NETWORK SECURITY AND PERFORMANCE EVALUATION OF ML-IPsec OVER SATELLITE NETWORKS

Michele Luglio
University of Rome “Tor Vergata”
ph: +39 06 7259 7449
email: luglio@uniroma2.it

Cesare Roseti
University of Rome “Tor Vergata”
ph: +39 06 7259 7775
email: roseti@ing.uniroma2.it

Abstract

The peculiar characteristics of the satellite links affect performance of the TCP protocol, largely used by most of the Internet applications. Then, to achieve good performance TCP Performance Enhancing Proxy mechanisms are often used. In principle, a TCP PEP mechanism accelerates TCP transfers requiring access to TCP headers in intermediate nodes. As a drawback, this conflicts with IPsec, which requires end to end semantic be respected. ML-IPsec has been identified as a suitable trade-off solution which can mitigate such a conflict. This paper addresses security issues for satellite systems highlighting the need to guarantee both security and performance. Finally, performance of different security schemes, carried out through simulations, are shown.

1. Introduction

Requirements concerning communication security originate from the utilization of shared resources to exchange information among multiple users. The level of security that a telecommunication system can ensure relies upon the degree of accessibility to unauthorized users that the network allows.

Satellite systems represent a very important component of the global information infrastructure for their intrinsic capability to efficiently and effectively support ubiquitous and broadband access to the internet, mobile services and multicast connectivity.

To counteract the impact of satellite link characteristics on TCP performance often Performance Enhancing Proxies (PEP) are utilized. The most commonly used PEP is implemented at the intermediate nodes of the network to split the end-to-end TCP connection, to intercept TCP packet and provide premature acknowledgements in order to improve performance. The further drawback of this solution is the break of end-to-end semantic, property of TCP protocols.

The growing development of telecommunication networks fostered the diffusion of Internet over a world geographic scale. As a consequence, an impressive number of users daily accesses to Internet and exploits its services (i.e. e-mail, e-commerce, e-banking, e-learning, telemedicine, etc.). In this context, intrinsic problems concerning security are playing a fundamental role in the design of a shared network. In particular, the IP protocol doesn’t offer significant protection to guarantee typical security services. To reach this goal, IPsec protocol suite was defined and standardized aiming to provide cryptographically-based security services at the IP level.

In this scenario, the implementation of IPsec end-to-end protection scheme is not compliant with the utilization of PEP, due to the to breaking of end-to-end semantics. In fact PEP needs to access and process information present in TCP/IP headers. ML IPsec has been proposed to solve this conflict.

This paper deals with the complex topic of the provision of security over satellite networks by initially presenting the intrinsic characteristics of a satellite system in terms of security (section 2) and TCP performance (section3). Section 4 introduces the problem of the security at IP layer, then section 5 illustrates the main characteristics of the IPsec protocol suite. Section 6 focuses on IPsec issues over satellite systems and reports details about its incompatibility with TCP PEP. Furthermore, section 6 presents principles of the multi-layer IP security (ML-IPsec) proposed in [1] as a solution able to guarantee both an end-to-end security scheme and good performance. Then, after the considerations on the security level provided by the ML-IPsec scheme, section 7 shows the most meaningful results of a simulation campaign performed through the network simulator ns-2 [2] and aiming to evaluate the performance at the transport level under different network security schemes. Finally, section 8 provides the conclusions.

2. Generalities on security in satellite systems

The level of security, in a broad sense, that a satellite system can ensure relies both upon its characteristics and upon the number of expedients that it is possible to implement [3]. The following peculiar characteristics of satellite systems assume a meaningful importance in providing security.

- The coverage area, typically quite extended, allows the network to serve a large number of users even in remote, isolated and impervious locations, where usually terrestrial infrastructures are not deployed (either for economic or feasibility reasons) and where typically some office that requires isolation for confidentiality can be located.
- The broadcast peculiarity of the satellite signal represents an advantage for key distribution to a large set of users spread over the territory. Moreover, this characteristic does not imply the capability to receive information for anybody. In fact, to receive just the electromagnetic signal from just one satellite a highly directional antenna is usually necessary and as a consequence accurate pointing is needed; then, even though the signal is received, to decode information may not be so easy requiring compliance with the adopted air interface. When omnidirectional antenna are adopted on the ground terminal, no pointing is necessary but the decoding information aspect still applies.
- The high latency limits performance of security protocols implementation introducing significant inefficiency if frequent key distribution updating is required, as for example in the multicast scenario [4].
- Facilities and equipment are well localized and not spread over the territory and thus extremely easy to be garrisoned. As a consequence, the unauthorized access to telecommunication resources is practically impossible.
- The practical absence of an open, diffused standard, which actually represents a weakness from market penetration point of view, represents an additional factor of security. In fact, to be able to not only receive the electromagnetic signal but to decode information, it is necessary to be authenticated by the NCC. The authentication process cannot be performed if the utilized standard is not known.
- Both the space segment and terrestrial segment equipment are extremely reliable increasing the confidence for the satellite user about security in terms of service continuity. This feature encourages the use of satellites for secure services that, in most of the cases dealing with, for example, human safety and disaster relief, need also the highest degree of survivability.

3. TCP performance over satellite links and PEP solutions

The main characteristics of an end-to-end path including a satellite segment that impact TCP performance are high latency, high bandwidth-delay product (for wideband multimedia services), exploitation of asymmetric connections, losses due to congestion and losses due to transmission errors [5]. Beyond link conditions, some default options of real operating systems (i.e., Linux, windows, BSD/OS, etc.) limit TCP performance [6][7].

Long latency. Three basic factors contribute to latency: propagation delay, processing delay and queuing delay. For satellite scenarios, the contribution of propagation delay often dominates latency. In the case of a GEO satellite, the very long distance between the satellite and the ground stations leads to a round-trip time ranging from 480 ms up to 600 ms. As a consequence, a TCP sender takes a long time to determine whether or not a packet has been successfully received at the final destination. For this reason, the growth of the TCP transmission window, based on the reception of the acknowledgement from the receiver, will proceed very slowly.

Large delay-bandwidth product. The delay-bandwidth product (DBP) defines the amount of data the sender must transmit at any instant to fully utilize the available bandwidth. Specifically, the DBP is the product of the bottleneck link bandwidth and Round Trip Time (RTT). When the delay is large, TCP needs to keep a large number of packets “in flight”, i.e., sent but not acknowledged.

Connection asymmetry. The throughput may depend on the characteristics of both links (forward and return) and asymmetry may impact TCP performance. For example, in DVB IP one-way systems the user terminal receives incoming traffic via a broadband satellite channel and sends all outgoing traffic over a narrowband terrestrial link.

Link availability and BER. Wireless links can be subject to lower link availability than wired networks due to random variations of atmosphere conditions, especially at high frequencies. In addition, errors often occur in bursts and, in case of mobile services, multipath and shadowing may reduce link availability. BER (Bit Error Rate) depends on service requirements and impacts link parameters such as transmitted power. When the link availability is low, the long term average BER decreases. Unfortunately, since there is no feedback information about the reason a packet is lost in the network (congestion or corruption), TCP interprets any packet loss as a notification of network congestion and reduces its transmission window in order to alleviate the congestion.

Operating System limitations. Optimum performance is obtained when the data pipe between sender and receiver is always kept full. On the other hand the receiver needs enough free buffer space to store all incoming data. For this purpose, TCP implements a mechanism called “Advertised Receive Window” [6]. With this mechanism, the maximum amount of data is limited by the free space in the receiver buffer and the sender is notified of this amount through the “Advertised window” field in ACK packets. In case of large pipe, the advertised window could be less than the optimal congestion window limiting the achieved throughput.

Therefore, enhancements in the standard protocol and/or in the system architecture are often necessary to guarantee the desired Quality of Service (QoS), when TCP run as transport protocol. In this frame, one of most common countermeasure in operational satellite networks is based on performance enhancement mechanisms called TCP PEP (TCP Performance Enhancing Proxy [8]). TCP PEP refers to a class of techniques that aims to improve performance through interactions between intermediate nodes and the TCP layer. Then, the implementation of a PEP in the middle of a link introduces the main implications of breaking the end-to-end semantic contrasting with one of the fundamental properties: the reliability. In fact, TCP implements a flow control mechanism that allows the sender to be sure of the correct delivery of a TCP packet to a receiver through an explicit notification (Acknowledgment). The most commonly used TCP PEP techniques (i.e., TCP spoofing, TCP splitting, etc.) are based on intermediate nodes that capture TCP traffic and generates premature acknowledgments without waiting for the delivery of the data segments to the real destination. Then, the sender is convinced that data have been successfully received, when they are still in flight. Therefore, a sudden crash of the system will cause the lost of data segment acknowledged but not received (unreliability).

3.1 PEP performance

The scope of this sub-section is to show TCP PEP benefits in terms of goodput meant as the number of payload bits per unit of time forwarded to the correct destination interface excluding retransmitted packets. Figure 1 shows the simulation scenario. In Figure 2a, the goodput is measured for both TCP PEP on and off, when a PER of 10^{-3} affects the sub-link connecting TCP sender and the SAT GW and TCP socket buffer size is 64 kbytes (typical value in setting of real operating systems). The graph clearly shows that TCP PEPs allow to recover errors in a transparent way for the receiver and to optimize performance of the transport protocol in the satellite link. On the contrary, in the end-to-end configuration errors continuously break down TCP transmission window [9] and maximum goodput is however limited to about half bottleneck capacity by the TCP socket buffer size [7]. Then, Figure 2b reports average goodput values in case of end-to-end connections for different PER values in the sub-link 1, and the corresponding improvement coming from the use of TCP PEP techniques. Such an improvement grows as the PER increases due to the substantial capacity of TCP PEP to quickly recover from errors.

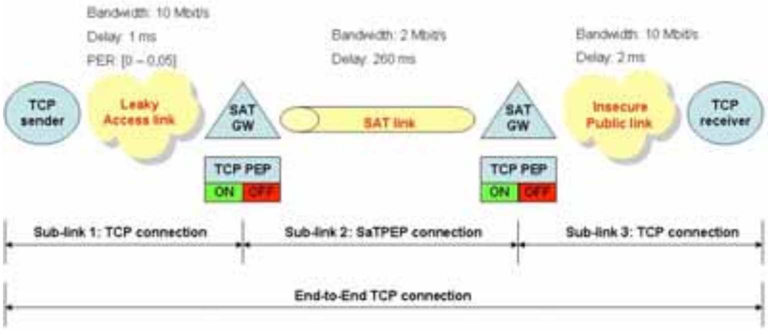


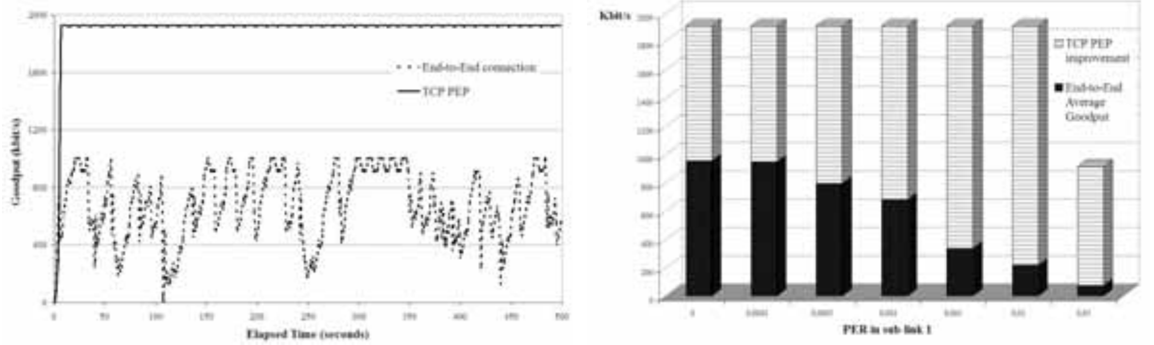
Figure 1: Simulation scenario

4. Internet security

Internet services are generally based on the client-server paradigm in which two machines exchange messages according to TCP/IP protocols [6] and the client receives the requested information from the server. In this frame, being the network resources shared by definition and thus easily accessible, the need to protect secret (or simply private) information arises. To ensure security in IP environment, a number of service requirements are identified [10].

- *Confidentiality*: protection from passive attacks, unauthorized release of message content, and traffic flow analysis.
- *Authentication*: the message sender is really who he claims to be.

- *Integrity*: the message content is not modified.
- *Non repudiation*: sender or receiver does not deny a transmitted message.
- *Access control*: limit access.
- *Key management and exchange*: negotiate security keys between communication entities.



(a) Goodput measurement: End-to-End connection vs. TCP PEP (PER in sub-link 1 = 10^{-3}, TCP window: 64 kbytes)

(b) TCP PEP goodput improvement for different PER values in sub-link 1 (TCP window:64 kbytes)

Figure 2: TCP PEP improvement

The types of attack performed by a hacker can be classified on the basis of the aspects of the service they compromise.

- *Denial of Service*: to interrupt the provision of a service in order to compromise its availability.
- *Eavesdropping*: to intercept data in order to compromise the confidentiality of the service.
- *Data manipulation*: to modify the information carried over the shared network in order to compromise the integrity and authenticity of the service.
- *Masquerade*: to insert forget information in order to compromise authenticity and non-repudiation functionalities of the service.

Therefore, the implementation of reliable secure infrastructure and protocols represents a primary need for Internet users. To reach this aim different approaches are viable.

An approach is based on the introduction of a cryptographic scheme at the session level, such as SSL or TLS [11], in order to protect data payload, while IP and transport layer (i.e. TCP) headers are left unencrypted in order to allow the routing and the management of the traffic through the intermediate nodes across the network. In this way only confidentiality of the transmitted data is guaranteed but it allows potential hackers to utilize IP spoofing that means assuming the identity of somebody else to access a protected system, for instance assuming the IP address of a machine authorized to access to the attacked system.

In 1995 IETF proposed an architecture, known as IP security (IPsec), that aims to solve security problems working at the IP layer [12]. IPsec represents the most advanced standardized solution concerning Internet security. The IPsec protocol suite offers cryptographically-based services providing confidentiality, authentication, integrity and non-repudiation. In particular, it includes an authentication protocol, *Authentication Header (AH)* [13], a confidentiality protocol, *Encapsulating Security Payload (ESP)* [14], an Internet security association establishment, *Security Association (SA)* [12], and a key management protocol (*ISAKMP*) [15]. These security protocols have been designed for both IP version 4 (IPv4) and IP version 6 (IPv6).

5. Network security: IPsec

IPsec architecture is quite complex since it does not define a single protocol, but rather a protocol suite aiming to provide interoperable cryptographically-based security services (i.e., confidentiality, authentication, integrity and non repudiation). There are three main protocols.

- The *Authentication Header* (AH) protocol: it provides connectionless integrity and data origin authentication for IP datagrams. It can also provide protection against replays [13].
- The *Encapsulating Security Payload* (ESP) protocol: it provides confidentiality for IP datagrams with optional connectionless integrity, authentication and anti-replay [14].
- The Internet *Security Association Establishment and Key Management Protocol* (ISAKMP) [15]: it is in charge to negotiate and manage a SA, that is a “contract” that defines the security mechanisms and the cryptographic keys to use. AH and ESP presume that the two end-systems have already established a *security association* (SA).

Both AH and ESP protocols have two operational modes: transport and tunnel. In transport mode, the IPsec header is inserted after the IP header in order to protect the headers of the upper layer protocols and the user data. In tunnel model, the entire IP datagram is encapsulated in a new IP header. Then, transport mode is used only for host-to-host security services, while tunnel mode can used between two hosts, a host and a gateway, two gateways. As shown in Figure 3 for the IP version 4 (IPv4), the main difference between AH and ESP is the part of an IP datagram covered by the authentication services. ESP authenticates the IP fields that it encapsulates, while AH authenticates the whole IP datagram (except for mutable fields). Furthermore, AH introduces less overhead than ESP, but if confidentiality is required AH relies upon the use of ESP or another mechanism requiring additional overhead and the execution of two security protocols.

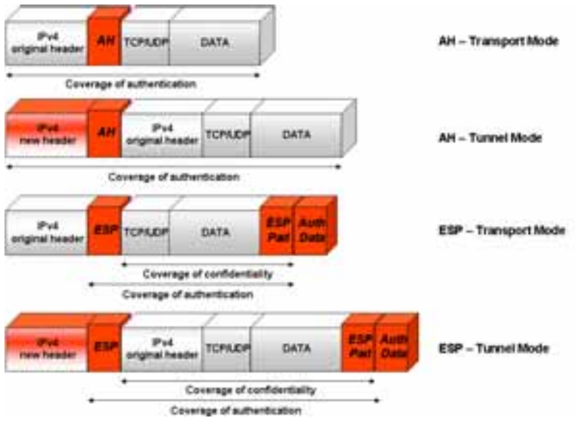


Figure 3: AH and ESP in transport and tunnel modes

6. Network security over satellite systems

Satellite systems represent a growing attractive for the provision of a broadband and ubiquitous Internet access. On the other hand their characteristics affect performance of the standard TCP protocol [5], used by most of the Internet applications. In this context, the set up of a reliable security scheme at the IP level is particularly complicated in case of typical geostationary satellite networks, which largely adopts PEP like solutions to improve performance of TCP based applications.

In fact, as mentioned, TCP PEP conflicts with IPsec. TCP PEP mechanisms, performed in intermediate node, need to process information contained into the TCP header (i.e. TCP flow identifier and sequence number), which is authenticated by the AH protocol and encrypted by the ESP protocol. In both AH and ESP cases, IPsec does not allow to perform TCP PEP mechanisms.

6.1 Conflict between IPsec and TCP PEP

When TCP PEP actions involve only the two end hosts, or involve the intermediate routers but do not require access to TCP data encapsulated in the IP packet, no particular incompatibility arises. In contrast, if splitting or some PEP [8] or network address translator (NAT) options are implemented, as in most of the commercial systems for wideband services, the end-to-end semantic is violated.

In particular, TCP PEP performing splitting are installed on the two end of a satellite link, and operate (at least) on two state information stored in the TCP headers:

- TCP flow identification;
- TCP sequence number.

The former consists of two pairs of values, IP source/destination address and source/destination ports, that are used to know TCP session for each TCP packet, while TCP sequence number allows to match acknowledgements with the corresponding TCP segment. As an example, a TCP PEP agent installed at the satellite gateways could inspect all incoming TCP segments in order to generate premature acknowledgements for TCP sender that will perceive a shorten latency. Inevitably, these techniques conflict with end-to-end security scheme defined by IPsec in both ESP and AH formats.

- *Encapsulating Security Payload (ESP)*: encrypts each IP datagram, including TCP header and thus encryption prevents TCP PEP from seeing and modifying any field of TCP headers;
- *Authentication Header (AH)*: does not encrypt IP payload, and then leaves TCP header visible. Nevertheless, the strong authentication process reject segments in which TCP PEP modifies header fields.

6.2 Possible solutions

To overcome these problems, a number of approaches have been proposed, such as replacing IPsec with a transport-layer security mechanism, tunneling one security protocol within another, using transport-friendly ESP format, and splitting IPsec into as many segments as the whole path is splinted, but each has its own limitations. In [1] all these approaches are described and their limitations are identified. Moreover, an approach based on a multi-layer security protection scheme for IPsec is proposed (ML-IPsec).

6.3 ML-IPsec

A smarter solution to perform network security over links, including TCP PEP agents installed in intermediate nodes, is based on a IPsec extension called Multi-Layer IPsec (ML-IPsec) [1]. The ML-IPsec principle is to divide IP datagram into different *zones* and apply different protection schemes at each zone. A protection scheme is then composed of: a set of security associations (SAs), a set of private keys and access control rules. Therefore, one or more authorized/trusted intermediate nodes can decrypt, modify and re-encrypt a certain part of the incoming segments according to what established in the security associations. An ML-IPsec security protection scheme well suited to satellite networks involving TCP PEP agents installed on both satellite terminal and gateway to accelerate TCP over satellite channel can be structured as follows (Figure 4).

- IP datagram payload is divided into two zones:
  - TCP header (21st to 40th octet) constitutes the zone 1;
  - TCP data portion (41st to above octet) constitutes the zone 2.
- An end-to-end protection scheme is used for TCP data portion indicated as zone 2: encryption key shares only between end-systems: source and destination.
- A separate protection scheme is used for TCP header indicated as zone 1: encryption key shared among source, destination and one or more trusted intermediate nodes (i.e., satellite gateway and satellite terminal).

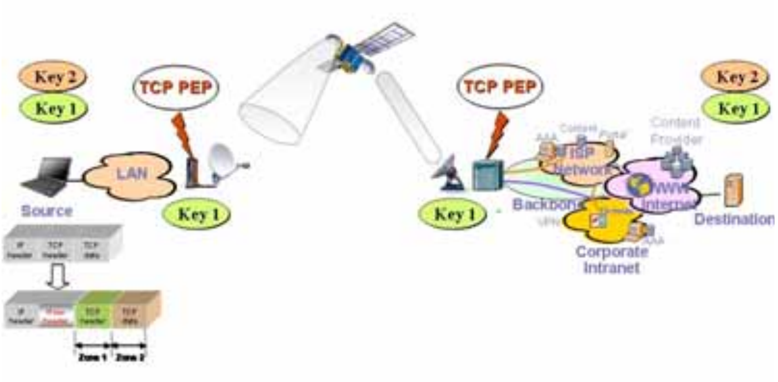


Figure 4: ML-IPsec in a TCP PEP satellite scenario

7. Performance evaluation

This section presents the results coming from a simulation activity by using the well-known Network Simulator 2 (ns-2) [2]. The simulation scenario reproduces an heterogeneous link, compliant to that one represented in Figure 4, envisaging networks with both different physical characteristics and different level of trust. In particular, the following three sub-links can be identified:

- *Sub-link 1*. It connects TCP sender to the satellite gateway and reproduces the local access network of the source system. Its physical parameters are: bandwidth = 10 Mbit/s, delay = 1 ms.
- *Sub-link 2.* It is a link connecting two satellite nodes: a satellite terminal (ST) and a satellite gateway (GW). For the sake of simplicity, hereafter we will refer to both as satellite gateway (SAT GW). Its physical parameters are: bandwidth = 2 Mbit/s, delay = 260 ms. It is the bottleneck of the overall link and PER is chosen in a range of values (from 10^{-4} to 10^{-2}).
- *Sub-link 3*. It represents the insecure public network (Internet) that connects the SAT GW to a remote TCP receiver. Its physical parameters are: bandwidth = 10 Mbit/s, delay = 1 ms.

At the SAT GWs, TCP PEP modules can be arbitrary enabled or disabled. Optionally, also on board the satellite a TCP PEP module can be enabled to further improve TCP performance as demonstrated in [16]. In particular, the Satellite TCP Performance Enhancing Proxy (SaTPEP) is considered. Mainly, SaTPEP performs connection splitting [17]. Then, in case TCP PEP is on, the end-to-end TCP connection is splinted into three (or four if TCP PEP on satellite is on) sub-connections. Furthermore, in the satellite link an enhanced TCP, adapting sender/receiver TCP buffer and the initial TCP window to the bandwidth-delay product, is used. The presented analysis evaluates end-to-end performance in terms of both goodput and file transfer time, enabling TCP PEP agents for the following security schemes:

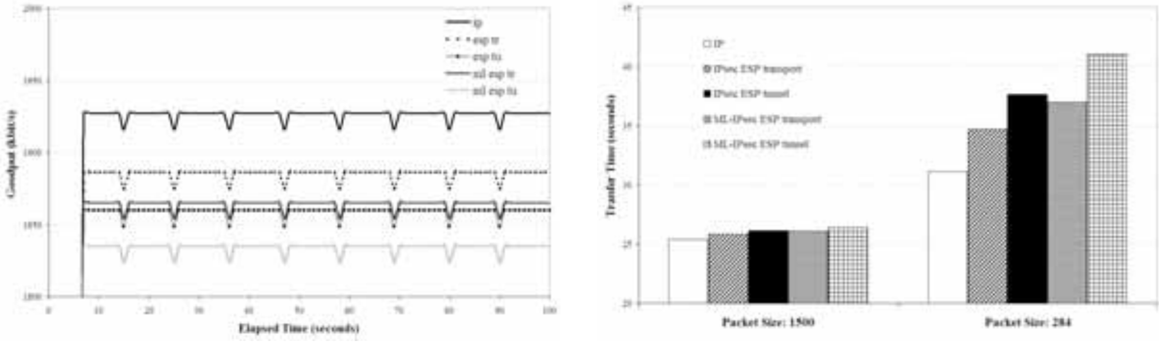
- IPsec protection provided “hop-by-hop”. Every sub-link establishes an own security association. As a drawback, this scheme requires “trusted” SAT GWs (the security level depends on competence and fairness of the network administrator).
- An end-to-end protection scheme based on ML-IPsec. SAT GWs access only TCP header fields. Data payload is protected until reaching the receiver.

Then, the question on the basis of the following analysis is: “which is the best trade-off between performance and security?”.

7.1 IPsec performance over TCP PEP

Once that the improvement deriving from the use of TCP PEPs over satellite links is demonstrated (section 3.1), the most suited security scheme must be defined. In the following analysis, confidentiality for IP datagram is assumed as a basic security requirement. Then, all the considered schemes are based on the ESP protocol. In particular, a “hop-by-hop” protection scheme, in which ESP protection is applied at each sub-link separately, is compared to ML-IPsec security architecture, where SAT GWs have the rights to decrypt, manipulate and re-encrypt TCP headers in order to perform TCP PEP operations. In both cases, transport and tunnel modes have been taken into account. ML-IPsec allows to protect end-to-end IP datagram payload, while some extra overhead may need to support the management of two cryptographic keys. On the other hand, by performing IPsec separately in each sub-link, SAT GWs must be assumed absolutely trusted. If a hacker is able to get behind the network administrator control, the communication confidentiality is completely lost. Then, considering the values of packet length overhead reported in [1], Figure 5 shows performance achieved in the different security schemes in terms of goodput and time needed to transfer a 5 Mbytes file. In this way, it is possible to associate the aforementioned considerations to performance outcomes. In Figure 5a, goodput measurements basically reflects the percentage of overhead associated to each security scheme (IP payload = 1460 bytes): ML-IPsec in tunnel model (ml esp tu) presents the higher overhead by leading to a goodput of 1834 kbit/s, while the higher goodput occurs when no protection is performed (ip). In general, tunnel mode introduces more overhead then transport mode as well as ML-IPsec requires more overhead than IPsec. Figure 5b focuses on measuring time needed to transfer 5 Mbytes of data produced by an application (i.e. FTP) by considering two different packet sizes: 1500 and 284 bytes. If the transfer time is not particular affected in case of large packets (low overhead percentage), it results significantly different from a security scheme to another in case of short packets (high overhead percentage).

An important aspect analysed in the simulation campaign has been the measurement of the transfer time increase, with respect the case in which protection is not applied (IP), for growing values of PER in the satellite segment. In particular, Figure 6 is concerned to a transfer of a 10 Mbytes file by FTP protocol when TCP modules are enabled in the SAT GWs and on satellite and PER over satellite link ranges from 10^{-4} to 10^{-2}. Outcomes show as the growth of the transfer time increase is not linear and, in addition, the performance gap between different protection schemes is not constant, varying PER value. Definitively, the influence of the overhead on end-to-end performance is more relevant for higher PER values.



(a) Goodput measurements under different IPsec schemes (Packet size: 1500 bytes)

(b) Transfer Time of a 5 Mbytes file under different IPsec schemes

Figure 5: IPsec & ML-IPsec performance over a TCP PEP-based satellite link

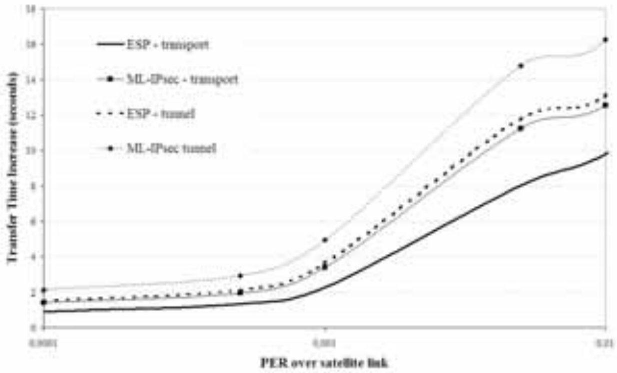


Figure 6: Transfer Time Increase vs. PER over satellite link

5. Conclusions

The design of IP security architecture is fundamental for the diffusion of new classes of Internet services requiring confidentially, authentication, data integrity, non-repudiation. In this frame, IPsec represents the strongest effort to standardize Internet security. On the other hand, satellite networks need to use TCP PEP techniques in order to achieve optimal performance, which conflicts with IPsec. This paper presents two approaches to provide IP security in a satellite network envisaging TCP PEPs: "hop-by-hop" IPsec protection, and the Multi Layer IPsec protection, Both the approaches present pro and cons in terms of security and of correlated performance. Then, performance evaluation has been carried out in order to help in identifying the best trade off solution.

References

[1] Y. Zhang, *A multilayer IP security protocol for TCP performance enhancement in wireless networks.* IEEE Journal on Selected Areas in Communications, Vol. 22, n. 4, pp. 767-776, May 2004.

[2] NS-2 Network Simulator (Vers. 2.27),URL: http://www.isi.edu/nsnam/ns/nsbuild.html

[3] M. Luglio, A. Saitto, "Security of Satellite Networks", chapter in H. Bidgoli (Ed), "The Handbook of Information Security", John Wiley & Sons, Inc., 2006, Hoboken, N.J., Vol. 1, pp. 754-771.

[4] M. P. Howarth, S. Iyengar, Z. Sun and H. Cruickshank, "Dynamics of key management in secure satellite multicast", IEEE Journal on Selected Areas in Communications, Vol. 22, n. 2, pp. 308-318.

[5] C. Partridge, and T. Shepard, *TCP Performance over Satellite Links*. IEEE Network, vol. 11, n. 5, 1997, pp. 44-49.

[6] W. Stevens, *TCP/IP illustrated, Volume 1*. Addison Wesley, 1994.

[7] M. Luglio, C. Roseti, and M. Gerla, *The Impact of Efficient Flow Control and OS Features on TCP Performance over Satellite Links*. ASSI Satellite Communication Letter (Sat-Comm Letter), 9th edition, special issue on Multimedia Satellite Communication, vol. III, n. 1, 2004, pp. 1-9.

[8] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby, *Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations*, Internet RFC 3135, Jun. 2001.

[9] W. Stevens, *TCP Slow Start, Congestion Avoidance, Fast retransmit, and Fast recovery Algorithms*, Internet RFC 2001, 1997.

[10] G. Noubir, L. Von Allmen, "Security issues in Internet protocols over satellite links", in proceedings of the 50th Vehicular Technology Conference, pp. 2726-2730.

[11] E. Rescorla, *SSL and TLS: Designing and Building Secure Systems.* Addison Wesley, 2001.

[12] R. Atkinson, *Security architecture for the Internet Protocol.* RFC 2401, Nov. 1998.

[13] R. Atkinson, *IP Authentication Header.* RFC 2402, Nov. 1998.

[14] R. Atkinson, *IP Encapsulating Security Payload (ESP).* RFC 2406, Nov. 1998.

[15] D. Maughan, M. Schertler, M. Shneider, J. Turner, *Interactive Security Association and Key Management Protocol (ISAKMP).* RFC 2408, Nov. 1998.

[16] J. Stepanek, A. Razdan, A. Nandan, M. Gerla, and M. Luglio, "The Use of a Proxy on Board the Satellite to Improve TCP Performance", IEEE Global Telecommunications Conference, GLOBECOM '02, Vol. 3, 2002, pp. 2950–2954.

[17] D. Velenis, D. Kalogeras, and B. Maglaris, *SaTPEP: a TCP Performance Enhancing Proxy for Satellite Links*. 2nd International IFIPTC6 Networking Conference, May 2002.