A Transport Interworking Protocol for the Support of TCP/IP based Applications over the ESW Satellite Network in the GMBS Environment

L. Secondiani (1), F. Mazzolini (1), N. Blefari-Melazzi (2), M. Femminella (2), L. Piacentini (2)

(1) Alenia Spazio S.p.A., Multimedia Business Directorate,
Via G. Vincenzo Bona 85, 00156 Roma, Italy
Tel: +39 06 4151353, email: l.secondiani@rmmail.alespazio.it; f.mazzolini@roma.alespazio.it

(2) University of Perugia, D.I.E.I., Via G. Duranti 93, 06125 Perugia, Italy
Tel: +39 075 5853630, email: {blefari,femminella,piacentini}@diei.unipg.it

I. ABSTRACT

We investigate the efficient handling of TCP/IP based applications over the GEO satellite segment of the Global Mobile Broadband System (GMBS), namely the EuroSkyWay (ESW) satellite network. We first briefly, summarize the state-of-the art of TCP protocol support over GEO satellite networks; then, we propose a novel approach, based on a "hybrid" inter-working protocol, named IWL-TCP, which operates at transport layer. IWL-TCP interfaces terrestrial IP-based networks with the satellite segment, and directly exploits the bearer services offered by the ESW satellite system. We also define a proprietary interworking Protocol Reference Model (PRM) for the ESW satellite gateways. The way of operation of IWL-TCP will be discussed and its performance analyzed by means of the ns-2 simulation package.

II. INTRODUCTION

II.A TCP issues over GEO satellite links

The earliest version of TCP (Transmission Control Protocol) [2] has not been designed to efficiently support satellite communications, therefore TCP based applications suffer a number of drawbacks in satellite environments, as deeply investigated by the Internet Engineering Task Force (IETF) [6]. Aspects such as large Round Trip time (RTT), asymmetry, and transmission errors jeopardize the behaviour of TCP over GEO satellite links. The introduction of mitigation techniques is mandatory to cope with a so challenging environment.

II.B TCP mitigation for GEO satellite networks

A lot of effort has been carried out by the IETF, and several Request For Comments (RFC) have been issued ([3], [4], [5], and [6]). The outcome of this research activity is represented by the definition of a flavor of TCP, the so called TCP-Sat, where the adoption of enhanced TCP options well suited for GEO satellite links is suggested.

Although in [11] it is evidenced how under certain conditions TCP can operate well over GEO satellite links, in some cases even the adoption of end-to-end TCP-Sat can not guarantee good performances. Since in an actual network with heterogeneous users population, both users and servers can not all be expected to be running satellite-optimized version of TCP, a viable interworking scheme may consist in splitting end-to-end TCP connections into different (both terrestrial and satellite) connection segments, in order to de-couple high-latency or lossy network segments, e.g., the satellite one, from the rest of the network in a way transparent to end-users [7]. Given space limitations, we assume that the reader is familiar with TCP Sat issues and TCP splitting techniques and omit relevant background discussions (in this regard, [15] is a useful reference).

III.ESW INTERWORKING PRM

Taking into account what summarised in section II, we propose a proprietary interworking Protocol Reference Model (PRM) for the M-ESW satellite gateways, which exploits the advanced characteristics of the ESW satellite network [12].

A split inter-working scheme seems to be suitable to both overcome traditional performance problems of TCP over satellite links, and "graft" ESW connection control procedures in the frame of end-to-end TCP procedures. However, it is to be stressed that TCP splitting techniques on the one hand guarantee better performances of TCP over satellite links, but, on the other hand, they threaten the end-to-end semantic of TCP sessions. This is a noticeable drawback for Internet applications, working in an end-to-end context. Therefore, a TCP split interworking scheme seems to be more appealing for applications that require good performance in satellite environment, while at the same time well tolerate the breaking of the end-to-end context, such as web browsing. On the other side, end-to-end inter-working scheme well fit Internet applications requiring preservation of the end-to-end semantic, with no special requirements of fast data transfer service, such as bulk data transfer in a secured end-to-end context. As a consequence, an application-configurable interworking scheme seems to be the better solution for the ESW-Internet interworking scenario, in order to cope with the different typologies of applications.

Traditional split interworking schemes have defined mechanisms to let standard TCP protocols interoperate with transport protocols optimised for satellite network segments (e.g., TCP-Sat). On the other side, since the ESW network will offer reliable satellite connections by implementing a robust ARQ mechanism at link layer and a FEC mechanism at physical layer level [12], the end-to-end TCP virtual connections could benefit from these mechanisms. Thus, it is not necessary to implement a transport layer function on the satellite network segment, to recover losses on the satellite link, as done in traditional schemes.

Such an approach guarantees a quasi-transparent transport mechanism on the satellite link: the ESW link layer service transparently transports the same IP datagrams carried on the terrestrial network segments of the end-to-end Internet path of which the ESW segment is a portion. Note that we do not consider TCP segments as PDUs (Protocol Data Unit) to be transferred by means of the ESW link layer protocol, rather IP datagrams themselves are handled by the ESW link layer protocol, segmented into ESW cells and carried over satellite links, as usual in standard IP implementations.

III.A GMMT Inter-working architecture

The proposed transport interworking protocol allows both an efficient use of satellite communication resources when internet flows are transported, and an improvement of TCP performance as perceived by end-users. Therefore, the functionality related to this interworking protocol is placed in a Quality of Service (QoS) support module of the T-IWU of the GMBS Multi-Mode Terminal (GMMT), and mapped onto proper Functional Entities (FE) [13]. Figure 1 provides a detailed representation of the adopted PRM, by showing the envisaged interworking layers and planes, as well as the interfaces between layers. Protocol entities are distributed among the T-IWU and the In-Door Unit (IDU) of the ESW satellite terminal. ESW Control protocol block (ESW-C), ESW User protocol block (ESW-U) and Inter-Working Synchronisation and Co-ordination Function (IW-SCF) are located into the IDU of the ESW satellite terminal, since these modules are involved only in ESW internal procedures. Moreover, Inter-Working Layer-IP (IWL-IP) has to be implemented on the IDU of the ESW satellite terminal since this module represents a relay entity between the TCP/IP protocol suite and the ESW-U protocol block. As far as Inter-Working Layer TCP (IWL-TCP) and ESW-Internet Protocol (E-IP) are concerned, their functions can be supported only by an upgraded IP router, and consequently are implemented in the T-IWU of the GMMT. Note how IWL-TCP interfaces IW-SCF and IWL-IP, for control and data flows, through a Service Access Point (SAP) for commands and a SAP for data transfer respectively.

III.B IWL-TCP

The connection oriented transport interworking layer, namely IWL-TCP, behaves differently from the standard TCP. While IWL-TCP does not have any interface with an upper level application, as in standard TCP, at the same time it is provided with an extra interface towards an interworking module of the ESW Management plane (ESW-M). Through this interface IWL-TCP requires services to both ESW-U, and ESW-C. From this point of view, IWL-TCP perceives IW-SCF like a lower layer's entity.

III.C IWL-IP

In transporting a given IP datagram over the ESW network, within a suitable number of ESW cells, two possible options that can be considered are: i) the transparent transport; ii) the use of header compression techniques and then again. The second solution has the advantage to reduce the overhead, while the first one can be more easily updated to future versions of IP protocol (for example to IPv6). The adopted solution is the first one. As far as the IWL-IP is concerned, its introduction arises from the necessity, at the ESW receiving terminal, to reconstruct the IP datagram, for transparency purposes. Unlike standard IP, IWL-IP does not perform addressing and fragmentation functions (the latter function being executed at ESW 2-I layer), but it simply re-formats the IP datagram header. Finally, IWL-IP manages the correspondence between TCP sockets and ESW Virtual Channel Identifiers (EVCI) all along the connection lifetime.

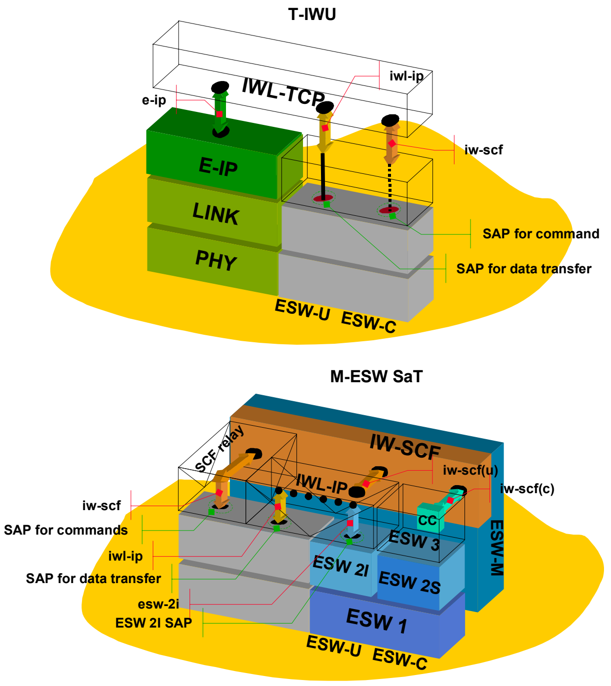

Figure 1: ESW-TCP/IP suite PRM

III.D E-IP

The M-ESW gateway will implement Internet routers functions, properly upgraded in order to handle data flows coming from and going to ESW Network. This entails an impact on the IP layer of such ESW gateways/routers. For this reason this layer is denoted as E-IP. E-IP performs the following functions, on datagrams going toward the ESW network,:

- standard IP routing (next hop selection),
- datagram reassembly (if necessary),

- datagram header processing, and
- data portion delivery to IWL-TCP.

As far as datagrams coming from ESW network are concerned, E-IP performs the following functions:

- segment reception from IWL-TCP,
- datagram header reconstruction,
- next hop selection, and
- fragmentation (if necessary).

III.E IW-SCF

From the terminal protocol stack standpoint, IWL-TCP asks for both ESW-U, and ESW-C services to IW-SCF; in this framework IW-SCF synchronises and translates primitives received from IWL-TCP onto primitives directed to both IWL-IP and ESW Connection Control module (CC) and vice versa. The need for this module arises from the necessity of co-ordination between processes of CC entity (ESW-C plane) and IWL-IP entity (ESW-U plane).

IV. IWL-TCP

This paragraph describes the main features of the connection oriented transport interworking layer IWL-TCP. Since it has been derived from standard TCP implementation [2], its main functionality and interfaces will be described by stressing differences with respect to standard TCP implementations.

IV.A Control plane Inter-working

In Figure 2 the Finite State Machine (FSM) of the IWL-TCP is sketched. Shadowed states have been introduced in order to “graft” ESW connection control procedures into the end-to-end TCP “three way handshake” procedure, while the remaining states have been derived from the corresponding states of standard TCP FSM.

The main features of IWL-TCP, and its main differences with standard TCP, are summarised below.

1. IWL-TCP does not provide, unlike standard TCP, any service to an upper level TCP/IP application.
2. The source and destination sockets are used by IWL-TCP in order to distinguish among the set of active TCP connections the one relevant to each incoming/outgoing segment.
3. IWL-TCP, like standard TCP, initialises and maintains status information for each handled data stream. Each TCP session is uniquely specified by the couple of sockets relevant to the end-to-end TCP session.
4. IWL-TCP handles any kind of error (internal errors, IWL-TCP errors, ESW errors and end-host’s TCP errors) that may occur.

From the IP end-host point of view, its TCP layer will seamlessly dialogue with the other end host, even if some header fields (concerning flow control and reliability, like acknowledgement number and window) of the received segment are filled by IWL-TCP. In other words the TCP layer of an Internet end-host is unaware of the dialogue it is actually maintaining with IWL-TCP of either a GMMT or a M-ESW gateway.

Figure 3 shows the whole connection set-up procedure relevant to a TCP session seamlessly transported over the ESW network. It is to be stressed that the ESW connection set-up procedure is nested into the end-to-end TCP synchronization procedure. Finally note as TCP segments relevant to such procedure are transparently handled by ESW gateways without spoofing the emitter IP end-host.

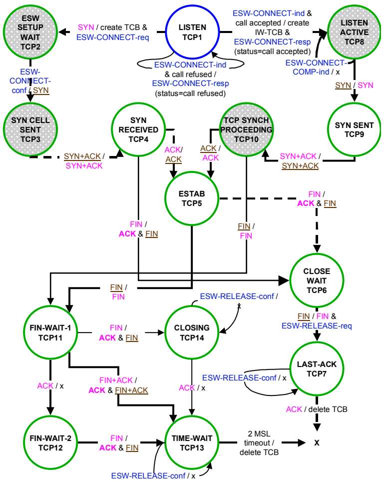

Figure 2: IWL-TCP FSM

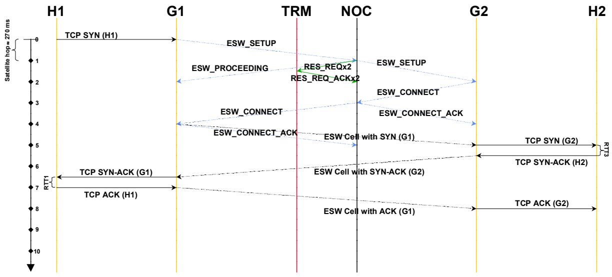

Figure 3: Interworking connection set-up phase

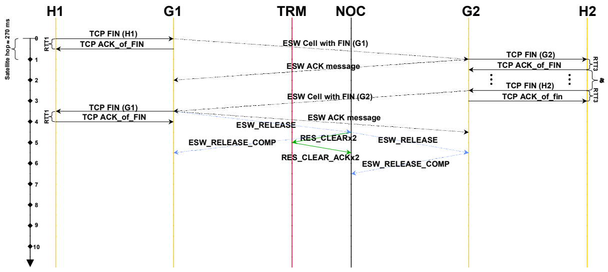

Figure 4: Interworking connection closing phase

Figure 4 sketches the overall connection closing procedure. Note how the ESW connection closing procedure is nested into the end-to-end TCP connection closing procedure as well, but in this case the relevant TCP segments are immediately acknowledged by the receiving M-ESW gateway.

IV.B User plane Interworking

As shown in Figure 5, the M-ESW gateway/proxy implements connection splitting at IWL-TCP layer level, to shield end-users from the effect of delay and bit errors on the satellite link. The proxy is a transparent active TCP Performance Enhancing Proxy (PEP), intended to be deployed at the ends of the satellite hop [7]. This way, the two proxies split the end-to-end TCP connection into three separate "connection segments". For example, by considering data flowing from end-host H1 to end-host H2, the IWL-TCP of the M-ESW gateway (G1) fakes the IP address of the remote end-host (H2), acknowledges the opposite end-host (H1) as if it was the other end-point of the connection, and stores data to be transferred to the other M-ESW gateway (G2). The same considerations can be made for the opposite direction as well.

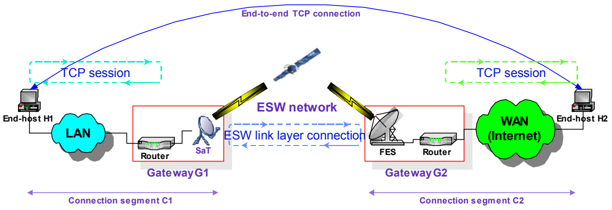

Figure 5: ESW-TCP connection splitting scenario

To summarize, IWL-TCP behaves as described below, on the user plane.

1. IWL-TCP exchanges segments with two peer entities: the standard TCP of the opposite IP-end-host, and the IWL-TCP module of the interfaced M-ESW gateway involved in the ESW connection.
2. For data transfer purposes, IWL-TCP exploits services offered by both E-IP and IWL-IP.
3. As far as outgoing segments are concerned, IWL-TCP usually does not format TCP segments. Let us consider two different cases:
    - segments sent to the M-ESW gateway: IWL-TCP does not fill any header field of such segments. As concerns output process, IWL-TCP exploits the acknowledged data transfer service offered by IWL-IP. In fact, data transfer reliability and flow control on the satellite segment are performed at ESW layer 2-I level.
    - segments sent to the opposite end-host: IWL-TCP usually does not fill any header field of these segments except acknowledgement number, ACK control bit and Window. As concerns output process, IWL-TCP performs all necessary functions to provide reliability and flow control during data transfer.
4. As concerns incoming segments, IWL-TCP differently handles segments coming from Internet end-hosts and segments coming from the other GMMT relevant to the ongoing ESW connection.
    - Segments coming from IWL-TCP of the M-ESW gateway: neither ACK's nor retransmission algorithm working at transport level are required. From the input process point of view, for each incoming segment, IWL-TCP checks whether a transition in the Finite State Machine (FSM) of the session the segments belong to occurred.
    - Segments coming from TCP of the opposite end-host: IWL-TCP will have exactly the same behaviour of standard TCP. Furthermore IWL-TCP, like standard TCP, performs a suitable algorithm to calculate the Acknowledgement number and to update the receive window size. IWL-TCP, like standard TCP, may receive from the opposite end-host segments out of sequence which are reordered before transmission. In other words IWL-TCP forwards in-sequence segments to IWL-IP, as standard TCP delivers in-sequence segments to application layer.

V. SIMULATION RESULTS

With reference to Figure 5, we have considered a simulation in which a generic host (H1) in a LAN connected to the ESW system is downloading a file from a remote ftp server (H2). Starting from this reference scenario, our performance study has been based on the following assumptions:

1. protocol stack modifications should be avoided in end-user systems, in order to hide the presence of satellite and to make its transport facility available to a generic Internet host;
2. the overall splitting scheme has to guarantee the information integrity of data transfer (as in standard TCP implementations); this means that a suitable algorithm has to be implemented in the gateways, to regulate the emission of acknowledgements and the advertised window size so as to avoid buffer overflows.

Let us consider the satellite gateway facing the sender, G2, (see also Figure 5), whose schematic architecture is shown in Figure 6. We employed the TCP flow control to fully exploit the ESW transport capabilities. The algorithm used to regulate the emission of acknowledgements and the advertised window size so as to avoid buffer overflows is derived from that proposed in [14]. The buffer B located on the satellite gateway G2 is at least great as the maximum value of the advertised window size of the TCP (i.e., in standard implementation Max_W=64 Kbytes). The remaining part of the buffer, labelled $TxBufferSize$ in Figure 6, is devoted to accommodate the TCP segments that have been sent on the satellite link but not yet acknowledged (B_in_fly). Obviously, such bytes are immediately acknowledged by the satellite gateway to H2. In order to obey to the constraints mentioned above, the TCP advertised window communicated from G2 to H2 is set to the value:

$$Advwnd=min\ (Max\_W,\ B-B\_in\_fly). \qquad (1)$$

Note that we do not exploit any information regarding the other two segments composing the end-to-end TCP connection (i.e., information coming from G1 and H1), but only the information locally available in G2. If such information would be available, for example by means of a parse and capture procedure of flow control information over the others two segments of the end-to-end connection, the advertised window could be raised to higher values than (1). However, a number of considerations, first of all the fact that the satellite link can not be considered surely error-free, even if ARQ and FEC schemes are present, suggests to use (1) as a safe choice.

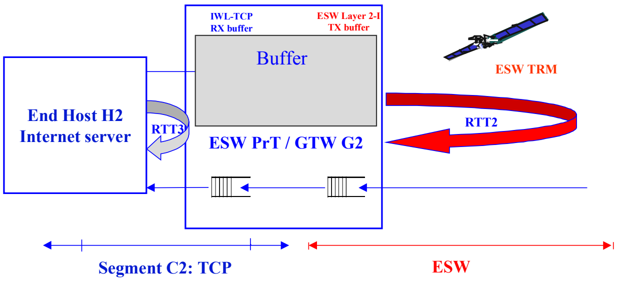

Figure 6: Gateway section at sender size

To evaluate performance, we used the simulation package ns-2 (see also [15]). We assumed an error-free satellite channel with variable capacity in the range 500 Kbps – 10 Mbps. The speed of terrestrial links was set to the same value of the satellite link. The size of the buffer *B* was fixed to 1 Mbytes, while two different sizes of the files to be transferred from the ftp server have been considered: 2 Mbytes and 30 Mbytes. The payload of the IP datagram was set to 1 Kbytes. Transfers of files with dimension smaller than *B* have not been simulated, because they simply would have been stored in the gateway buffer and then forwarded towards the remote host. However, the TCP splitting scheme is interesting also for files of small sizes, because, in these cases, this scheme is able to transfer information avoiding the initial latency that affects end-to-end implementations (also those with values of *Max_W* higher than 64 Kbytes, that usually behave well with files of big size). In Figure 7 and Figure 8 we compare the throughput of a "classical" end-to-end implementation with that of our splitting scheme as a function of the link capacity and for two different size of the files (2 Mbytes in Figure 7 and 30 Mbytes in Figure 8). Also plotted is the maximum theoretical throughput achievable, (i.e., the bisector, corresponding to a 100% throughput efficiency). As expected, the performance of our scheme is closer to the maximum theoretical value (this effect is obviously more evident in the case of 30 Mbytes file transfer), while the throughput of the end-to-end implementation remains under its theoretical limit of 1 Mbps.

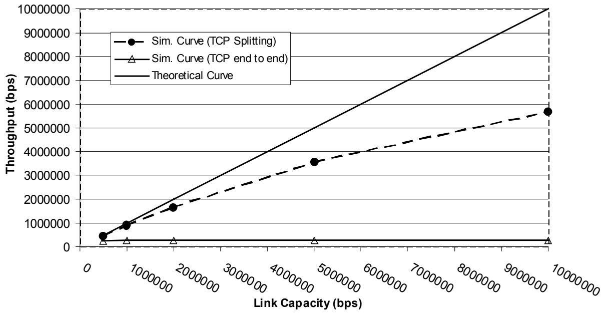

Figure 7: Throughput vs. link capacity (file transfer of 2 Mbytes)

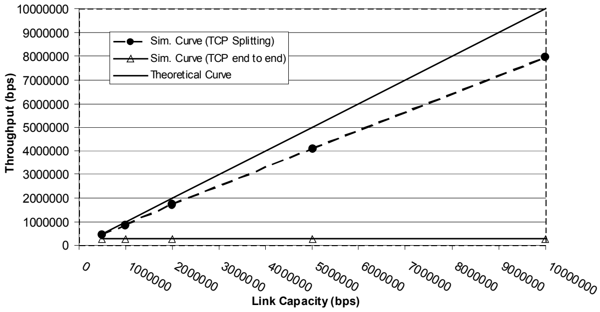

Figure 8: Throughput vs. link capacity (file transfer of 30 Mbytes)

REFERENCES

[1] RFC 791, J. Postel, Internet Protocol, Sept. 1981.

[2] RFC 793, J. Postel, Transmission Control Protocol, September 1981.

[3] RFC 1323, V. Jacobson, R. Braden, D. Borman: "TCP Extensions for High Performance", May 1992.

[4] RFC 2414, M. Allman, S. Floyd, C. Partridge: "Increasing TCP's Initial Window", September 1998.

[5] RFC 2488, M. Allman, D. Glover, L. Sanchez: "Enhancing TCP Over Satellite Channels using Standard Mechanisms", January 1999.

[6] RFC 2760, M. Allmann, et al,: "Ongoing TCP Research Related to Satellites", February 2000.

[7] RFC 3135, J. Border, M. Kojo, J. Griner, G. Montenegro, Z. Shelby: "Performance Enhancing Proxies", June 2001.

[8] V. J. Bharadwaj: "Improving TCP Performance over High-Bandwidth Geostationary Satellite Links", CSHCN MS 99-7, 1999.

[9] V. J. Bharadwaj, J. S. Baras, N. P. Butts: "Internet Service via Broadband Satellite Networks", CSHCN TR 99-11, 1999.

[10] V. J. Bharadwaj, J. S. Baras, N. P. Butts: "An Architecture for Internet Service via Broadband Satellite Networks", CSHCN TR 99-12, 1999.

[11] T. R. Henderson: "Transport Protocols for Internet-Compatible Satellite Networks", IEEE JSAC, Vol. 17, No. 2, February 1999.

[12] G. Losquadro, V. Podda, F. Carducci: "The EuroSkyWay Worldldwide system providing broadband service to fixed and mobile end-users", 4th Ka-band Utilisation Conference, Venezia, Italy, November 1998.

[13] Project IST 1999–10469 SUITED; Deliverable D7: "GMBS Terminal Specifications and Design", 2001.

[14] A. Baiocchi, N. Blefari-Melazzi, M. Listanti, M. Soprano: "Definition and performance analysis of a simple, ABR-like congestion control scheme for satellite ATM networks with guaranteed loss performance", IEEE JSAC, Vol. 17, N. 2, Feb. 1999

[15] M. Karaliopoulos, R. Tafazolli, B.G. Evans: "TCP Performance on Split Connection GEO Satellite Links", 19th AIAA International Communications Satellite Systems Conference, Toulouse, France, 17-20 April 2001.