP splitting [50,52,56,57] and TCP spoofing [50,57] bypass the concept of end-to-end service by either dividing the TCP connection into segments or introducing intermediate gateways, with the aim of isolating the satellite link. The drawback is losing the end-to-end characteristics of the transport layer. The recent RFC 3135 [62] is dedicated to extend this concept by introducing Performance Enhancing Proxies (PEPs) intended to mitigate link-related degradations. RFC 3135 is a survey of PEP techniques, not specifically dedicated to the TCP, even if emphasis is put on it. Motivations for their development are described, as well as consequences and drawbacks. Transport layer PEP implementations may split the transport connection and, as a consequence, the end-to-end characteristic is lost. The concept will be investigated in the following sections more deeply, when the solutions proposed by the author are listed.

Many national and international programs and projects (listed extensively in Marchese [59]) in Europe, Japan and the USA concern satellite networks and applications. In particular, some of them are aimed at improving performance at the transport level. NASA ACTS (in Brooks *et al.* [63] and Ivancic *et al.* [64]), ESA ARTES-3 [65] and CNIT-ASI [66] deserve particular attention, among many others.

4.4.4 Solution Frameworks

4.4.4.1 Framework: Black Box and Complete Knowledge Approaches

In the following, two possible frameworks where the different solutions to improve the performance of the transport layer over satellite channels may be classified are proposed: the Black Box (BB) approach and the Complete Knowledge (CK) approach. The former implies that only the end terminals may be modified; the rest of the network is considered non-accessible (i.e. a black box). The latter allows tuning parameters and algorithms in the network components. Most of the state-of-the-art has been based on the Black Box approach.

The classification proposed is not the only possible one and, probably, it is not exhaustive (i.e. not all the algorithms and methods in the literature can be classified within one of the two classes), but it is useful to understand the problems and to introduce the analysis proposed in the section.

A simple example of a GEO satellite telecommunication network, which is also the test-bed network used to obtain the results reported in the section, is reported in Figure 4.24. The box identified as APPLICATION PC may also represent a Local Area Network (LAN). The real system employs the ITALSAT II satellite, and it provides coverage in the single spot-beam on Ka band (20–30 GHz). The overall bandwidth is 36 MHz. Each satellite station can be assigned a full-duplex channel with a bit rate ranging from 32 kbits/s to 2 Mbits/s, the latter used in the experiments, and it is made up of the following components:

- Satellite modem, connected to the RF device.
- RF (Radio Frequency) device.
- IP Router connected to:
  - Satellite modem via RS449 Serial Interface.
  - Application PC via Ethernet IEEE 802.3 10BASE-T link.
- Application PC: PCs Pentium III. They are the source of the user service.

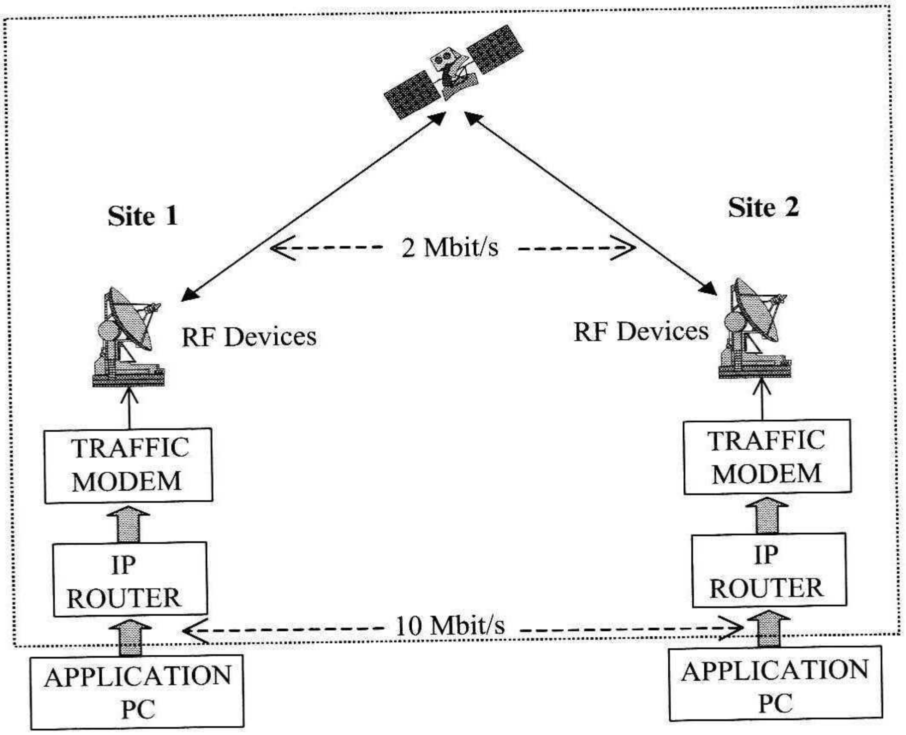

Figure 4.24 Test-bed network

The problem may be considered in two different ways from a network point of view. The first is considering the network as a Black Box, ignoring each particular configuration of the devices. This approach has been used in previous works by the same author [59,67] and it has been modeled in Figure 4.25. The transport layer (i.e. TCP) is modified and tuned by acting only on the end user terminals. The rest of the network is considered as a black box, where even if the intermediate devices and the configurations are known, they cannot be modified.

If Figure 4.25 is taken as a reference, the Black Box corresponds to the part of the network contained within the dashed line.

An alternative approach is supposing Complete Knowledge of each network device (e.g. routers, modems and channel characteristics), and the possibility of modifying the configurations to improve the performance of the overall satellite network (or of the satellite portion of the network). This approach is possible if the network is proprietary.

The intervention is very different in the two cases. Within the Black Box approach, only a modification of the TCP protocol is possible, while a structural revision of the entire protocol stack is allowed within the Complete Knowledge approach. The proposals concerning the latter are explained in Section 4.4.6. The solutions of the Black Box approach are contained in the next section.

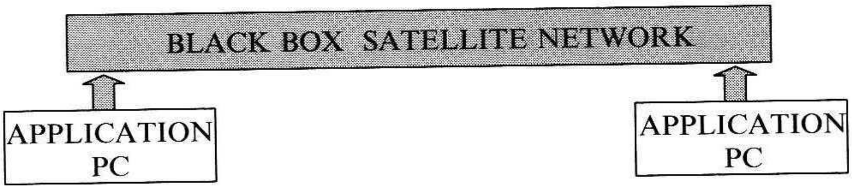

Figure 4.25 Black box approach

4.4.5 The Black Box Approach

4.4.5.1 Parameterized TCP

A possibility to improve the performance within the Black Box approach is to parameterize the TCP protocol. The problem, as said in Section 4.4.3, concerns the TCP congestion control summarized in Table 4.8.

Table 4.11 contains a proposal for the parameterization of TCP. The parameters IW and Th, along with the two functions $F(\cdot)$ and $G(\cdot)$, may be tuned following both the characteristics of the physical channel (delay, loss, bit error rate,) and the network status (e.g. congestion). The function $F(\cdot)$ is aimed at regulating the size of the congestion window in the SLOW START phase. The characteristics of $F(\cdot)$ affect the increase of the window and, as a consequence, the transmission speed and protocol performance. The definition of $F(\cdot)$ is not trivial, and many considerations may affect the decision. The increment in cwnd strictly depends upon the current value of the cwnd itself, and on the number of received acknowledgements, as indicated in Table 4.11. The choice allows us to tune the behavior of the protocol in dependence of the congestion window, and to measure, to some extent, the network status represented by the arriving acknowledgements. The function $G(\cdot)$ is aimed at regulating the behavior of the congestion avoidance algorithm. The modification of the congestion avoidance scheme has not provided outstanding results over GEO channels, but it might be very useful in LEO or radio-mobile environments.

An approach in the field has been carried out. The various parameters have been tuned step-by-step using the simple but real network presented in Section 4.4.4. Even if the results obtained depend upon the particular implementation and on the particular network and device, the general methodology is not affected. The aim was, on the one hand, to investigate precisely the role of the various parameters, and on the other, to find on optimal solution for a small private network so as to improve the performance of the services offered within the framework of a project described by Adami *et al.* [66]. The experimentation corresponds exactly to the steps followed to fulfil the project.

Table 4.11 Parameterized TCP

TW=min{cwnd, min(source buff, rwnd)}	

SLOW START [cwnd<ssthr]	cwnd=IW · SMSS
	ssthr=Th
	ACK → cwnd = cwnd + F (# of received acks, cwnd) · SMSS

CONGESTION AVOIDANCE [cwnd≥ssthr]	< cwnd > ACK → cwnd = cwnd + **G** (cwnd, •)

FAST RETRANSMIT / RECOVERY	ssthr = max{FlightSize/2, 2 · SMSS}
	cwnd = ssthr + 3 · SMSS
	Duplicated ACK → cwnd = cwnd + 1 · SMSS
	cwnd=ssthr

Among the parameters indicated above, the buffer length both at the source (source buff) and at the destination (which affects rwnd), the initial window (IW) and the function $F(\cdot)$ have been studied. The value of the parameter Th has been set to a very high value (infinite); the function $G(\text{cwnd}, \cdot)$ has been set to 1.

The study concerning the buffer length and the initial window partially derives from Marchese [59], and the investigation of function $F(\cdot)$ partially from Marchese [67,68]. It is briefly summarized in the following.

4.4.5.2 The Real Test-bed

The real test-bed has been reported in Figure 4.24): two remote hosts are connected through a satellite link using IP routers. An average Round Trip Delay (RTT) of 511 ms has been measured, and the TCP/IP protocol stack is used. The data link level of the router uses HDLC encapsulation on the satellite side, where a serial interface is utilized, and Ethernet on the LAN side. A raw Bit Error Rate (BER) (i.e. BER with no channel coding) of approximately 10^{-2} has been measured; the use of a sequential channel coding with code rate 1/2, to correct transmission errors, has allowed us to reach a BER of about 10^{-8}. As a consequence, the data link protocol 'sees' a reliable channel. The system offers the possibility of selecting the transmission bit rate over the satellite link, and a bit rate of 2048 kbits/s has been used for the tests.

Test application

The application used to get the results is a simple ftp-like one, i.e. a file transfer application located just above the TCP. It allows data transfer of variable dimension (H (bytes), in the following) between the two remote sites. The application designed allows the transfer of a single file at a time, which is a case often reported in the literature, both as a benchmark for working and as a configuration used in real environments. Two types of files have been utilized to perform the tests, and to study the behavior of the modified TCP: a file of relevant dimension of about 2.8 Mbytes (2,800,100 bytes), indicated with $H = 2.8$ Mbytes, and a small file of about 100 kbytes (105,100 bytes), indicated with $H = 100$ kbytes. The multiple connections case, reported to show the effect of a loaded network on the modified TCP, is obtained by activating a fixed number (N, in the following) of connections at the same time. File transfers of 100 kbytes each are assumed for the multiple case.

4.4.5.3 Buffer Length and Initial Window (IW)

The analysis is dedicated to investigating the behavior of TCP by varying the value of the initial window (IW is measured in bytes, i.e. the notation IW=1 means IW $= 1 \cdot$ SMSS (bytes)) and of the buffer dimension. The latter is intended as the memory availability in bytes, for source and destination, which is kept equal, i.e. the buffer has the same length both for the source and the destination. It is identified with the variable 'buf' in the following.

Concerning the initial congestion window, the issue has also been treated in the literature. Simulation studies, though not for the specific satellite environment [69], show the positive effect of an increased IW for a single connection. RFC2414 [70] clarifies the strict dependence of the performance on the application environment, and suggests that 'larger initial windows should not dramatically increase the burstiness of TCP traffic in the

Internet today'. The IW is set to 1 in the TCP version taken as the reference, as shown in Table 4.8.

The quantity 'gain' is computed as follows: if T_{REF} is the reference transmission time and T is a generic transmission time, the percentage gain is defined as

$$\%\mathrm{Gain} = \begin{cases} \dfrac{T_{\mathrm{REF}} - T}{T_{\mathrm{REF}}} \cdot 100, & \text{if } T < T_{\mathrm{REF}} \\ 0, & \text{otherwise} \end{cases} \tag{4.51}$$

The IW is mainly responsible for the behavior in the first part of the transmission. As a consequence, short file transfers receive a real advantage from an increased IW. Tuning of the initial congestion window allows mitigation of the problem introduced by the large 'delay per bandwidth product'.

The congestion window represents the network bottleneck until it reaches the buffer length, which happens after a few seconds if the bandwidth available is large, then the buffer rules the system. The buffer length is very important for the system performance. A short buffer drastically limits performance, but an excessively long buffer makes the system congested. When the system is congested, the throughput is strongly reduced, even if the efficiency is high at the beginning of the connection. The congestion issue deserves particular attention because, even if the qualitative behavior does not change, the specific measures depend heavily upon the network devices' configuration. In more detail, some TCP segments might be lost "due to the inability of the router to handle small bursts" [70]. No parameter tuning has been realized in the routers used to perform the tests; default configurations have been maintained within the Black Box scenario. Nevertheless, precise optimization and knowledge of every configuration detail would be very useful. The idea of using a Complete Knowledge approach, the object of Section 4.4.6, comes from just from this observation.

Table 4.12 summarizes the effects of the parameter tuning reported in detail in Marchese [59]. The table contains the combination of the two parameters analyzed (IW and buf), the time required for the overall transmission and the gain in percentage obtained with respect to the basic configuration (IW = 1, buf = 64 kbytes). The measures obtained with $H = 2.8$ Mbytes have been chosen. The gain in the overall transmission time (up to 71.63%) is mainly due to the buffer length, which may represent a real bottleneck for the system. It has to be remembered that the effective transmission window is the minimum between cwnd and the minimum between the source buffer length and the receiver's advertised window (rwnd), which is strictly dependent on the receiver buffer length. A large buffer guarantees that the bottleneck of the system (concerning the packets in flight)

Table 4.12 Comparison of TCP configurations by varying the initial congestion window and the buffer length, *H=2.8 Mbytes*

IW, buf (kbytes)	Transmission time (s)	% gain
1, 64	49.21	–
6, 64	47.55	3.37
1, 320	15.87	67.75
6, 320	13.96	71.63

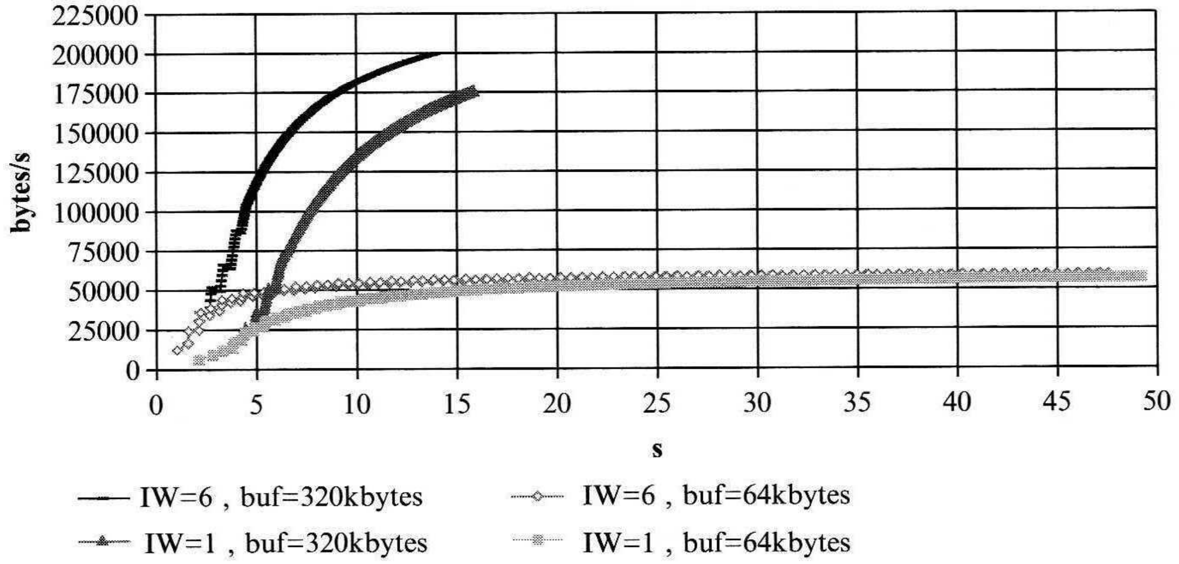

Figure 4.26 Throughput (bytes/s) versus time for different values of the buffer length and of the initial congestion window, $H = 2.8$ Mbytes

is not so severe: if the buffer is short, the congestion window quickly reaches the limit imposed by the buffer length, as indicated in Table 4.11. The values of the source and receiver buffer length being the same, the limit, in this case, is represented by the value of rwnd. On the other hand, IW governs the throughput in the initial phase. It is sufficient to observe Figure 4.26, where throughput versus time is shown for the same configurations of Table 4.12. If the configurations with the same buffer length are analyzed, the difference between the increase in speed for different values of IW is outstanding. The behavior after 5 s may be taken as an example: (IW=1, buf=64 kbytes) has a throughput of about 26 kbytes/s (IW=6, buf=64 kbytes) of 48 kbytes/s. The throughput for (IW=1, buf=320 kbytes) is about 35 kbytes, whereas it is 118 kbytes/s for (IW=6, buf=320 kbytes).

Figure 4.27, which shows the average throughput per connection, compares the behavior of different TCP configurations versus the number N of active connections for a 100 kbyte transfer. The test is aimed at verifying the effect of the modifications proposed in the presence of a network loaded with multiple connections. Four configurations are taken into account: the reference configuration (IW=1, buf=64 kbytes); (IW=1, buf=320 kbytes), where only the buffer is varied; and two configurations where both buf and IW are increased: (IW=2, buf=320 kbytes) and (IW=6, buf=320 kbytes).

The gain in throughput is evident up to 15 active connections; for larger values of N, the traffic load due to the number of connections in progress makes the TCP insensitive to the modifications introduced.

The effect of these factors measured in a real Web tele-learning session may be found in Adami *et al.* [66]).

4.4.5.4 The Function $F(\cdot)$

The definition of $F(\cdot)$ is not trivial, and many considerations may affect the decision: the choice performed in this chapter is aimed at increasing the transmission speed in the initial phase without entering a congestion period. The behavior has been tested with different types of functions. The increment of cwnd strictly depends upon the current value of the cwnd itself, and on the number of received acknowledgements. The choice allows us to tune the behavior of the protocol depending on the congestion window, and to measure, to

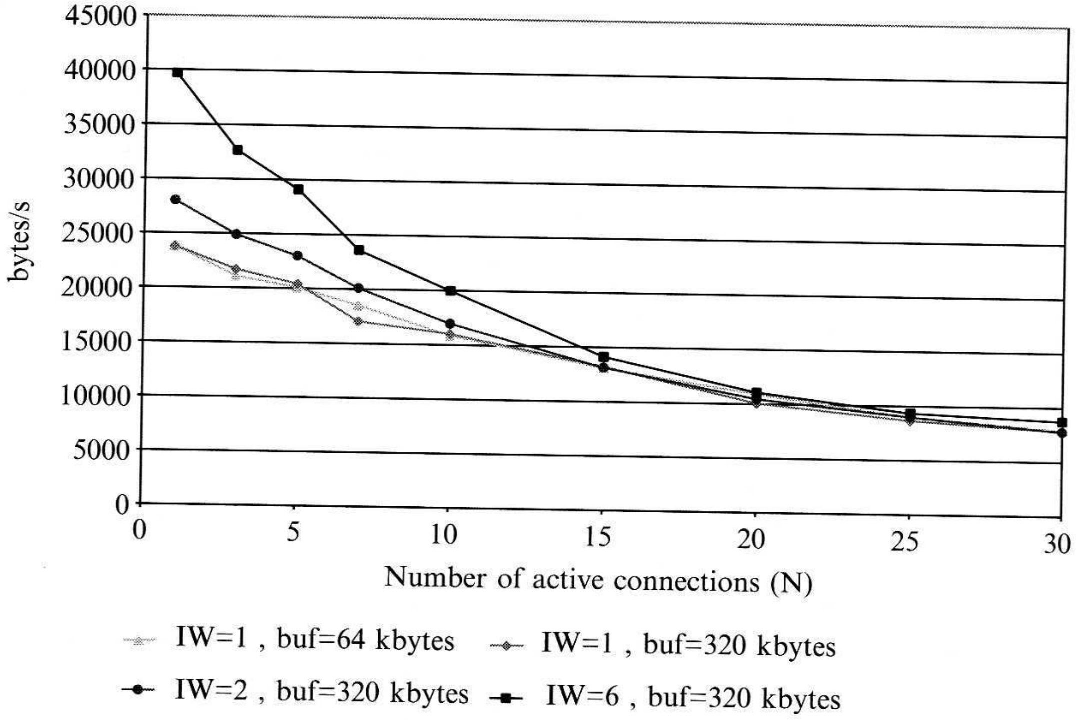

Figure 4.27 Throughput (bytes/s) versus the number of active connections (N), $H = 100$ kbytes

some extent, the network status represented by the arriving ACKs. Let the variable N_ack be the number of received acknowledgements in a single TCP connection.

The reference TCP sets the function

$$F(\text{N\_ack}) = 1 \tag{4.52}$$

The alternatives chosen set the function $F(N\_ack)$ as follows:

$$F'(\text{N\_ack}) = K \tag{4.53}$$

At each acknowledgement received, the increment is constant.

$$F''(N\_ack) = \begin{cases} N\_ack & \text{if cwnd} \leq \text{thr} \\ 1 & \text{otherwise} \end{cases} \tag{4.54}$$

The increment is linear up to the value 'thr' of a fixed threshold; it is constant after this value. This method is referenced as 'Linear thr' in the results presented to simplify the notation (i.e. if thr=20, the method is identified as 'Linear 20'). It is important to note that, when the increment is linear, the protocol behavior is very aggressive: if no loss is experienced, the number of received acknowledgements as shown in Equation (4.55) rules the size of cwnd.

$$\text{cwnd}(N\_ack) = \text{cwnd}(N\_ack - 1) + N\_ack \cdot \text{SMSS} \tag{4.55}$$

$$F'''(N\_ack) = \begin{cases} K_{\text{thr}_1} \cdot N\_ack & \text{if cwnd} < \text{thr}_1 \\ K_{\text{thr}_2} \cdot N\_ack & \text{if cwnd} < \text{thr}_2 \\ K_{\text{thr}_3} \cdot N\_ack & \text{if cwnd} < \text{thr}_3 \\ . & . \\ . & . \\ K_{\text{thr}_n} \cdot N\_ack & \text{if cwnd} < \text{thr}_\text{n} \\ 1 & \text{otherwise} \end{cases} \tag{4.56}$$

In this case a variable number of thresholds (i.e. thr_n, where $n \in N$) may be used. Function (4.56) is aimed at adapting the protocol behavior through the constants (K_{thr_n}, if n thresholds are used). Three thresholds have been heuristically estimated to be a proper number to increase the rate in the first phase of the transmission, and to smooth it on time. The function chosen appears as in Equation (4.57):

$$F'''(N\_ack) = \begin{cases} K_{\text{thr}_1} \cdot N\_ack & \text{if cwnd} < \text{thr}_1 \\ K_{\text{thr}_2} \cdot N\_ack & \text{if cwnd} < \text{thr}_2 \\ K_{\text{thr}_3} \cdot N\_ack & \text{if cwnd} < \text{thr}_3 \\ 1 & \text{otherwise} \end{cases} \tag{4.57}$$

The notation used is Linear ($\text{thr}_1 - \text{thr}_2 - \text{thr}_3$); the value of the constant K_{thr_n}, with $n \in \{1, 2, 3\}$, represents the angular coefficient of the increase line; its value governs the speed of the increase, and rules the TCP behavior.

Table 4.13 summarizes the results concerning the overall transfer time for the configurations resulting from the analysis in the single connection case. The gain is really good for any configuration, and it rises up to 74.5% if function $F'''(\cdot)$ is utilized with $\text{thr}_1 = 20 - K_{\text{thr}_1} = 4$, $\text{thr}_2 = 30 - K_{\text{thr}_2} = 2$, $\text{thr}_3 = 40 - K_{\text{thr}_3} = 1$.

The configurations providing the best results when more connections are routed have been selected and shown in Figure 4.28, along with the reference configuration. Use of the configuration Linear 20–30–40 is very efficient for a limited number of connections and, at the same time, it allows congestion avoidance when the load increases. All the modified configurations are largely equivalent if the number of active connections ranges from 7 to 15. After this latter value there is no gain in using a modified version of TCP.

It is important to note that all the modifications proposed have been performed 'blindly' (Black Box approach), without any information about the internal components of the network. If this information is used (completely or partially), the performance may be improved further, as well as the comprehension of the overall system.

4.4.6 Complete Knowledge Approach

4.4.6.1 Introduction

The Complete Knowledge approach supposes complete control of any network devices, and allows a joint configuration of all the functional layers involved to get an optimized network resource management aimed at improving the overall performance offered by the network.

Table 4.13 Overall transfer time and throughput, different increment functions, $H = 2.8$ Mbytes

TCP configuration	Transfer Time (s)	Throughput (kbytes/s)	Gain (%)
Reference: IW = 1-buf = 64	49.21	56.7	–
IW = 6-buf = 320	13.96	199.8	71.6
IW = 6-buf = 320-K = 2	13.22	210.9	73.1
IW = 6-buf = 320-K = 4	12.65	220.3	74.3
IW = 6-buf = 320-Linear10	13.81	201.9	71.9
IW = 6-buf = 320-Linear20	13.13	211.9	73.3
IW = 6-buf = 320-Linear50	12.80	217.8	74.0
IW = 6-buf = 320-Linear20-30-40	12.57	221.9	74.5

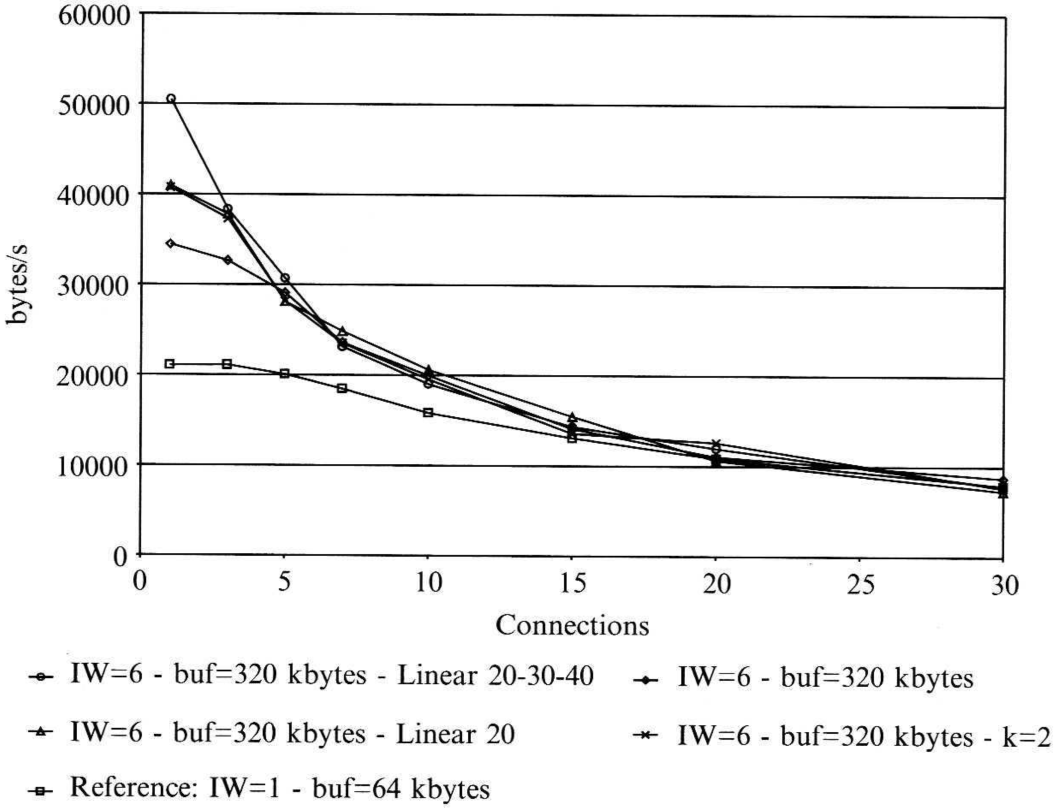

Figure 4.28 Throughput vs. number of connections, different slow start algorithm, multi-connection case ($H = 100\,\text{kbytes}$)

Within this framework, it is feasible to propose a protocol architecture, designed for a heterogeneous network involving satellite portions, which conveys the benefits both from the Black Box and the Complete Knowledge approaches.

The advantages that derive from the Black Box approach are utilized to design a novel network architecture suited for satellite transportation, where the transport layer is divided into two parts, one completely dedicated to the satellite portion of the network (Satellite Transport Layer – STL). The two components of the transport layer are joined by Relay Entities, which imply a complete redefinition of the protocol stack on the satellite side (Satellite Protocol Stack – SPS).

The new Satellite Transport Layer uses a specific Satellite Transport Protocol (STP), which can also be obtained by parameterization of the TCP presented in the Black Box approach, to meet all the network requirements and characteristics in terms of delay, reliability and speed.

The Satellite Protocol Stack will also operate at the network layer, and will benefit from possible resource allocation features of the layer 2 (data link) and, in particular, of the MAC (Medium Access Control) sub-layer.

In practice, the CK approach allows us to use all the possibilities to improve the performance without any limitation. It is clear that, from a Black Box situation to a real Complete Knowledge one, there are many intermediate solutions that may improve the network performance. The following study is partially taken from Marchese [71].

4.4.6.2 Operative Environment and Scope

The general architecture is reported in Figure 4.27. The network is composed of terrestrial portions, represented by the Internet in the figure, and of a satellite portion (a backbone, in

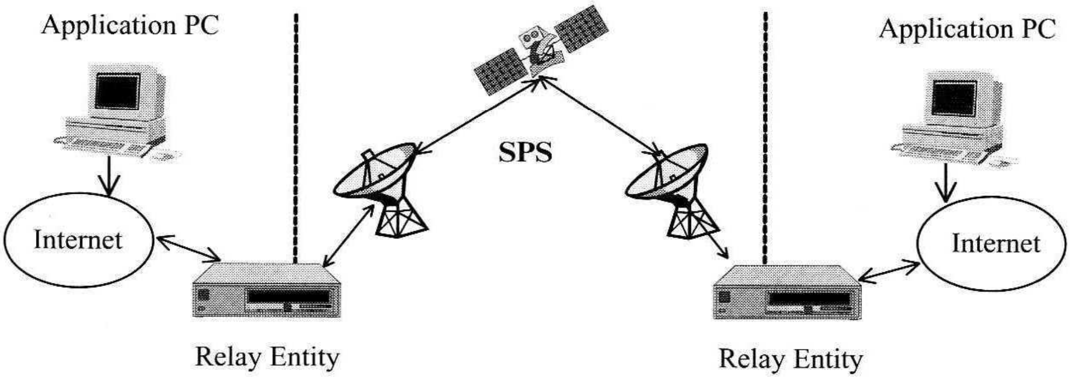

Figure 4.29 SPS architecture

this case). The latter is isolated from the rest of the network using Relay Entities. Only two of them are shown in the figure, but actually one Relay Entity is required whenever a satellite link is accessed.

The transport layer of this new Satellite Protocol Stack (SPS) is called the Satellite Transport Layer (STL), and it implements a Satellite Transport Protocol (STP) suited for the specific environment.

The architecture proposed may be a valid alternative both when the satellite portion represents a backbone network and when it represents an access network. Figure 4.29 contains the proposal already presented in the backbone case. Figure 4.30 shows a possible solution in the access case: the Relay Entity is a simple tool, directly attached to the Application PC. It may also be a hardware card inside the Application PC. In this case, it may be a simple plug-in module, such as a network or a video card to be inserted inside the PC.

4.4.6.3 The Satellite Protocol Stack (SPS) Architecture

From the protocol layering point of view, the key point is represented by the two Relay Entities, which are two gateways towards the satellite portion of the network. The SPS acts on the satellite links by using the necessary information, because it has knowledge and control of all the parameters. The Relay Layer guarantees the communication between the satellite transport layer and the protocol used in the cable part (i.e. TCP).

Two possible alternatives may be chosen concerning the transport protocol:

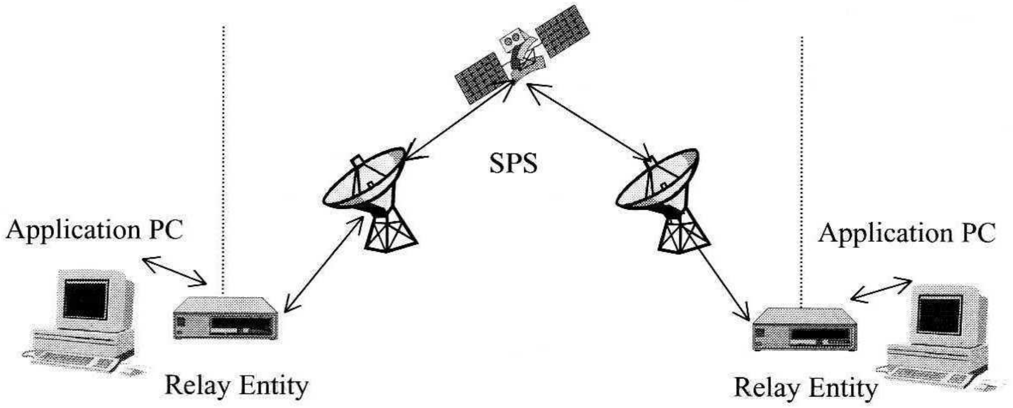

Figure 4.30 Access network

- completely bypassing the concept of end-to-end service at the transport layer;
- preserving the end-to-end characteristic of the transport layer.

The first choice is represented in Figure 4.31. The connection at the transport layer is divided into two parts, dedicated, respectively, to the cable and the satellite part. The source receives the acknowledgement from the first Relay Entity, which opens another connection, with different parameters based on the current status of the satellite portion, and allocates the resources available. The Relay Entity on the other side of the satellite link operates similarly towards the destination. The transport layer of the cable portions is untouched. The end-to-end connection may only be guaranteed statistically. This case concerns some of the PEP (Performance Enhancing Proxy) architectures introduced in RFC 3135 [62].

The second architecture is aimed at preserving the end-to-end characteristic of the transport layer. In this case also, the transport protocol in the terrestrial portion should be modified. The transport layer is divided into two sub-layers: the upper one, which guarantees the end-to-end characteristic, and the lower sub-layer, which is divided into two parts and interfaces the STL. The terrestrial side of the lower transport layer may be also represented by the TCP. Figure 4.32 shows the layered protocol architecture. The transport layer is modified even if the interface with the adjacent layers may be the same as in the TCP. The Upper Transport layer will include the TCP and the UDP implementation to allow full compatibility with a different architecture; for example, to guarantee a correct working even if a TCP/IP stack is present at the destination. The TCP/IP stack includes the use of UDP. The transport layer to use is properly identified during the set-up.

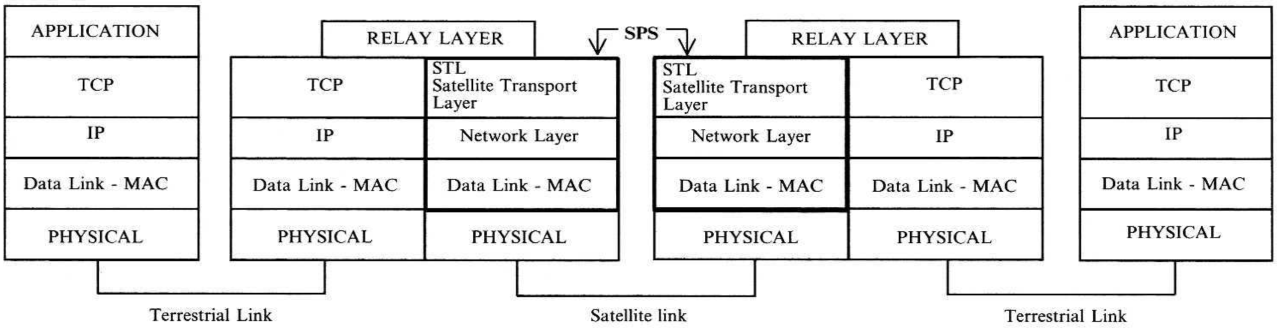

Figure 4.31 SPS architecture

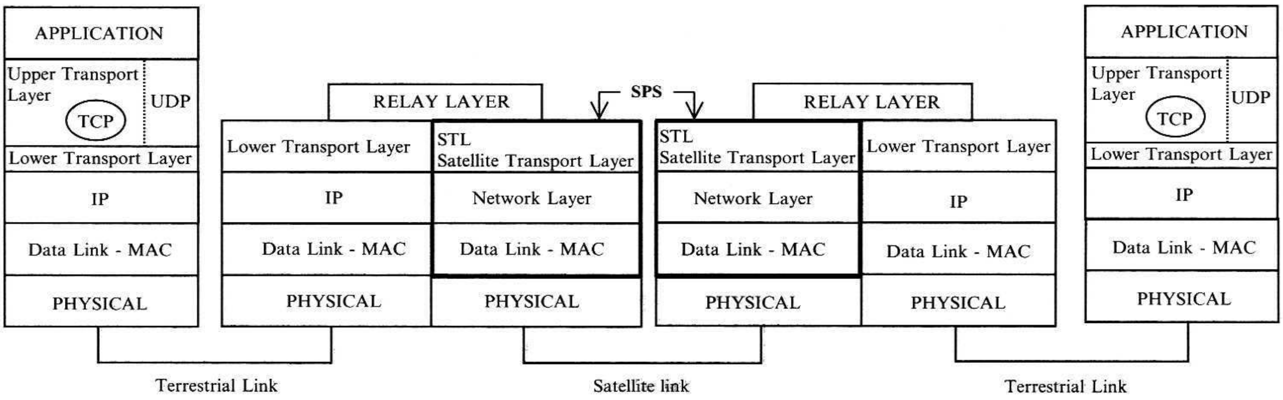

Figure 4.32 End-to-end SPS architecture

Both the choices are interesting. The preservation of the end-to-end characteristic is one of the objects of the project 'Transport Protocols and Resource Management for Mobile Satellite Networks' funded by the European Space Agency (ESA) and carried out by CNIT (Italian National Consortium for Telecommunications), Marconi Mobile (as Project leader) and Etnoteam.

The performance of the Relay Entity strictly depends upon the design of each layer. One idea is reported in Figure 4.33 where the architecture of a Relay Entity is shown. The protocol stack is completely re-designed on the satellite side. The essential information concerning each layer (Transport, Network and Data Link) of the terrestrial side is compressed in the Relay Layer PDU (Protocol Data Unit), i.e. a specific unit of information created in the Relay Layer. The Data Link layer (the Medium Access Control sublayer, in this case) offers to the upper layer a Bandwidth Reservation service, a sort of Bandwidth Pipe available to the Network Layer, which can itself reserve resources for the Transport Layer. The Network Layer may use the structure of the IP layer, but it may be properly designed together with the STL layer, so as to avoid the possibility of the event 'congestion' (and, for instance, the consequent 'congestion avoidance' phase, if a standard TCP was used), and to optimize the performance of the overall transmission on the satellite side. The Network Layer may reserve resources by using the Integrated Services [72] or the Differentiated Services [72] approach, considering the two possibilities offered in the IP world. In any case the aim is to create a bandwidth pipe (Relay Entity-to-Relay Entity, in the satellite portion), so as to offer a dedicated channel to a single connection at the transport layer or to a group of connections at the transport layer. If it is not possible, the pipe shown in Figure 4.33 may be simply represented by the transfer capacity of the physical interface. In this latter case, all the connections of the STL share the same portion of bandwidth and the STL design must take it into account.

4.4.6.4 The Satellite Transport Protocol (STP)

The transport layer will be properly designed to consider all the possible peculiarities of the application environment. Some guidelines (concerning the STL and its implementation through the Satellite Transport Protocol (STP)) may also be introduced. The modified version of the TCP proposed in the previous section can be considered a former implementation of the transport layer and a basis for the design of STP. For example, functions $F(\cdot)$ and $G(\cdot)$ may be of help, but the protocol STP can also be completely re-written.

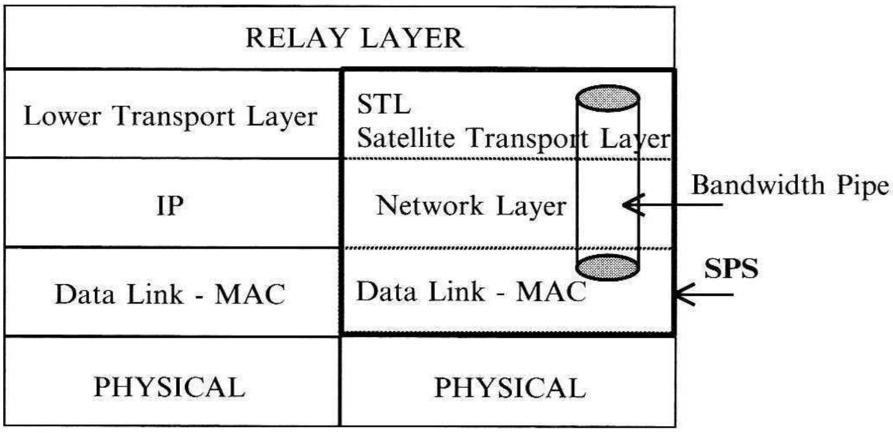

Figure 4.33 Design of the Relay Entity

Slow Start Algorithm: the mechanism no longer has need of testing the network congestion at the Relay Entity, because the status is known. The algorithm has to rule the flow in accordance with the contemporary presence of other flows, whose characteristics are known. The function $F(\cdot)$, along with the other parameters involved (e.g. the initial window IW), might help to get achieve the goal. A proper tuning of the IP buffer is fundamental.

Congestion Avoidance Algorithm: the schemes currently used take into account only congestion conditions; a loss is attributed to a congestion event. Now, due to knowledge of the IP buffer status, a loss should be attributed mainly to transmission errors. The function $G(\cdot)$ might have the responsibility for this part.

4.4.6.5 Some Preliminary Result

Three transport layer configurations are compared. The test-bed used is the same as in the previous section:

- A TCP configuration, adapted to the satellite GEO environment in the Black Box approach, identified as Modified TCP (Reference), which applies an IW=2 and a TCP buffer of 320 Kbytes both at the source and at the destination. It was as efficient in terms of the congestion risk in the multi-connection case within the Black Box approach (see Figure 4.27).
- The TCP configuration that is the most efficient among the solutions experimented within the Black Box approach (Table 4.14 and Figure 4.26). It applies a 'IW=6 – buf=320 – Linear 20–30–40' solution that means an Initial Window of 6·SMSS, a source/receiver buffer length of 320 kbytes and a function $F'''(\cdot)$, utilized with $\text{thr}_1 = 20 - K_{\text{thr}_1} = 4$, $\text{thr}_2 = 30 - K_{\text{thr}_2} = 2$, $\text{thr}_3 = 40 - K_{\text{thr}_3} = 1$. It is identified as Modified TCP (Best).
- The new Complete Knowledge configuration, identified as STP, which adapts the parameters to the different situations by choosing the best configurations, including the IP layer buffer tuning, time by time. It is important to note that this configuration is only a first step towards a real Complete Knowledge scheme. In practice, for now, it is little more than a smart choice of the best configurations of the Black Box approach along with a proper IP buffer tuning.

The comparison is aimed at giving a first idea of the further improvement of the STP with respect to the modified TCP configurations, already adapted to satellite channels in the Black Box approach.

Figure 4.34 contains the throughput versus time for the three configurations mentioned and a file transfer of 2.8 Mbytes. The overall transmission time is 15.2 s for the Reference configuration, 12.57 s for the Modified TCP (Best) configuration and 11.69 s for STP. The gain of STL, computed as a percentage $100 \cdot (15.2 - 11.69)/15.2$), is 23.1% with respect to the Reference and 7% $(100{\cdot}(12.57 - 11.69)/12.57)$ with respect to the Best. STP has a shorter transmission time than the other configurations; thus the performance gain is actually the metric of 'reduction' of the overall transmission time. Figure 4.35 shows the behavior in the multi-connection case. The throughput in bytes/s is reported versus the number of connections in the network, each performing a file transfer of 2.8 Mbytes, only for the Reference configuration and for STP. An improvement is noticeable up to five connections. After that the bandwidth available (2 Mbits/s) is in-sufficient to match the

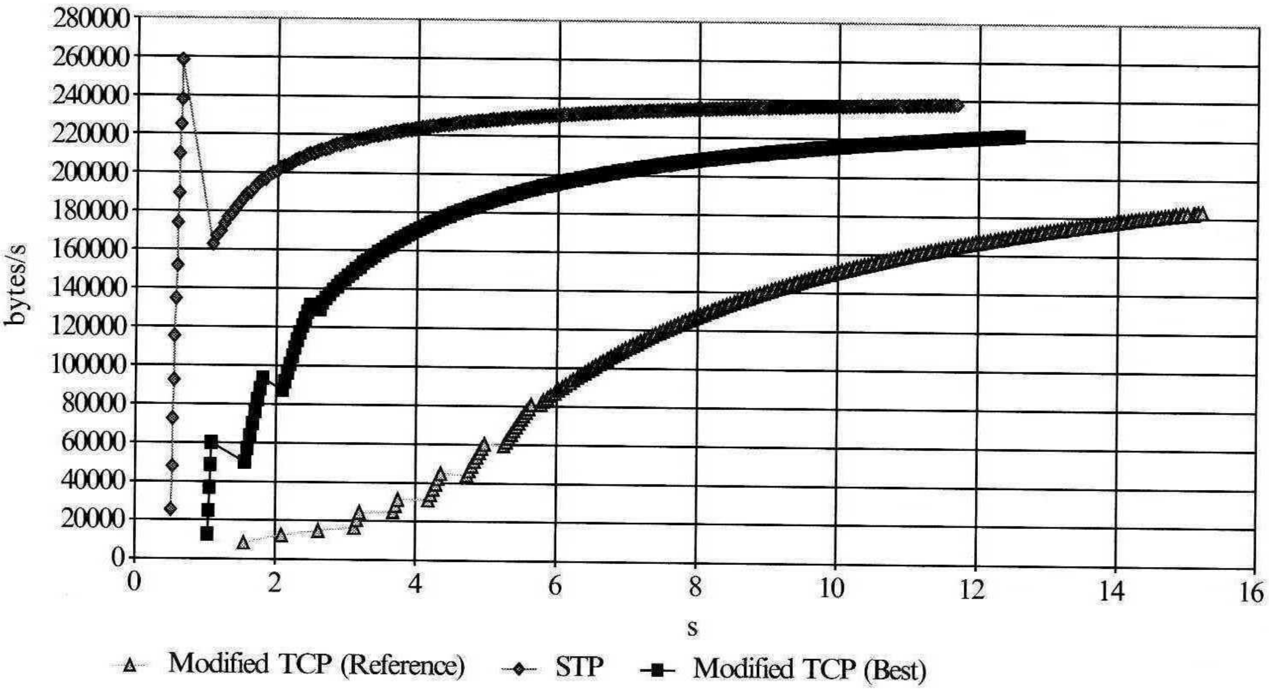

Figure 4.34 Throughput vs time, $H = 2.8$ Mbytes, mono-connection

requirements. The advantage is much more evident if a shorter transfer of 100 Kbytes is performed. Figure 4.36 shows the same quantities as in Figure 4.35 along with the Modified TCP (Best) configuration, for a 100 Kbytes file transfer. The configurations deriving from the Black Box approach, although very convenient with respect to the TCP commonly used, as is clear from the results in the previous section, may be strongly improved. The overall transmission time in the mono-connection case (a file of 100 kbytes) is about 3.7 s, for the Modified TCP (Reference) case, 2.1 s for the Modified TCP (Best) case and 1.4 s for STP. It corresponds to gains of 62% and 33%, respectively. A recent STL version particularly

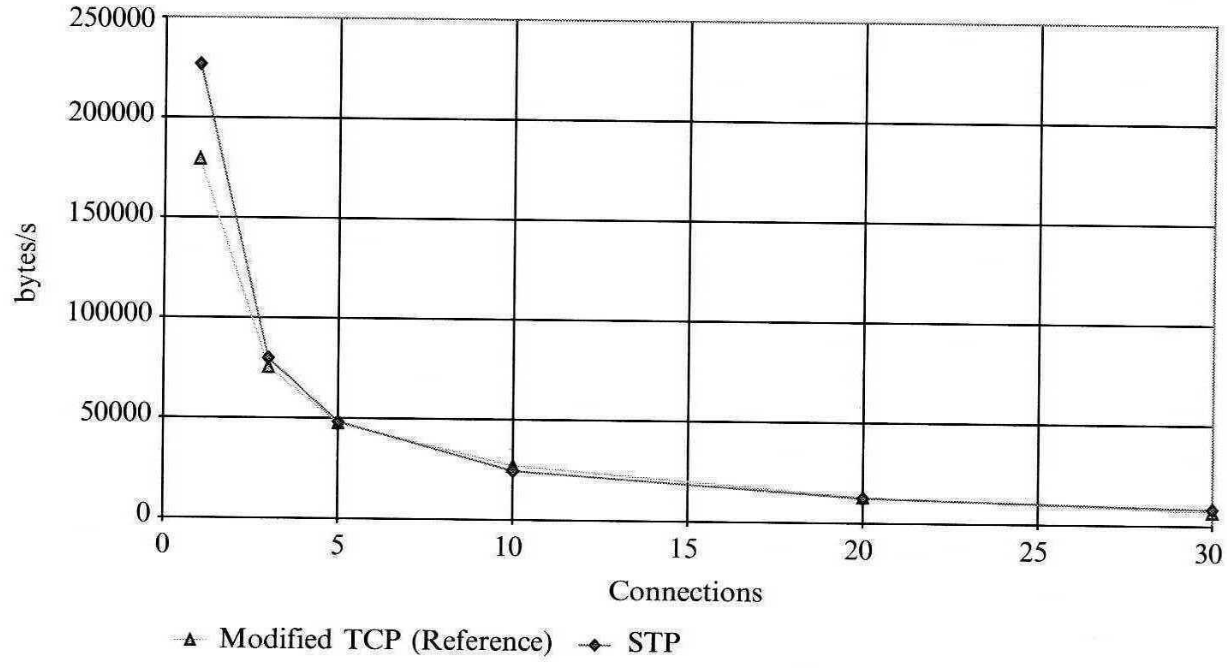

Figure 4.35 Throughput vs. number of connections, $H = 2.8$ Mbytes, multi-connection

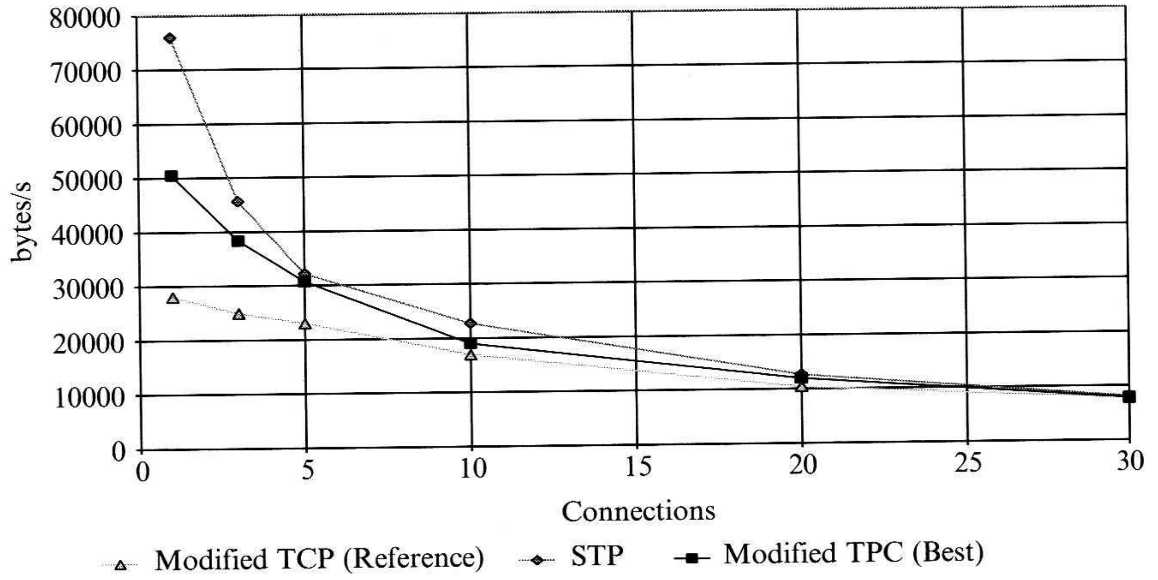

Figure 4.36 Throughput vs. number of connections, $H = 100$ Kbytes, multi-connection

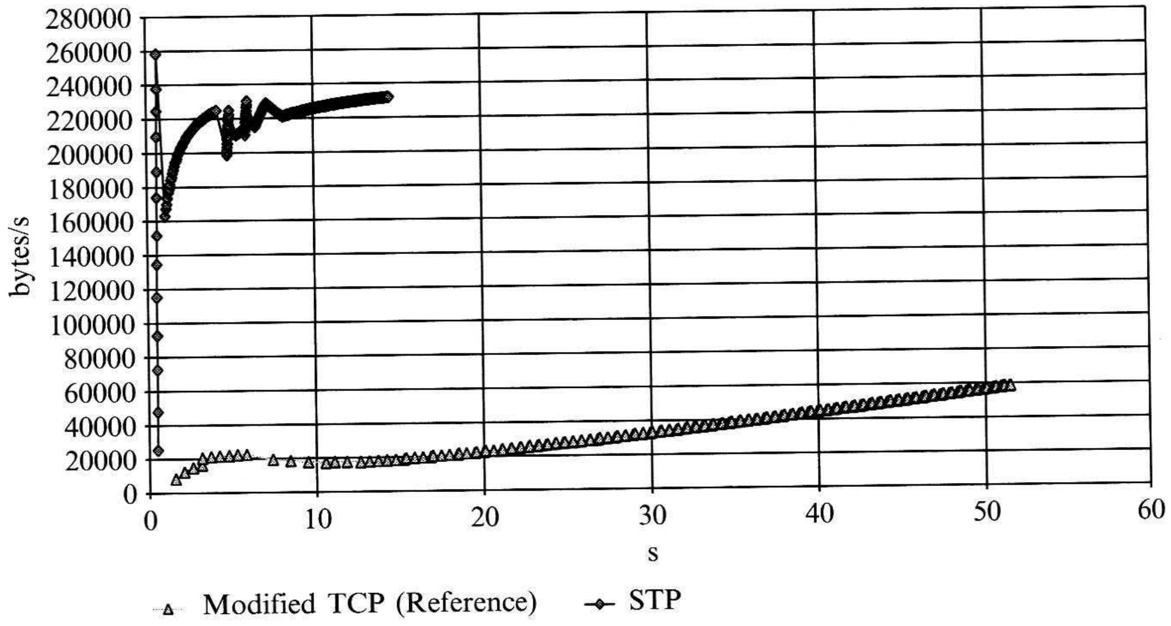

Figure 4.37 Throughput vs. time, $H = 2.8$ Mbytes, mono-connection, packet loss

optimized provided a 100 kbytes transfer in a time of about 1 s. The effect of such improvement in a remote control system (e.g. tele-robot, tele-control) is evident.

The last part of the results investigates the behavior of the new transport protocol when there are packet losses due to channel errors. Only STL and Modified TCP (Reference) have been used. The losses have been artificially introduced in the cases reported. The loss has been obtained by shutting down the modem for a fraction of second in the first phase of the connection. The IP router has been properly configured to avoid losses due to congestion. Figure 4.37 reports the throughput versus time for a 2.8 Mbytes transfer in the mono-connection case. The packet loss is much more intense for STP, due to the aggressive behavior in the first phase of the connection, where the shut down happens, but it recovers thanks to the correct interpretation of the losses, which are not due to congestion, as

Table 4.14 Overall transmission time and gain, 2.8 Mbytes file transfer, mono-connection, packet loss

Transport Protocol	Overall Transmission Time [s]	Gain (%)
Modified TCP (Reference)	51.5	–
STP	14.5	71.8

Table 4.15 Overall transmission time, 2.8 Mbytes file transfer, mono-connection, comparison of loss and no loss, Reference

Modified TCP (Reference)	Overall transmission time (s)
Loss	51.5
No loss	15.2

Table 4.16 Overall transmission time, 2.8 Mbytes file transfer, mono-connection, comparison of loss and no loss, STL

STP	Overall Transmission Time (s)
Loss	14.5
No loss	11.7

estimated by the Reference configuration. Table 4.14 contains the gain in the same situation. The last two tables (Table 4.15 and Table 4.16) show the overall transmission time for the Modified TCP Reference configuration and the STP, respectively. The tables report the cases with losses and with no losses. It is important to note that the Reference case is heavily affected by the presence of losses, and this is due to the misinterpretation of the loss cause. The STP is robust, and allows good performance in the loss case: the difference among the loss and no loss case is only 13.7%.

Acknowledgements

The author would like to thank colleagues and students of the Italian National Consortium of Telecommunications (CNIT) and of the University of Genoa for the useful discussions during the development of the work. In particular, the author thanks Igor Bisio, Luca Caviglione, Tomaso Decola, Maurizio Mongelli, Marco Perrando and Giancarlo Portomauro of the CNIT Laboratory 'Satellite Multimedia Communications' of the University of Genoa for the work of revision and for their precious suggestions.