LIFT: a Local IPSec-aware Freezing Protocol to improve TCP Performance in Satellite Networks

Giovanni Ciccarese, Mario De Blasi, Sebastiano Elia, Cosimo Palazzo, Luigi Patrono

***Abstract*: In this paper a protocol, local to the satellite link, is defined in order to boost TCP performance in mobile integrated wired-satellite Internet. It has been conceived to help to overcome the well known retransmissions competition problem that arises when a satellite reliable link layer protocol is used to face satellite link errors. This protocol, called Local IPSec-aware Freezing proTocol (LIFT), has been designed to allow the satellite gateway, even in the presence of communications secured by IPSec, to freeze the TCP sender when it perceives a possible delay due to satellite channel conditions. The effectiveness of LIFT has been evaluated, using the ns-2 tool, in terms of Web page download mean time for a satellite mobile host. Simulation results have shown that the adoption of LIFT protocol provides substantial improvements in TCP performance.**

***Index terms*: Satellite, Performance, Mobility, TCP Freezing, ARQ, IPSec.**

I. INTRODUCTION

In recent years, the research community has been driving Internet evolution towards an internetwork able to support wireless and mobile communications. A set of wireless technologies have been considered for enabling mobile computing in Internet. Among these, satellites certainly represent an attractive transmission medium, but some problems need resolving. Satellite networks, in particular those which are geo-stationary, are characterized by a high bandwidth-delay product, a high bit error rate (BER) and burst errors mainly due to shadowing and multipath fading related to mobility. These characteristics may cause poor performances in TCP protocol because of its congestion control mechanism: a TCP sender interprets any packet loss as a sign of congestion and reacts by slowing down the traffic sent over the connection, even if losses are due to transmission errors.

A number of schemes [1-8] have been proposed in order to deal with TCP performance problems in wireless networks, but only a few of them are applicable and effective in a mobile satellite scenario and when IPSec encryption is used. Some recent works [4] have proved that an accurate design of a reliable data link protocol represents an attractive way to face the problems related to TCP over wireless. In particular, in [9, 10] it was shown that, simply by using an ARQ link layer protocol well suited to satellite channel characteristics, TCP performance improves much more than by adopting an end-to-end solution such as SACK [8] option of TCP. Recently, another transport protocol, the Stream Control Transmission Protocol (SCTP) [11, 12], has emerged, that could overcome some problems of TCP in a mobile scenario. For this reason, it will be considered in this work. Let us observe that the main advantage of local solution based on ARQ is that its implementation is confined to the satellite link at data link layer, avoiding both any TCP software modifications and any interference with encryption schemes as IPSec. On the other hand, the unique drawback of this local approach also, highlighted in [10], is the well known retransmissions competition problem between ARQ mechanisms at transport layer and data link layer. In order to mitigate this problem, taking into account the contribution of some proposals, such as TCP-Freeze [6], thought to allow the Mobile Host (MH) to freeze the TCP sender on Fixed Host (FH), a particular Performance Enhancing Proxy (PEP) was subsequently defined in [13]. Let us observe that the originality of this solution is to have moved the possibility to freeze the TCP sender on satellite gateway. This has made it possible to bypass the inability of TCP freezing solutions to apply to satellite networks mainly due to the high propagation delay on the satellite link. Briefly, the PEP defined in [13] runs on a satellite gateway, snoops all TCP segments in transit and tries to freeze TCP by means of a "freezing" ACK segment, built by forcing to zero the WINDOW field of the last TCP ACK coming from MH and buffered temporally on the Base Station (BS), whenever the time needed by the reliable data link protocol on satellite channel to deliver data may cause an useless TCP timeout. In [13] it was shown that TCP performance substantially improves by combining an ARQ data link protocol on satellite link with this PEP, contributing to render access to web services attractive in every mobility scenario. This PEP requires no changes in TCP, maintains end-to-end semantics and is able to mitigate the retransmission competition problem, but it has one drawback: it no longer works if, due to network encryption schemes such as IPSec, TCP segments flowing through the satellite gateway cannot be snooped and modified.

Taking into account the lack of this PEP, the research work summarized in this paper introduces a novel protocol, called Local IPSec-aware Freezing proTocol (LIFT), which serves to lift TCP performance when it is combined with a reliable link layer protocol running on a satellite link, overcoming the IPSec constraints. It enables the satellite gateway to carry out

Manuscript received December 31, 2005 and revised November 6, 2006.

Giovanni Ciccarese, Mario De Blasi, Sebastiano Elia, Cosimo Palazzo and Luigi Patrono are all with the Dept. of Innovation Engineering, University of Lecce, Via Monteroni, 73100 Lecce, Italy, (e-mails: {gianni.ciccarese, mario.deblasi, sebastiano.elia, cosimo.palazzo, luigi.patrono}@unile.it.)

monitoring of TCP connections and TCP freezing without violating IPSec goals: confidentiality, authentication, data integrity and availability. Indeed, LIFT entity on MH snoops TCP segments on behalf of its peer entity on the satellite gateway, sends it the result of snooping, and accompanies each TCP ACK with an additional "freezing ACK" obtained by setting the WINDOW field to zero. This segment can be exploited by the satellite gateway to force the TCP sender to go into *persist* mode whenever a possible delay in the reliable data delivery on satellite link is perceived.

Through computer simulation, the effectiveness of LIFT has been evaluated in terms of Web page download mean time to a satellite Mobile Host. Moreover, the additional bandwidth required by LIFT has been calculated in order to assess how convenient its employment is with respect to the simpler approach based only on a reliable link layer protocol.

Finally, the performance achievable using LIFT with a reliable satellite data link has been compared with that related to a Web scenario which exploits SCTP running on an unreliable data link layer, which is endowed with some features, not available in TCP, which could improve the performance of a satellite network.

The rest of the paper is organised as follows. The proposed LIFT protocol is presented in Section II. Section III describes the effects of LIFT on IPSec security. Section IV reports the network model under consideration. In Section V simulation results are discussed. Finally, conclusions are provided in Section VI.

II. The LIFT Protocol

The main goal in designing Local IPSec-aware Freezing proTocol (LIFT) has been to provide a complete solution able to improve TCP performance in a satellite mobility scenario (Fig. 1), when a reliable satellite data link protocol is adopted to face effectively and locally the errors on the satellite link and when, moreover, communications are secured by IPSec. Specifically, the two key requirements which have been considered when defining LIFT are:

- *To allow the satellite gateway to freeze the TCP sender,* in order to mitigate the competition problem between ARQ mechanisms at transport layer and data link layer, even when IPSec is used (in end-to-end mode). In an integrated satellite scenario supporting the user mobility, the satellite channel is often characterized by off periods, due mainly to shadowing, whose existence cannot be predicted and communicated effectively by mobile hosts to fixed hosts (for example by freezing the TCP sender) because of the high propagation delay of geostationary satellite links. To allow an intermediate node such as a satellite gateway to trigger the TCP freezing is complicated using IPSec encryption.

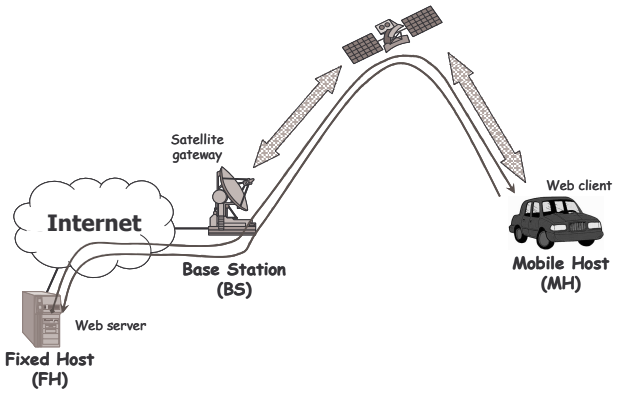


Fig. 1. A mobile host accesses an Internet fixed host through a satellite link.

- *To allow the satellite gateway to become acquainted with some information contained in the TCP header,* needed for freezing the sender, even when segments are encrypted by IPSec. Ensuring this, without compromising IPSec's benefits, has represented the main challenge faced by LIFT.

LIFT protocol is able to work with both encrypted and non-encrypted traffic, simply by switching between two modes: ENCRYPT and CLEAR. In CLEAR mode, the PEP [13] previously described is adopted. On the other hand, when the satellite gateway, in the role of Base Station (BS), observes that a IPSec Security Association (SA) has been established, the LIFT will decide to switch to ENCRYPT mode. In the rest of the paper, the ENCRYPT mode will be considered. Note, also, that LIFT can operate whether the Authentication Header (AH) protocol [14] or the Encapsulating Security Payload (ESP) protocol [15] is used; in the following, however, it is assumed that ESP in transport mode is adopted. Another assumption has been that TCP sender exploits only one retransmission timer as suggested in [16, 17].

LIFT, operating at network layer, manages the communication between two LIFT entities residing on BS and Mobile Host (MH), called BS-LIFT and MH-LIFT respectively. To operate, LIFT uses a header, whose position is shown in Fig. 2.

The MH-LIFT can easily analyze and, if necessary, modify the TCP segments before encryption. Hence, the MH-LIFT snoops the segments on behalf of the BS-LIFT, sends it the resulting information by filling some fields in the LIFT header, and accompanies each TCP ACK with an additional "freezing ACK" that the BS-LIFT can use when necessary. MH-LIFT builds the "freezing ACK" by cloning the TCP ACK segment and making the following changes: setting to zero the WINDOW field and removing the payload. The "freezing ACK" will be referred to using the short form

Fig. 2. Addition of LIFT header to an IPSec packet.

FrACKs to differentiate it from an unaltered TCP ACK segment. Let us observe that both ACKs will be regularly encrypted by ESP protocol and that, before being passed to IP entity, they will also be encapsulated in LIFT packets. Let us assume that only the delivery of packets containing TCP ACKs on satellite link is reliable, whereas that of FrACKs is unreliable for two reasons: to avoid slowing down TCP ACK delivery, and to keep the overhead on the satellite channel contained. In addition, the ACKs will be sent respecting a predefined order: FrACK before unaltered TCP ACK segment.

Generally, the BS-LIFT does not immediately propagate anyone of the two members of the pairs (FrACK, TCP ACK) when it receives them from MH-LIFT, but stores them for a certain time. When the BS-LIFT intercepts a new pair of ACKs, this one will replace the pair received and maintained previously on the BS, and one of the two members of the old pair will be forwarded to the Fixed Host (FH). When the channel conditions are good and therefore the reliable satellite data link protocol does not delay data delivery, the BS-LIFT will decide to forward the previous TCP ACK towards the FH. On the contrary, whenever the arrival of the TCP ACK of a new pair is late, because of bad conditions on the satellite channel, the BS-LIFT will try to freeze the TCP sender by forwarding the FrACK buffered previously on BS and keeping the TCP ACK, just buffered, in order to have the possibility to unfreeze the TCP sender. Indeed, this packet will be forwarded as soon as a fresher TCP ACK is intercepted.

The BS-LIFT is able to perceive a possible delay, due to channel conditions, because it manages a local timer, reset whenever a TCP ACK, that acknowledges new data, is forwarded to FH, whose expiration value is slightly lower than the minimum Retransmission Timeout (RTO) of TCP. In any case, the BS-LIFT will decide to forward a FrACK when a timeout occurs, only if the BS has still not intercepted an ACK that acknowledges all TCP data forwarded towards the MH. This contributes towards avoiding useless TCP freezing when there are not segments in flight on the connection. To this end a procedure is used to locally acknowledge the IPSec packets that are in transit through the BS towards the MH and that carry TCP data segments. This procedure exploits, in an original way, the values of Sequence Number field of IPSec packets, thought to avoid "replay" attacks, as sequence numbers for local IPSec acknowledgment by LIFT.

A. Setup and Teardown procedures

As BS-LIFT sees only SAs, it is necessary that MH-LIFT communicates to it a mapping of SAs (only downlink SAs) with TCP flows. More specifically, whenever a new TCP connection is established between MH and FH, the MH-LIFT generates a number, two bytes long, called the *TCP Flow Identifier*, that, together with the IP address of MH, uniquely specifies the established connection in LIFT. MH-LIFT will invoke a setup procedure in order to communicate this to BS-LIFT. For this phase, the two LIFT entities will use pure control packets, referred as *Setup Packets*, whose LIFT header is shown in Fig. 3A. LIFT control packets can be distinguished from LIFT packets encapsulating ACK segments by means of the *A* field, while the *T* field defines the particular type in each class. Further information is contained within the header of a LIFT Setup Packet, such as the Security Parameters Index (SPI) and the Security Protocol Identifier of the downlink IPSec SA. A reliable service is required for the delivery of LIFT Setup packets. The BS-LIFT will exploit the values put in the LIFT header to update the *TCP Flow Table* (Fig. 4) with a new entry. This table allows the BS-LIFT to associate each TCP connection with the corresponding downlink SA. Let us observe that in order to complete the entry, the BS-LIFT will also have to use the IP address of the MH reported within IP header of the same packet.

Another case that triggers sending a LIFT control packet occurs when the MH-LIFT intercepts a TCP RST segment sent by FH. In this case, the MH-LIFT builds and sends to the BS a *Teardown Packet*, with LIFT header shown in Fig. 3B, in which the TCP Flow Identifier indicates the related TCP flow. The delivery of a Teardown Packet is reliable.

Instead, whenever the MH-LIFT has to send LIFT packets carrying ACK segments (TCP ACK or FrACK), it uses a different format of LIFT packet, shown in Fig. 5 and described in the next subsection.

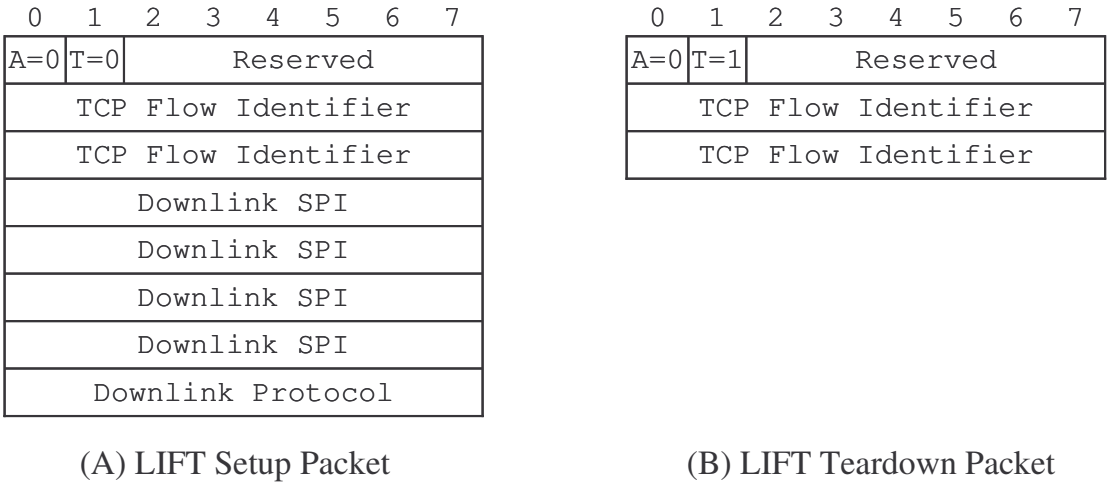


Fig. 3. Header formats of LIFT control packets.

TCP Flow Identifier (16)	MH IP Address (32)	Downlink SPI (32)	Downlink Protocol (8)

Fig. 4. TCP Flow Table of BS (sizes in bits).

Once the setup phase is over, the core of LIFT protocol goes about the *Freezing* and the *Local Acknowledgement* procedures, which are also detailed in the following subsection.

Special treatment is reserved for the case in which the BS intercepts an out-of-order IPSec packet over a given downlink SA, due to possible packet losses over the wired portion of the network. In this case, the BS-LIFT will decide to disable the freezing procedure for all TCP flows established over this SA in order to give way to the TCP congestion control mechanisms. The freezing procedure will be re-activated as soon as all out-of-order IPSec packets have been acknowledged.

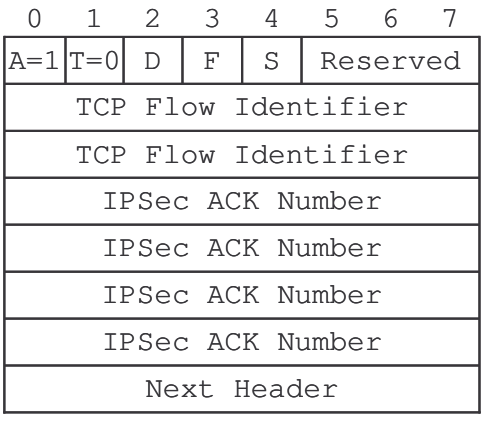
B. Freezing and Local Acknowledgment procedures

The MH-LIFT begins to prepare the LIFT headers of packets carrying ACK segments as soon as the TCP entity delivers a TCP ACK to network layer, and completes them after the IPSec encryption has been made. These are characterized by having the value of *A* field equal to 1 and *T* field defining the particular ACK (TCP ACK or FrACK). Let us observe that the MH-LIFT starts to build the additional freezing ACK (FrACK) if the ACK segment received from the TCP entity, with the WINDOW field greater than zero, acknowledges new data or is piggybacked. Both the TCP ACK and FrACK segments are packaged in two different LIFT packets by using the headers shown in Fig. 5A and Fig. 5B respectively. In these headers, the MH-LIFT inserts, in any case, the *TCP Flow Identifier*. Further flags are also configured by MH-LIFT in LIFT packets containing TCP ACKs to assist BS-LIFT in the storing and forwarding phases of encrypted packets containing a TCP ACK or a FrACK. These are:

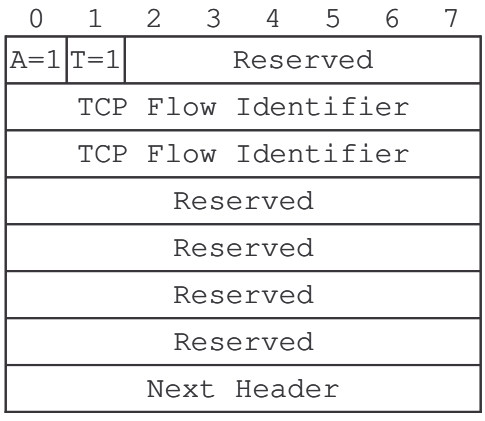
- *D (Delete)*: it is set to indicate that the entry of TCP Flow table corresponding to specified TCP Flow identifier has to be deleted. This happens, for example, when the TCP ACK confirms a TCP FIN segment or identifies a Reset (RST) segment.
- *F (Forward)*: it is used to tell BS not to delay the forwarding of the received TCP ACK. Some events that trigger this setting are: the TCP ACK is piggybacked, TCP ACK has a WINDOW field set to zero, TCP ACK serves to unfreeze the TCP sender or confirms a TCP FIN segment.
- *S (Store)*: it is used to indicate to BS that TCP ACK forwarding has to be delayed if the correspondent local timer is running and consequently the buffered FrACK is still able to freeze TCP. This happens for example when the TCP ACK is a duplicate.

In order to prepare full compatibility between the LIFT and the other protocols at network layer, both LIFT headers have a *Next Header* field used for the multiplexing/demultiplexing procedures.

The LIFT implementation is able to avoid unprofitable effects on the TCP performance in transitory phases that occur at the activation or re-activation of freezing procedure (such as initial TCP Slow-Start). Each time the freezing procedure is activated, the BS-LIFT will forward all TCP ACKs coming from MH, without starting the timer, up to the first one acknowledging new data. This TCP ACK will also be propagated and the timer will be started. Since this moment on, if a new TCP ACK, not piggybacked, acknowledging new data, arrives and the timer is running, the TCP ACK will be



(A) LIFT header for a TCP ACK



(B) LIFT header for a FrACK

Fig. 5. Header format of LIFT packets carrying ACK segments.

stored in the buffer together with the corresponding FrACK, otherwise the transitory phase will continue.

By means of the Local Acknowledgement procedure, BS-LIFT knows whether the last intercepted TCP ACK has acknowledged all TCP data segments forwarded towards the MH until that time. Specifically, whenever the MH-LIFT receives an IPSec packet, it keeps track of the value contained in the ESP Sequence Number field. Furthermore, after the decryption phase, MH-LIFT holds the TCP Sequence Number if a TCP data segment is found. This operation is fundamental because it allows the MH-LIFT, when it intercepts an ACK segment that has to be sent to FH, to take into account the highest ESP Sequence Number of IPSec packets that encapsulated data segments acknowledged with this ACK segment. The MH-LIFT writes the value of the saved ESP Sequence Number in the *IPSec ACK Number* of the LIFT header (Fig. 5A). Naturally, the MH-LIFT will have to complete the header formatting phase before passing this LIFT packet to the IP entity for sending to FH.

If the IPSec ACK Number of a TCP ACK received by BS-LIFT is lower than the ESP Sequence Number of the last IPSec packet forwarded towards the MH on the SA corresponding to the TCP Flow Identifier carried by the TCP ACK, the BS-LIFT will assume that it has still not intercepted a TCP ACK that acknowledges all data segments forwarded towards the MH.

Finally, note that, over a downlink SA, the BS-LIFT is not able to distinguish if an IPSec packet carries TCP data or a TCP ACK segment not piggybacked. Furthermore, the same lack of ability of the BS-LIFT is related to recognizing the IPSec packets containing segments belonging to a specific TCP flow (if more flows are on a SA). This could involve FrACK forwarding even when a TCP ACK that acknowledges all data has already been intercepted, i.e. a temporary abnormal behaviour of the freezing procedure, soon recovered at the next ACKs.

III. LIFT EFFECTS ON IPSEC SECURITY

The suite of IPSec protocols and associated default algorithms was designed to provide high quality security for Internet traffic guaranteeing interoperability with IPv4 and IPv6. The set of security services offered includes access control, connectionless integrity, data origin authentication, protection against replays and confidentiality. It is well known that these services are provided at the IP layer, offering protection for IP and/or upper layer protocols and operating independently of the application (for example, web, email, file transfer, VoIP, etc.) that may use it. Naturally, these features and services have contributed significantly to increasing the popularity of IPSec. For this reason, every proposal thought to act in an Internet scenario should take into account the constraints imposed by IPSec.

LIFT has been designed to enable the satellite gateway to carry out monitoring of TCP connections and TCP freezing without violating IPSec goals.

Let us observe that the values carried within the LIFT header are totally different from any information generally encrypted by ESP in transport mode. More specifically, no values of TCP header (i.e. Sequence Number, Acknowledgment Number, Port Number, Window, flags, etc.) is copied in the LIFT header. Note that the *TCP Flow Identifier* values are numbers generated locally by the MH-LIFT and interpreted exclusively by the BS-LIFT. However, in order to assure that transmission between MH and BS of LIFT packets, header included, occurs in protect mode, let us assume that suitable mechanisms at the network layer (i.e. IPSec tunnelling) or at data link layer (i.e. satellite security protocol) are adopted. In this way, every attack could be excluded over satellite link. Furthermore, the FrACK segments, before being encapsulated in LIFT packets, are regularly encrypted by ESP protocol.

The only possible weak point is related to actions performed by the BS-LIFT entity. More in detail, an intruder could inadequately change the typical behaviour specified by LIFT. Naturally, an efficient protection system is desirable in order to avoid similar risks (for example, Trojan horses, viruses, etc.). Note that a possible attack could force the BS-LIFT to forward all FrACK segments respecting no LIFT rules. In this case, the TCP sender on FH, associated to a precise flow, would be constantly frozen also during "good" periods of the satellite channel. Let us observe that this drawback, however, does not impact on the security services (confidentiality and integrity of user data) guaranteed by IPSec on Internet traffic. Whenever the BS-LIFT loses control of the forwarding and holding activities, the interested connections could be closed by the correspondent TCP sender. In this case, the poor quality of service perceived by the end-user results in an inaccessibility to the Internet server. Note that the main disadvantage relapses exclusively on a single flow, and so the LIFT does not generate any strong irregularity for Internet servers.

In order to face the problem related to repeated and forced FrACK forwarding that results in user disconnections, the MH-LIFT reacts by temporarily inhibiting the LIFT protocol and trying to discover the cause of the current disconnection: "bad" satellite channel conditions or attack of an intruder on the BS. If the MH-LIFT continues not to receive data when the LIFT is off, it will debit the service lack to errors on satellite link. In this case, MH will decide to re-activate the LIFT protocol. Otherwise, MH will decide to maintain out of use LIFT protocol until the possible attack is solved.

IV. SIMULATION MODEL

In order to evaluate the effectiveness of the LIFT protocol, secure Web transactions, by using ESP, in an integrated satellite-terrestrial network like that illustrated in Fig. 1, have been simulated by using Network Simulator v2 tool. Although the use of the SACK option is more and more widespread in Internet, the TCP Reno has been adopted conservatively for this analysis.

Recently, a new reliable transport layer protocol, SCTP [11, 12], was standardized by IETF. This protocol introduces some features, not available in TCP, which could improve the performance of a satellite network in the presence of Web traffic. For instance, multi-streaming can reduce latency, allowing data from an upper layer application (i.e. HTTP) to

be split into multiple streams whose reliable delivery is managed independently, thus alleviating the head-of-line blocking effect. In addition, multi-homing can make a satellite network highly reliable and fault tolerant by exploiting the possibility to span an end-to-end connection across multiple IP addresses. Taking into account these features, a performance comparison with an end-to-end approach based on SCTP has also been performed.

Let us assume that an end-to-end connection, between FH and MH, is mapped over a pair of downlink and uplink IPSec SAs. Furthermore, EuroSkyWay (ESW) has been adopted as the reference satellite architecture [9] in order to model the physical and data link layers of the wireless portion. ESW is a connection-oriented network based on a constellation of geo-stationary re-generative Ka-band satellites and exploits TDMA as up-link access scheme.

With regards to TDMA, the time axis for up-link is subdivided into *Frames*, whose duration is constant and equal to 26.5 ms; a *Frame* consists of *N Frame Unit*s (*FU*), with *N* depending on the terminal type. Each *FU* contains the basic data unit transferred across the ESW network, called an *ESW cell*. An ESW cell is 60 bytes long: 7 bytes are reserved for the header, the remaining 53 bytes constitute the payload.

ESW architecture defines the specifications for an optional ARQ data link protocol [9] able to guarantee a reliable and in-order delivery of ESW cells. It is characterized by an error recovery procedure based on a particular Selective Repeat ARQ mechanism which encodes, in an ESW acknowledgement (ESW-ACK) cell, blocks of contiguous cells received and queued in the receiver window. More specifically, each block is encoded by a pair of values, the former representing the left edge of the block, that is the sequence number of the first cell in the block, and the latter indicating the block length (number of cells in the block). The maximum number of coded blocks in the payload of one ESW-ACK cell is equal to 14. The transmission of one ESW-ACK cell is triggered by one of the following events: a complete packet has been delivered to higher layer or the elapsed time from the last ESW-ACK transmission is equal to *Tack,* whose value is set to 636 ms in ESW architecture.

LIFT packets are encapsulated into IP packets that are then fragmented into one or more ESW cells. The last ESW cell used in the fragmentation of an IP datagram includes a padding if the datagram length is not a multiple of cell payload size. The ESP overhead is assumed equal to 18 bytes: 8 bytes for ESP header, 2 bytes for ESP trailer and 8 bytes for ESP authentication information. Instead, the LIFT header size is equal to 8 bytes.

Note that the size of the FrACK is equal to 66 bytes: 20 bytes for the IP header, 8 bytes for the LIFT header, 18 bytes for ESP information and 20 bytes for the TCP header. Moreover, the value of the timer used for the freezing procedure has been set to 0.98 s, taking into account the minimum retransmission timeout (RTO) of TCP Reno, as recommended in [17].

Furthermore, the simulation model also offers the possibility to evaluate SCTP performance. In this case, the new protocol stack includes neither LIFT nor the satellite reliable data link protocol. The number of data chunks [11] for the SCTP segment and the data chunk size have been set respectively to 2 and to 600 bytes: these values allow us to maximize SCTP performance in most of the mobility scenarios considered.

TABLE I
SIMULATION MODEL PARAMETERS.

Application (HTTP)	
Mean of the main object size	10 KBytes
Std of the main object size	25 KBytes
Mean of the Number of in-line objects	5.5
Std of the Num. of in-line objects	11.4
Mean of the in-line object size	7.7 KBytes
Std of the in-line object size	126 KBytes
Mean of the Think time	39.5 s
Std of the Think time	92.6 s
Transport (TCP and SCTP)	
TCP Header size	20 Bytes
TCP MSS	1224 Bytes
TCP Window	65536 Bytes
SCTP Common Header size	12 Bytes
SCTP Data Chunk Header size	16 Bytes
SCTP Data Chunk payload size	600 Bytes
SCTP Window	65536 Bytes
Maximum number of SCTP streams	10
Network (IP and IPSec)	
IP MTU	1282 Bytes
ESP Header and Trailer size	18 Bytes
LIFT Protocol	
LIFT Header size	8 Bytes
Local timer	0.98 s
Data Link (EuroSkyWay)	
ESW LL-2 Tx Window on client/server side	1003.82 KB
Data link timeout	31.8 s
Frame duration	26.5 ms
ESW cell Header size	7 Bytes
ESW cell payload size	53 Bytes
Bit rate	344 Kbit/s
Tack	636 ms
Physical	
Mobile terminal speed	10 m/s
Propagation time on wireless	265 ms
Propagation time on wired	50 ms

Observe that, in order to make the simulation results related to the two different transport protocols comparable (TCP Reno and SCTP), the Maximum Transfer Unit (MTU) of IP protocol has been set to 1282 bytes, that is the size of an IP packet containing an encrypted SCTP segment with two chunks. Consequently, the Maximum Segment Size (MSS) of TCP is equal to 1224 bytes.

Only the transmission errors due to user mobility have been considered on the satellite channel. In particular, the Lutz model [18] has been adopted for modelling fading due to mobility. Applying ESW system parameters to the Lutz model, a Gilbert-Elliot digital model, that is, a two-state discrete-time Markov chain, has been derived. The two states, called “good” and “bad”, represent, respectively, the unshadowed and the shadowed periods.

The bit rate between MH and FH has been supposed symmetric and equal to 344 Kbps. Finally, it has been assumed that packet losses due to congestion are negligible.

The performance analysis has been carried out by evaluating the *Web page download mean time*. This metric has been calculated as a function of the mean sojourn time in good state, *Tgood*, and the mean sojourn time in bad state, *Tbad*.

All the results of the simulations are characterized by a 95% confidence interval whose maximum relative error is equal to 6%. Finally, Table I summarises the values related to the main parameters of the simulation model.

V. Simulation Results

In order to evaluate the effectiveness of the LIFT, a performance comparison of the TCP Reno using the satellite reliable data link layer with and without LIFT protocol has been performed. More specifically, Fig. 6 reports the Web page download mean time versus Tgood, for a number of values of Tbad; the range 0.5 ÷ 8.5 s has been chosen for both Tgood and Tbad because it covers some interesting environments (urban, suburban and rural) of mobility. As reported in [18], low values of Tbad are related to rural environments, whereas higher values of Tbad are typical for an urban and suburban scenario. The curves show that the LIFT protocol improves TCP performance by reducing the Web page download mean time in any scenario. This reduction, even greater than 20% in some cases, renders access via satellite to Web services attractive even in an urban environment. Note that, in presence of an ideal wireless channel without transmission errors, the Web page download mean time is about 4 s. It is interesting to observe that for some rural environments, for example for Tbad equal to 0.5 s, the contribution of the LIFT protocol is negligible.

In order to better appreciate the improvement introduced by the LIFT, the mean number of TCP timeouts per Web page has been evaluated and reported in Fig. 7. The curves show that LIFT is able to almost completely solve the competition problem in any environment. In particular, the simulation results confirm a substantial reduction in the number of TCP timeouts in urban and suburban environments. Instead, it is interesting to note that for some rural environments, for example for Tbad equal to 0.5 s, the mean number of TCP timeouts is quite low even without adopting LIFT protocol. This is mainly due to the minimum value of RTO of TCP Reno, which is equal to 1 s as recommended in [17]. When the estimated RTO is lower than 1 s, the TCP sender sets the RTO to the minimum value suggested. This implies that the probability that the round trip time of a TCP segment does not exceed RTO is much higher in rural environments. This would also explain the modest improvement achieved by the LIFT in such channel conditions as shown in Fig. 6.

The results reported above have shown that the advantages obtained by adopting the LIFT protocol are, indeed, significant. However, the employment of this protocol involves the use of further resources due to the addition of the LIFT header for each TCP ACK and to the transmission of FrACKs. In order to determine the increase in usage of bandwidth due to LIFT, the mean uplink bandwidth used per Web page download has been evaluated. With regards to this aspect, the comparison between the two configurations with and without LIFT is reported in Fig. 8 in the same conditions of mobility as before. The curves show that the additional uplink bandwidth due to LIFT protocol is about 25% in almost any environment. Nevertheless, note that the overhead due to the LIFT header for TCP ACKs results as being negligible in the reference satellite architecture. Indeed, generally it does not require additional ESW cells after fragmentation, but simply a shorter padding in the last cell.

Finally, a comparison with SCTP running over an unreliable satellite link layer in the same previous channel conditions is reported in Fig. 9. These curves show that the approach exploiting the LIFT protocol greatly outperforms SCTP in almost any mobility environment. In particular, note how the SCTP has poor performance in both urban and suburban environments: the mean time needed to complete an entire Web page download is so high that it renders access to Web services unattractive in these scenarios.

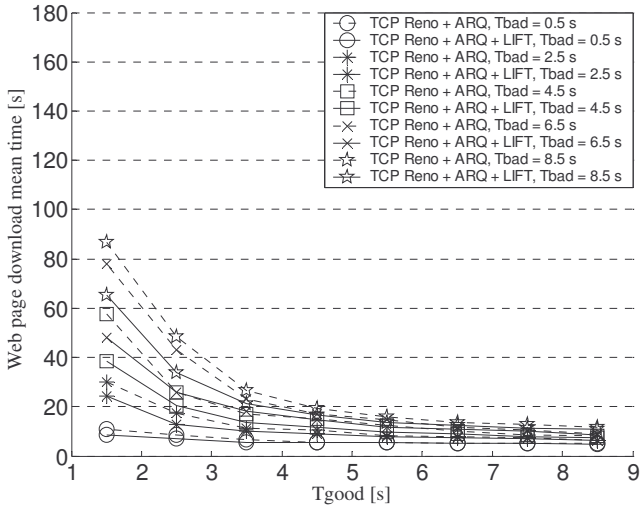


Fig. 6. Performance comparison between TCP Reno over a reliable link layer with and without the LIFT protocol in terms of the Web page download mean time.

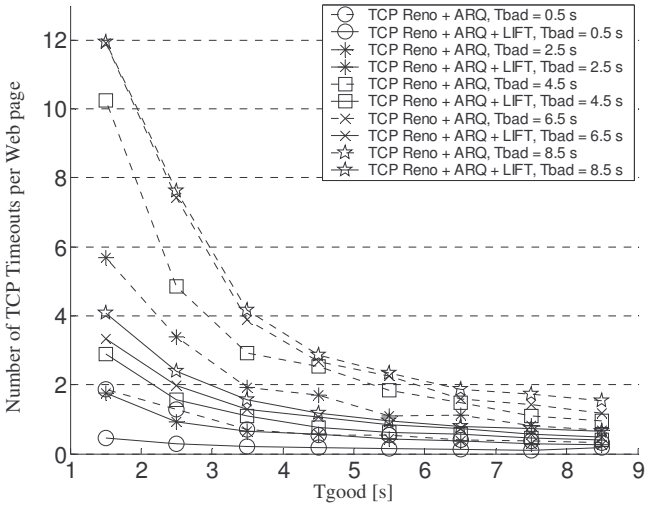


Fig. 7. Number of TCP timeouts per Web page when TCP Reno is over a reliable link layer with and without the LIFT protocol.

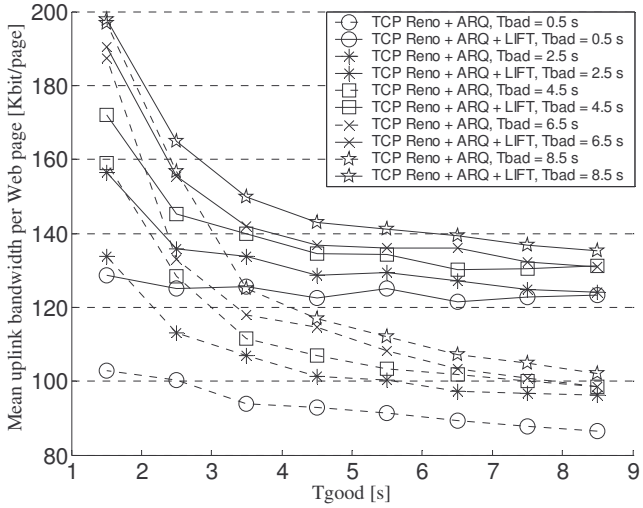


Fig. 8. Mean uplink bandwidth used for a Web page download when TCP Reno is over a reliable link layer with and without LIFT.

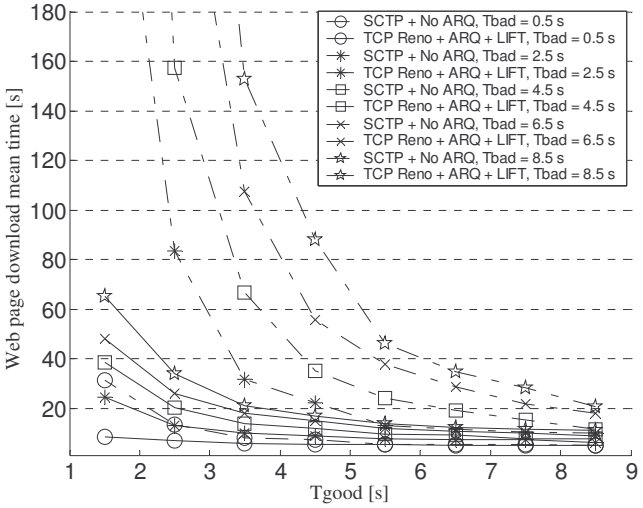


Fig. 9. Performance comparison between TCP Reno over a reliable link layer with the LIFT protocol and SCTP over an unreliable link layer in terms of the Web page download mean time.

VI. Conclusions

In this paper, a novel protocol, called Local IPSec-aware Freezing proTocol (LIFT), has been defined in order to improve TCP performance in a satellite network when it is combined with a reliable link layer over satellite link. This protocol aims at mitigating the retransmission competition problem, introduced by the use of a reliable data link protocol, by enabling the satellite gateway to freeze the TCP sender even if TCP segments are sent over end-to-end IPSec Security Associations. LIFT is local to the satellite link, requires no changes in TCP sender and receiver and maintains TCP semantics too.

Through computer simulation, the effectiveness of the LIFT has been validated considering Web traffic. Simulation results have proved that the adoption of LIFT protocol makes it possible to obtain valid further improvements of TCP performance compared with a solution that adopts only a reliable link layer over satellite channel. In addition, it makes it possible to achieve better performance than with SCTP.

Acknowledgements

The authors gratefully acknowledge the support of the ALENIA SPAZIO Spa – TLC Mission System Unit of Rome (Italy), relating to the details of the reference satellite architecture EuroSkyWay.

References

[1] H. Balakrishnan, et al.: *A comparison of Mechanisms for Improving TCP Performance over Wireless Links*, IEEE/ACM Transactions on Networking, Vol. 5, pp. 756-769, Dec. 1997.

[2] H. Elaarag: *Improving TCP Performance over Mobile Networks*, ACM Computing Survey, Vol. 34, No.3, Sept. 2002, pp. 357-374.

[3] N. Ghani, S. Dixit: *TCP/IP Enhancements for Satellite Networks*, IEEE Comm. Magazine, pp.64-72, July 1999.

[4] M. Zorzi, A. Chockalingamm, and R. R. Rao: *Throughput Analysis of TCP on Channels with Memory*, IEEE J. Select. Areas Commun., Vol.18, pp.1289-1300, July 2000.

[5] H. Balakrishnan, et al.: *Improving TCP/IP Performance over Wireless Networks*, Proc. 1st ACM Conf. on Mobile Computing and Networking, Nov. 1995.

[6] T. Goff, J. Moronski, D.S. Phatak, V. Gupta: *Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments*, Proc. of the IEEE INFOCOM '00, pp. 1537-1545, 2000.

[7] J. Brown, et al.: *M-TCP: TCP for mobile cellular networks*, ACM Computer Communications Reviw, Vol.27, 1997.

[8] M. Mathis, J. Mahdavi, S. Floyd, A. Romanow: *TCP Selective Acknowledgement Options*, Internet Society, Internet RFC 2018, Oct. 1996.

[9] M. De Blasi, G. Ciccarese, L. Casone, L. Patrono, G. Tomasicchio: *An Efficient ARQ Protocol for a Mobile Geo-stationary Satellite Channel*, IEEE Globecom 2001, pp. 2692-2697, San Antonio, Texas, 2001.

[10] M. De Blasi, G. Ciccarese, L. Patrono, A. Palmieri, G. Tomasicchio: *Improving HTTP Performance in a mobile terrestrial-satellite network*, Proc. of the 9th Ka Band and Broadband Communications Conference, Italy, Nov. 2003.

[11] R. Stewart et al.: *Stream Control Transmission Protocol*, Internet Society, RFC 2960, Oct. 2000.

[12] Shaojian Fu, M. Atiquzzaman, W. Ivancic: *SCTP over satellite networks*, Proc. IEEE 18th Annual Workshop on Computer Communications - CCW 2003, pp.112 – 116, Oct. 2003.

[13] M. De Blasi, G. Ciccarese, L. Patrono, S. Elia, G. Tomasicchio: *A Performance Enhancing Proxy for mobile satellite Internet*, Proc. of the Fifth IEEE Inter. Conf. on Mobile and Wireless Comm. Networks, Singapore, Oct. 2003.

[14] S. Kent: *IP Authentication Header*, Internet Society, RFC 2402, Nov. 1998.

[15] S. Kent: *IP Encapsulating Security Payload*, Internet Society, RFC 2406, Nov. 1998.

[16] J. Postel: *Transmission Control Protocol*, Internet Society, RFC 793, Sept. 1981.

[17] V. Paxson, M. Allman: *Computing TCP's Retransmission Timer*, Internet Society, RFC 2988, Nov. 2000.

[18] E. Lutz, D. Cygan, M. Dippold, F. Dolainsky, W. Papke: *The Land Mobile Satellite Communication – Recording, Statistics and Channel Model*, IEEE Trans. Veh. Technol., Vol. 40, pp. 375-386, May 1991.

Giovanni Ciccarese received the Laurea degree in Electronic Engineering from the Politecnico di Torino, Italy, in 1989. He is currently an Assistant Professor at the Innovation Engineering Department of Faculty of Engineering (Lecce). His research interests include design, modelling and performance evaluation of communication protocols, with particular emphasis on protocols for wireless networks. In his scientific work, he often collaborates with Alenia Spazio and STMicroelectronics.

Mario De Blasi is full professor of Computer Networks at the University of Lecce, Italy. His research and teaching activities involved computer architecture, computer networks and distance learning. He has published papers and he has written some books within these areas. He has coordinated or participated in several large national and international projects (Med-Campus Project of EU, Interactive Satellite multimedia Information System – ISIS in ACTS of EU, ESA SkyNet and ESA MODUS).
Furthermore, he coordinates industrial research projects with Alenia Spazio S.p.a. and ST Microelectronics.
The main areas of the research group he coordinates are: modelling and performance analysis, mobile Internet, IEEE 802.11e, IEEE 802.16e.
Dr. De Blasi has been a Chair of a Symposium on Future Wireless Systems organized within the Conference on Software in Telecommunications and Computer Networks (SoftCOM) since 2000.

Sebastiano Elia received the Laurea degree in Computer Engineering from the University of Lecce, Italy, in February 2003. He is currently a Ph.D. student in Innovative Materials and Technologies at the ISUFI of Lecce, Italy. His main research interests address: power saving in IEEE 802.11e and TCP performance issues over satellite and IEEE 802.16e-based networks. In his research work, he collaborates often with Alenia Spazio and STMicroelectronics.

Cosimo Palazzo received the Laurea degree in Computer Engineering from the University of Lecce, Italy, in 2003. He is currently a Ph.D. student in Computer Engineering at the University of Lecce. His research areas deal with design, modelling and performance evaluation of communication protocols over networks based on IEEE 802.11e and IEEE 802.16e and related QoS issues. For some research activities, he has recently collaborated with Alenia Spazio and STMicroelectronics.

Luigi Patrono received the Laurea degree in Computer Engineering from the University of Lecce, Italy, in 1999. He received Ph.D. in Innovative Materials and Technologies at the ISUFI of Lecce, Italy, in 2003. He is currently an Assistant Professor at the Faculty of Engineering of the University of Lecce (Italy) in the Computer Network group. In his scientific work, he collaborates often with Alenia Spazio and STMicroelectronics. His research interests include design, modelling and performance evaluation of wireless communication protocols. In particular, main research topics are: mobile Internet, satellite networks, power saving in IEEE 802.11e, and IEEE 802.16e.