Performance Enhanced Proxy Solutions for Satellite Networks: State of the Art, Protocol Stack and Possible Interfaces

Igor Bisio, Mario Marchese, and Maurizio Mongelli

Abstract. There are many types of *Performance Enhancing Proxies* (PEPs). Different types of PEPs are used in different environments to overcome different link characteristics which affect protocol performance [1]. The main examples concerns satellite and wireless networks, which represent the most challenging environments.

In the following, the main concepts representing the state of the art of the PEP will be introduced. An extension to the ETSI *Broadband Satellite Multimedia* (BSM) is proposed, in order to include PEP interfaces.

1 Introduction

There are many types of Performance Enhancing Proxies (PEPs). Different types of PEPs are used in different environments to overcome different link characteristics which affect protocol performance [1]. The main examples concerns satellite and wireless networks, which represent the most challenging environments. More specifically, PEPs are network agents designed to improve the end-to-end performance of some communications protocol such as Transmission Control Protocol. PEPs function by breaking the end-to-end connection into multiple connections and using different parameters to transfer data across the different legs. This allows the end systems to run unmodified and can overcome some problems with TCP window sizes on the end systems being set too low for satellite communications. A typical system uses transport layer PEPs to improve TCP performance over a satellite link. The end systems use standard TCP with no modifications, and do not need to know of the existence of the PEPs in between. The transport layer PEPs intercept the TCP connections from the end systems and terminate them. The PEPs then use some other protocol to transfer data between them before translating back to TCP to send the data to the destination.

2 Types of PEPs

a) Transport Layer PEPs

Transport layer PEPs operate at the transport level. They may be aware of the type of application being carried by the transport layer but, at most, only use this information to influence their behavior with respect to the transport protocol; they do not modify the application protocol in any way, but let the application protocol operate end-to-end.

K. Sithamparanathan (Ed.): Psats 2009, LNICST 15, pp. 61–67, 2009.

b) Application Layer PEPs

Some application protocols employ extraneous round trips, overly verbose headers and/or inefficient header encoding which may have a significant impact on performance, in particular, with long delay and slow satellite links. This unnecessary overhead can be reduced, in general or for a particular type of link, by using an application layer PEP in an intermediate node.

3 PEPs' Implementations

Another important characteristic of PEPs concerns the Distribution. A PEP implementation may be integrated (i.e., it comprises a single PEP component implemented within a single node), or distributed (i.e., it comprises two or more PEP components, typically implemented in multiple nodes). An integrated PEP implementation represents a single point at which performance enhancement is applied. A distributed PEP implementation is generally used to surround a particular link for which performance enhancement is desired. For example, a PEP implementation for a satellite connection may be distributed between two PEPs located at each end of the satellite link. A typical example of a distributed PEP is the Satlabs I-PEP [2] and an example of integrated PEP is the PEPsal solution [3].

a) Split Connections Concept

A very important role in PEP architecture is played by the split connection TCP implementation. It terminates the TCP connection received from an end system and establishes a corresponding TCP connection to the other end system. In a distributed PEP implementation, this is typically done to allow the use of a third connection between two PEPs optimized for the link (for example the I-PEP protocol that is recommended by Satlabs).

4 Overview of PEP Mechanisms

An obvious key characteristic of a PEP implementation is the mechanism(s) it uses to improve performance. Some examples of PEP mechanisms are described in the following subsections. A PEP implementation might implement more than one of these mechanisms:

- TCP ACK Handling: many TCP PEP implementations are based on TCP ACK manipulation. The handling of TCP acknowledgments can differ significantly between different TCP PEP implementations.
- TCP ACK Spacing: in environments where ACKs tend to bunch together, ACK spacing is used to smooth out the flow of TCP acknowledgments traversing a link.
- Local TCP Acknowledgements: in some PEP implementations, TCP data segments received by the PEP are locally acknowledged by the PEP.

- Local TCP Retransmissions: a TCP PEP may locally retransmit data segments lost on the path between the TCP PEP and the receiving end system, thus aiming at faster recovery from lost data.
- TCP ACK Filtering and Reconstruction: on paths with highly asymmetric bandwidth the TCP ACKs flowing in the low-speed direction may get congested if the asymmetry ratio is high enough.

Other important functionalities may play important role in the overall PEP action. In more detail, they are:

a) Tunnelling

A Performance Enhancing Proxy may encapsulate messages in a tunnel to carry the messages across a particular link or to force messages to traverse a particular path. A PEP at the other end of the encapsulation tunnel removes the tunnel wrappers before final delivery to the receiving end system.

b) Compression

Many PEP implementations include support for one or more forms of compression. In some PEP implementations, compression may even be the only mechanism used for performance improvement.

c) Handling Periods of Link Disconnection with TCP

During link disconnection or link outage periods, a TCP sender does not receive the expected acknowledgments. Upon expiration of the retransmit timer, this causes TCP to close its congestion window with all of the related drawbacks.

d) Priority-based Multiplexing

Implementing priority-based multiplexing of data over a slow and expensive link may significantly improve the performance and usability of the link for selected applications or connections. A user behind a slow link would experience the link more feasible to use in case of simultaneous data transfers, if urgent data transfers (e.g., interactive connections) could have shorter response time (better performance) than less urgent background transfers. If the interactive connections transmit enough data to keep the slow link fully utilized, it might be necessary to fully suspend the background transfers for awhile to ensure timely delivery for the interactive connections.

5 PEP Protocol Stack

In Fig. 1, the PEP node functional architecture has been reported [4]. The node owns a specific application layer employed to manage the needed functions of the PEP. The functionalities of each individuated layer have been listed in the following by considering the PEP node acting together with the Satellite Terminal (ST) node. In practice, as previously said, the ST nodes operates as QoS Gateway (GW) and the PEP node implements all the functions aimed at optimizing the end-to-end data transfers based on the TCP.

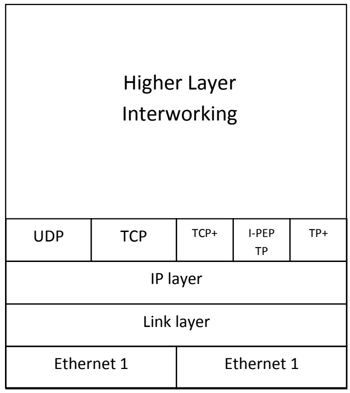

Fig. 1. General PEP protocol architecture

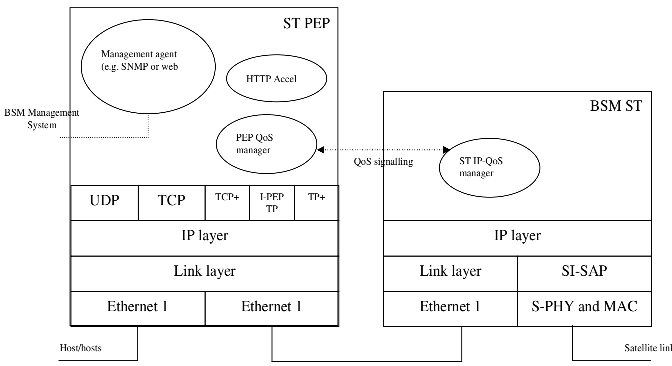

Fig. 2. BSM ST PEP

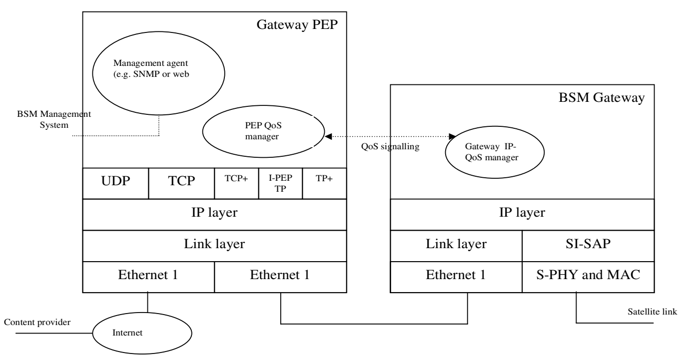

Fig. 3. BSM Gateway PEP

Figs. 2 and 3 show the ST and GW combined with PEP protocol architectures respectively. On the satellite network sides, the ST/GW PEP are connected to BSM ST/GW through an Ethernet LAN. However, the GW PEP can be located remotely from the BSM GW terminal (such as GW PEP run by a service provider). The transport protocol in the PEP is divided between standard TCP/UDP and PEP specific transport protocols. The PEP specific transport protocol can be as follows. A modified TCP (TCP+), such as TCP Hybla, which is used in integrated PEP configurations, where only GW PEP will be used (no ST PEP). Standards I-PEP transport protocol (I-PEP TP [2]) is recommended for distributed PEP configurations.

6 PEP and ST/BSM Nodes' Interfaces

The ETSI-TS102292 and ETSI-TR101984 standards define the internal and external interfaces of the BSM architecture. In this section, their approach is generalized to the presence of PEP devices, coherently with the protocol architecture envisaged in last section. The interfaces including PEP are depicted in Fig. 4. The reference topology is the star one, where ST sends/recieves packets to/from the GW of the network. The case of mesh topology is in coherence with the proposed solution, too.

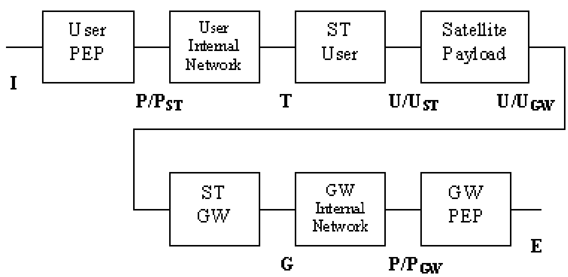

Fig. 4. Reference model of BSM with PEP access

The corresponding description is reported in following table. The Internal network considered regards any possible connection used to link the PEP node to the ST/BSM node.

Fig. 5 deals with the reference model of the BSM architecture together with PEP node interfaces. The following table details all the named interfaces of Fig. 5. Interfaces between I.10 and I.16 (right side of Fig. 5) are specular to interfaces I.1-I.7, the only difference relies on referring to GW PEP device in place of ST PEP device (left side of Fig. 5).

Ref.	Physical Interface Name	Description of Interface
I	User PEP/ External Network Interface	Interface between user PEP and premises network
P/P_{ST}	PEP/User Internal Network inteface	Interface between PEP node on user side and internal network
T	ST/Internal Network Inteface	Interface between User ST and Internal Interface
U/U_{ST}	ST/Satellite Network Interface	Satellite Radio Interface for User ST
U/U_{GW}	ST/Satellite Network Interface	Satellite Radio Interface for Gateway
G	ST/ Internal Network Interface	Interface between Gateway ST and internal network
P/P_{GW}	PEP/ Internal Network Interface	Interface between Gateway PEP and internal network
E	PEP/ External Network Interface	Interface between Gateway PEP and external network

Ref.	Interface Name	Description of Interface
I.1	External network interface	Interface between end system and customer premises network
I.2	PEP ST interface	External interface between premises network and PEP adaptation function
I.3	PEP ST Technology Independent interface	Internal interface between PEP adaptation function and Technology Independent layer
I.4	PEP ST Technology Dependent interface	Internal interface between Technology Independent layer and Technology Dependent layer
I.5	ST interface	External interface between PEP Technology Dependent layer and satellite access function
I.6	BSM subnetwork service access point	Internal interface
I.7	BSM satellite independent service access point SI-SAP	Internal interface

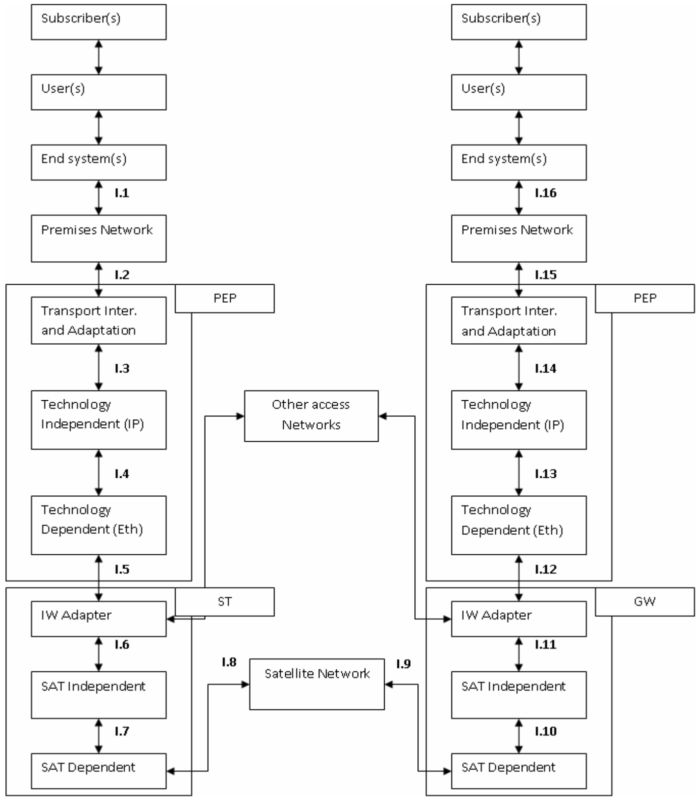

Fig. 5. PEP node Interfaces

7 Conclusions

The paper has presented most important aspectsconcerning the Performance Enhancing Proxies (PEPs), suited to be used over satellite channel. These indication may represent a useful set of suggestions to design, realize and test a real PEP system in future Quality of Service oriented satellite networks. Specific attention has been devoted to include PEP interface in the ETSI Broadband Satellite Multimedia (BSM) standards.

References

[1] IETF RFC 3135: Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations
[2] I-PEP specifications, Issue 1a. Satlabs group recommendations (October 2005), http://www.satlabs.org
[3] Caini, C., Firrincieli, R., Lacamera, D.: PEPsal: a Performance Enhancing Proxy designed for TCP satellite connections. In: IEEE 63rd Vehicular Technology Conference, 2006. VTC 2006-Spring, May 2006, vol. 6, pp. 2607–2611 (2006)
[4] Cruickshank, H., Mort, R., Berioli, M.: PEP Architecture for Broadband Satellite Multimedia (BSM) Networks. In: Workshop on Satellite PEPs Current Status and Future Directions, Nordwijk, The Netherlands (December 2, 2008)