Satellite Communications: from PEPs to DTN

C. Caini, R. Firrincieli
DEIS
University of Bologna
Bologna, Italy
{ccaini, rfirrincieli}@arces.unibo.it

H. Cruickshank
Centre for Communication Systems Research
University of Surrey
Guildford, United Kingdom
h.cruickshank@surrey.ac.uk

M. Marchese
Department of Communication, Computer and System Sciences
University of Genoa
Genoa, Italy
mario.marchese@unige.it

Abstract **Delay-/Disruption-Tolerant Networking represents an innovative way to cope with satellite communications impairments. In this view, the paper presents an in-depth analysis of implications of a DTN approach to satellite communications, focusing on these fundamental aspects: network architecture, security, and Quality of Service (QoS). For each topic, commonalities and differences between DTN and Performance Enhancing Proxies (PEPs) are highlighted, to show that the DTN architecture can be seen as a promising evolution of PEPs, at present the most widely adopted architecture. The analysis shows that DTN can effectively improve PEPs functionalities in the presence of intermittent and disruptive channels and/or a high level of network heterogeneity. In particular, DTN offers the possibility to operate with intermittent channels, a better resilience to channel disruptions, the possibility to implement both end-to-end and hop-by-hop security, and a greater flexibility in the use of advanced QoS techniques.**

Index Terms— **DTN, PEPs, Security, QoS, Satellite Communications.**

I. INTRODUCTION

Satellite communications present some distinctive features which deserve to be briefly analyzed. On the positive side, they offer a very effective way to offer a fast coverage of large areas. Through satellites, ubiquitous Internet access can be offered at reasonable costs in developing countries and in scarcely populated areas, thus helping in reducing the digital divide. Moreover, satellite communications are essential to support rescue teams in case of natural calamities, like earthquakes and flooding, when the terrestrial communication infrastructure is usually seriously damaged. On the other hand, satellite systems, and in particular GEO, have to cope with a series of peculiar challenges at different levels of the protocol stack. In particular, if we focus on Transport and upper layers, performance is challenged by the following impairments [1]: long Round Trip Times (RTTs), especially for GEO systems (about 600 ms); possible presence of segment losses due to residual errors on the satellite link; possible channel disruptions, especially for mobile terminals, due to satellite link obstructions (buildings, tunnels, etc.).

To take full advantage of satellites it is necessary to cope with the impairments mentioned above. Although an end-to-end approach, i.e. the use of an optimized transport protocol (or an optimized version of TCP) on both end nodes (client and server) is theoretically possible, it is not practical for general servers. In fact, as satellite clients are a small user niche for general content providers, they have no real advantage in introducing a modification of the customary protocol stack just to offer a better Quality of Service (QoS) to the satellite user segment. To retain the possibility to adopt transport protocol variants optimized to the satellite link, the usual solution is given by Performance Enhancing Proxies (PEPs), or protocol accelerators, based on the TCP splitting technique [2], [3]. They are intermediate nodes, inserted either at one end (integrated PEP), or more frequently both ends (distributed PEPs), of the satellite link, to isolate the satellite link (and its impairments) from the rest of the network. In short, they split the original end-to-end connection in two (integrated) or three (distributed) separate connections, thus allowing the use of optimized protocol on the satellite segment. PEPs are an effective solution and have the important advantage of being transparent to end user. By contrast, they violate the end-to-end semantics of transport protocols, and have other serious disadvantages from the point of view of security, as TCP splitting is incompatible with IPSec. A different approach, which somewhat retains and actually extends the concept of TCP splitting, is that based on the DTN architecture [4], [5], [6]. DTN is particularly suited to cope with intermittent connectivity provided by single LEO satellite (e.g. for data sensing) or incomplete constellations (e.g. for vehicle and good tracking) [7]. However, it can represent a valid alternative to PEP also in GEO systems, as shown in [8] and [9] for continuous and disruptive channels, respectively.

The present paper aims to focus the reader attention on the most relevant features of DTN, when applied to satellite communication in general, and GEO in particular. To this end, the core of the paper, which follows this introduction, consists of three sections, each of which devoted to the analysis of a different macro-aspect: architecture, security and QoS. The analysis is comparative, DTN vs. PEPs, to highlight the novelty aspects of the DTN approach. The aim is twofold: first, to make aware the reader who is expert on satellite communications of the new opportunities offered by DTN; second, to convince the reader who is expert on DTN, but perhaps less familiar with the peculiar characteristics of satellite communications, that these represent an important and promising application field.

II. DTN ARCHITECTURE

The most common DTN architecture is that based on the introduction of the Bundle layer between Transport

and Application layers. The corresponding "Bundle protocol" can interface with different transport protocols through "convergence layer adapters", as shown in the figure below [5], [6].

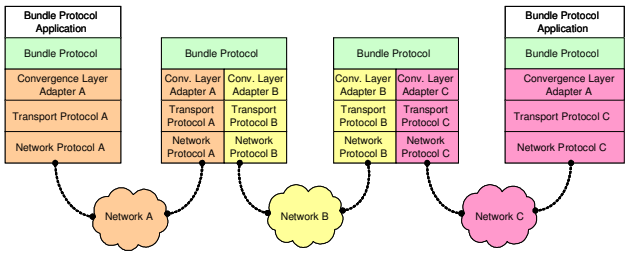

Figure 1. DTN architecture and protocol stack.

The new architecture has many novelty aspects with respect of TCP/IP based networks [4], [10]. The most prominent, when dealing with satellite communications, are summarized below by comparing, whenever possible, the new DTN architecture with satellite PEPs.

A. *DTN as an overlay*

First, although TCP/IP protocols are not replaced (at least not necessarily), their role is changed. In particular, the new DTN architecture is particularly useful when it acts as an overlay on top of a heterogeneous network consisting of many homogeneous segments, such as wireless sensor/ad-hoc networks, wired Internet, wireless LANs, satellite links, etc. By installing the DTN Bundle protocol on end-points and on nodes at the border of homogeneous segments, the end-to-end path is divided in many DTN hops. On each DTN hop a different protocol stack can be used, or, when the same stack is retained, which is the most common case, just different protocols, like TCP or UDP, or different versions of the same protocol (e.g. different variant of TCP). In this architecture, the end-to-end transport protocol features are redefined, being confined inside each DTN hop. In fact, real end-to-end data transfer from DTN sender to DTN receiver is now provided by the Bundle protocol, which exchanges large data packets, called "bundles", between DTN nodes through a store-and-forward relay mechanism.

Readers familiar with satellite communications can easily realize that the DTN multi hop architecture can be seen as a generalization of the TCP splitting concept widely used in satellite PEPs. In particular, both allow the use of optimized protocols (or optimized versions of the same protocol) on the satellite segment. However, in DTN the "splitting" is a direct consequence of the new architecture, while in PEPs it implies a severe violation of the end-to-end TCP semantics. In fact, intermediate PEPs must operate at Transport and Application layers, while the protocol stack reserves these functionalities to end nodes only. It is clearly unsafe that intermediate nodes disguise themselves as end nodes, by forging fake ACKs. More practically, this prevents the use of IPSec (see the next section). On the other hand, by contrast to PEPs, the DTN architecture is not transparent to end nodes.

B. *Information storage at intermediate nodes*

The second, but not less important, difference between DTN and customary TCP/IP network is related to information storage. In standard networks, because of usual assumptions of continuous connectivity and short delays, information is stored only at end nodes, i.e. outside of the network core. This because, dealing with reliable transmission, information is supposed to be easily retrieved directly from the source. Of course, this is not the case in challenged networks, where the usual assumptions do not hold anymore. Therefore, to deal with long RTTs and channel disruptions, and to cope with the extreme case of possible absence of end-to-end connectivity, in DTN networks information is stored at intermediate DTN nodes.

This feature actually differentiates DTN architecture from usual PEPs. In PEPs too, some segments can be stored, but this storage is temporary and just finalized at synchronizing the incoming with the outgoing segment flows. Note that the segment rate of the incoming flow can be easily controlled by PEPs by increasing or decreasing the advertised window. Summarizing, in PEPs only few segments are stored; they are stored in volatile memory and in case of long link interruptions or PEP failures, they get lost. By contrast, DTN bundles, which usually are much larger than segments, can be stored at intermediate nodes for long period of times, and, when the custody option (see the next subsection for details) is enabled, saved on non volatile memory (e.g. on local hard disk). This makes DTN much more robust against disruptions, disconnections, and temporary node failures. On the other hand, memorization in local databases requires raise congestion control issues that still need to be addressed.

C. *Custody transfer option*

By enabling the custody transfer option [11], intermediate DTN nodes are asked to accept bundle custody, i.e. to accept responsibility for bundle reliable delivery to the final destination. If the request is accepted, bundles are written in local databases where they are safely kept until, after successful transmission, custody is transferred to another "custodian", or the bundle is delivered to the final destination, or the bundle lifetime expires. This feature offers a significant reliability improvement in the presence of discontinuous or disruptive channels. To see why, let us recall that while the Maximum Tolerable Disruption Length (MTDL) of a TCP connection is about 20 minutes (Linux defaults), the Bundle protocol MTDL is longer than 24 hours (DTN2 reference implementation defaults) [12]. Although this is independent of the actual use of custody transfer, this option makes bundle storage safe even against intermediate node failures. Moreover, the acceptance of custody by intermediate nodes, allows the sender to delete data accepted in custody. This can be useful whenever the sender has limited memory resources (or good reasons not to keep in its memory sensitive information, like in military applications).

In summary, DTN architecture is much more resilient to long disruptions than usual TCP connections and PEPs. It must, however, be emphasized that the actual resilience of TCP to long disruption is highly configurable. So in principle, in a TCP splitting PEP the TCP connection on the satellite segment can be effectively tuned to this end. This in turn requires, in commercial services with large number of users, to keep open a huge number of TCP connections on intermediate PEPs to cope with (possible) disruptions, which is highly inefficient (large buffer memory is required) and not desirable from the point of view of service providers. By contrast, not only does DTN architecture offer better resilience against long disruptions, but, thanks to custody transfer, also resilience against possible node failures and a better use of end-nodes memory resources.

D. Proactive and reactive bundle fragmentation

An interesting feature of DTN Bundle protocol is the possibility of fragmenting bundles. RFCs [5] and [6] distinguish between proactive and reactive fragmentation. The former has been conceived to cope with intermittent periodic connectivity, where there may be a stringent constraint on the maximum amount of data that can be transferred (contact volume) on a DTN hop at each availability time window (contact time). Whenever the contact volume is known a priori, as in LEO and in deep space communications, proactive fragmentation allows large bundles to be divided "a priori" into multiple fragments compatible with the contact volume. By contrast, reactive fragmentation works a posteriori, when long channel disruptions interrupt a bundle transfer. In order not to retransmit successfully received data, the bundle only partially transmitted is split into two "fragments". The first contains data already sent, the second the other ones. At link re-establishment, only the second fragment is transmitted. Bundle fragments are treated as ordinary bundles, and consecutive fragmentations are possible. The reactive fragmentation is particularly useful when disruptions may be relatively frequent, as in satellite communications with mobile terminals, when obstacles (buildings, tunnels, etc.) may prevent satellite signal reception.

Both proactive and reactive fragmentations are distinctive features of DTN.

III. DTN SECURITY ARCHITECTURE

Due to the DTN characteristics described in section II, the security architecture requires some distinctive features that will be detailed below in sub-section A. By contrast, PEPs do not have any specific security architecture and they borrow the traditional security mechanisms from the Internet such as IPSec and Transport Layer Security (TLS), as detailed in sub-section B.

A. DTN Security state of art

Current Internet security protocols (such as IPSec and TLS) do not perform well in high delay/disruption conditions, because of underlying assumption on which they are built, such as: end-to-end connectivity always present; low link delays between communicating parties and low error rate on link channels. Thus, new security architecture is needed to meet DTN requirements [13], [14] and [15].

Let us focus the attention on Figure 2, which shows two DTN Bundle Nodes BN1 and BN4 from two different networks connected to each other through DTN gateways BN2 and BN3. Any DTN node originating or forwarding a bundle, stores it in its memory until it has been delivered to the next node, showing a "Store and Forward" style of communication as explained in section II.C.

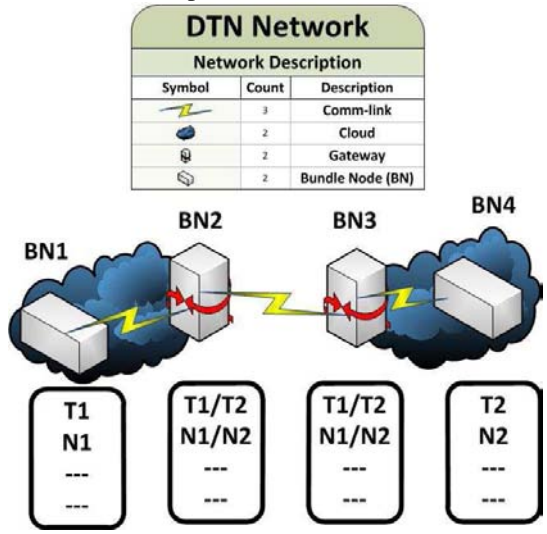

Figure 2. Internetworking of DTN networks using bundle gateways

The security architecture supports hop-by-hop and end-to-end authentication and integrity validation, to ensure data is correct before forwarding. Figure 3 shows the hop-by-hop authentication/integrity check using Bundle Authentication Block (BAB). The BAB is used to assure the authenticity and integrity of the bundle along a single hop from forwarder to intermediate receiver. Thus, the communication path is divided into security zones (as shown in Figure 3). Similarly, the Payload Integrity Block (PIB) and Payload Confidentiality Block (PCB) are used for end-to-end security services. Further details on security architecture in DTN can be found in [14].

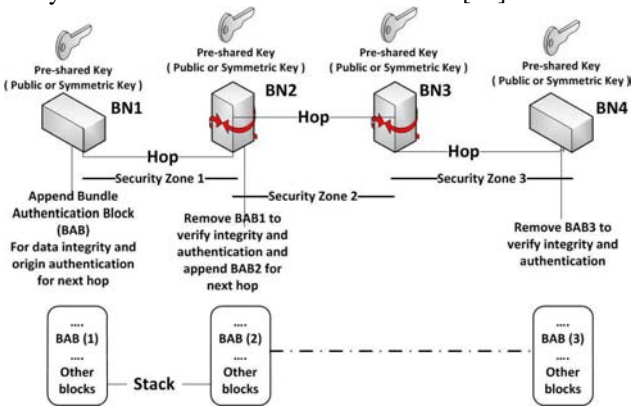

Figure 3. Hop-by-hop authentication and integrity check (from [14]).

Key Management is one of the most difficult problems in DTN security. DTN security requires that before forwarding the bundle it must be authenticated and integrity checked. In DTN, link availability is an

important resource and special techniques need to be applied to maximise the utilization of such link and minimise the overheads of key management.

B. *Security impact on Performance Enhancing Proxies*

There are limitations for implementing end-to-end security in the presence of PEPs and other intermediate network entities [3]. Examples of such limitations are presented below.

Conflicts between IPSec and TCP PEP. TCP PEP operates on information stored in the header of a TCP packet such as TCP flow identification and sequence and ACK numbers. When a TCP session is performed on top of the IPSec ESP (Encapsulating Security Payload) protocol, the TCP header is encrypted inside the ESP header. It is, thus, impossible for an intermediate gateway (like TCP PEP) outside sender or receiver's security enclaves to analyze an IPSec header to extract TCP flow information.

Application Layer Proxies/Agents. Web proxies need to parse the TCP and HTTP header of a passing IP datagram and serve it with the web page from local cache. It is transparent to end-users but boosts the responsiveness of satellite and wireless networks. Again, end-to-end IPSec will prevent the operation of these web proxies.

Traffic Engineering. Flow classification is essential in providing classes of services and QoS support. These include Random Early Detection (RED) and router-based congestion control and policing, integrated services (intserv) with Resource Reservation Protocol, (RSVP), and differentiated services (diffserv) and Multi Protocol Label Switching (MPLS). Again, this may potentially conflict with IPSec (especially in IPSec in tunnel mode).

To overcome these limitations security must be implemented in such a way that allows Satellite Terminal (ST) and Gateway PEPs to access the transport protocol headers for Transport PEPs (T-PEPs) and HTTP content for Application PEPs (A-PEPs). Transport/application layer security (such as Transport Layer Security, TLS and secure HTTP) will work seamlessly with T-PEPs because the TCP header is not encrypted by the security system (see in Figure 4). However, transport/application layer security will not function with A-PEPs. The reason is that application layer data will be encrypted by the security system. Hence, it will not be possible to perform techniques such as HTTP pre-fetching, caching and header and payload compressions.

End-to-end network layer security (such as IPSec) will encrypt the TCP header and user data; therefore, both T-PEPs and A-PEPs will not work. As such, T-PEPs will not be able to perform techniques such as TCP spoofing, ACK reduction and flow control. In addition, A-PEPs will not be able to perform HTTP prefetching, caching and compression. Thus a user or network administrator must choose between PEPs and end-to-end IPSec.

As shown in Figure 4, PEPs can be used successfully with IPSec in tunnel mode between the satellite ST and Gateway. Here the encryption is performed on incoming traffic after the PEP operations and decryption is performed on outgoing traffic before the PEP operations. The IPSec operations are under the control of the satellite network operator. In terms of overhead, IPSec tunnel mode requires an extra IP header, where basic IPv4 header is 20 bytes and IPv6 header is 40 bytes. Figure 4 shows also the link layer security mechanism that can be used (e.g. DVB-RCS [16] security or Unidirectional Link Encapsulation (ULE) security [17]). Here T-PEPs and A-PEPs will work seamlessly over the secure satellite link. The reason is TCP header and user data are handled in clear text (no encryption) both in the Gateway PEP and in the ST PEP. Then, the satellite link layer security is only applied between the BSM ST and GW (satellite terminals).

Although link layer security does not provide the desired end-to-end security, it is more efficient than using IPSec (in tunnel mode). It also can provide extra security functions that are not possible with IPSec or upper layer security such user identity hiding (such as IP and terminal MAC addresses). This allows providing strong privacy service over the satellite broadcasting link.

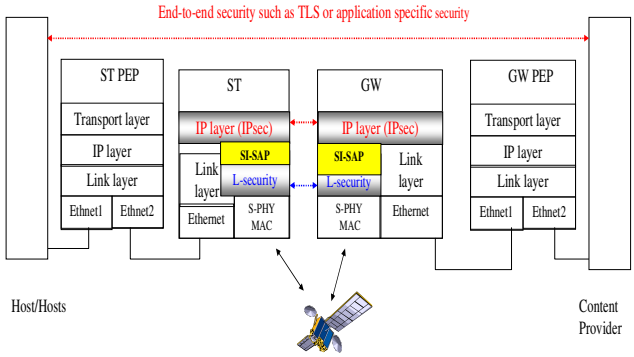

Figure 4. Security solutions with PEPs.

C. *Comparison between DTN and PEP security*

Examining security issues in DTN and PEPs highlights that intermediate entities require access to some parts of the end-to-end packets exchanged between the source and destination. The presence of PEPs limits the security solution to link layer security such as the DVB-RCS or ULE security. On the other hand, DTN and the Bundle protocol are application layer functions, which means that both hop-by-hop and end-to-end security can be provided by BAB and PCB (as shown in Figure 2 and Figure 3). Moreover, also within a single DTN subnet, it is possible to use link layer security such as the DVB-RCS and ULE security.

In summary, by comparing DTN and PEPs, it results that DTN bundle security is a better and more flexible solution when dealing with network heterogeneity and the presence of intermediate entities. Moreover, the hop-by-hop authentication allows DTN intermediate nodes to reject unwanted traffic and preserve the limited DTN resources.

IV. DTN AND QUALITY OF SERVICE

A. *The importance of QoS*

As mentioned earlier an essential aspect of modern

telecommunication networks, which include DTN, is QoS. The importance of QoS increases in parallel with the evolution of telecommunication networks, characterized by a great heterogeneity. On one hand, many applications require a specific level of assurance from the network. On the other hand, communication networks are characterized by many levels of heterogeneity: network portions managed by different Service Providers; different transmission means, such as cables, satellites, and radios; different implemented solutions, such as ATM, IPv4, IPv6, and MPLS. Moreover, a network may be heterogeneous also from the point of view of users, who can require different services and have a different methods to pay for them. The challenge is to offer end-to-end QoS guarantees over such heterogeneous networks transparently to the users. The requirements are:

- QoS requests should traverse the overall network from the source to the destination, through portions that implement different technologies and different protocols;
- QoS requests should be received and understood by each specific portion where QoS may have a different meaning and interpretation, which depend on used protocols and network features;
- QoS requests should be managed by control mechanisms suited for the aim;
- Each single QoS solution is composed of layered architectures and each layer must have a specific role in QoS provision.

As stated in [18], the overall problem of QoS interworking may be structured into two different actions: vertical QoS mapping and horizontal QoS mapping.

The concept of vertical QoS mapping is based on the idea that a telecommunication network is composed of functional layers and that each single layer must have a role for end-to-end QoS provision. The overall result depends on the QoS achieved at each layer and it is based on the functions performed at layer interfaces. The idea is to define an interface between adjacent layers through which to offer a specific QoS service. For example, if layer 3 implements efficient QoS mechanisms, it is topical that layer 2 can assure a specific service to layer 3; otherwise the implementation of complex QoS mechanism at layer 3 is useless. QoS requirements flow vertically and need to be received, understood, and satisfied by the layer below.

The concept of horizontal QoS mapping, even if linked to the previous concept when implemented, is represented by the need to transfer QoS requirements among network portions implementing different technologies and protocols.

The implementation of both vertical and horizontal mappings requires the use of specific QoS management functions and QoS architectures. As also envisaged in previous sections, the idea is that each single network portion composing the heterogeneous network deserves a peculiar solution. Special tools called QoS gateways and implemented through QoS Relay Nodes can take charge of that [18].

B. QoS Gateways

Today's Internet protocols are not particularly suited for heterogeneous environments, in particular if characterized by very long delay paths and possible link disruptions, as in DTN networks. In more detail, the heterogeneity introduces the need of proper architectures to manage the inter-working of satellite/wireless/cable network portions and to connect heterogeneous, possibly non-IP end systems. A possible reference concerning networking is represented by the Broadband Satellite Multimedia (BSM) architecture, developed by the European Telecommunications Standardization Institute (ETSI). It separates the layers identified as Satellite Dependent (SD) (data link and physical layer) from the ones identified as Satellite Independent (SI) (IP and upper layers). The interface between SI and SD layers is defined through SI-SAPs (Satellite Independent – Service Access Points). A possible action is to generalize the interface also for radio and cable interfaces so getting a common management of the lower layers interfaces. The new interface can be called TI-SAP (Technology Independent – Service Access Point), as done in [18]. Within the TI-SAP, as well as within any other interface of this type, there is the need of QoS Mapping. The aim is to define a mapping between various QoS definitions and capabilities used in the different network portions. The mapping mechanism and implementation should give origin to a “seamless” communication. The mapping should be provided both “vertically” and “horizontally”.

Within the mentioned architecture, the design of specialized protocols is topical. Novel solutions may be applied at each protocol layer. Physical and data link layers are fundamentally concerned with the implementation of resource allocation schemes. The network layer has to efficiently use the bandwidth offered by the lower layers and implement QoS reservation and QoS mapping mechanisms. Transport and application protocols must efficiently use the services offered by the network layer. In this view, a cross-layer based approach is envisaged. The cross-layer definition allows a protocol entity to exploit the knowledge of a set of available parameters (measured or estimated) from the underlying layers and, hence, to provide an optimization framework involving all the layers. More specifically concerning resource allocation, the aim is to find efficient and flexible allocation and reservation schemes, which also include congestion control and monitoring. As said, this topic is strictly connected with the implementation of physical and data link layers. The need to guarantee a specific QoS has implied the development of dynamic bandwidth allocation techniques, which take into account the current status of the channel.

The features mentioned above should be developed and implemented within QoS Gateways, whose design may also be object of a dedicated research project. A similar approach is already applied in EU projects [19] and [20]. The way to implementation is long and steep but some literature can help fix some basics. [18] has proposed a network node, called Quality of Service Relay Node

(QoS-RN), which is a basic QoS Gateway and includes the essentials of the features mentioned above. QoS-RN should be located among networks that implement different technological solutions. QoS-RN may also implement extended functions within the Relay Layer including transport and application layer enhancements such as PEPs (Performance Enhancing Proxies) functionalities. Figure 5 shows the architectural proposal reported in [18] to implement the QoS-RN between two networks. Network B in the middle deserves a dedicated special protocol stack to be optimized and the Relay Layer takes care of that. It means that the Relay Layer may implement, in case of need, two different protocol stacks: one towards Network B and one towards the external parts (Networks A and C).

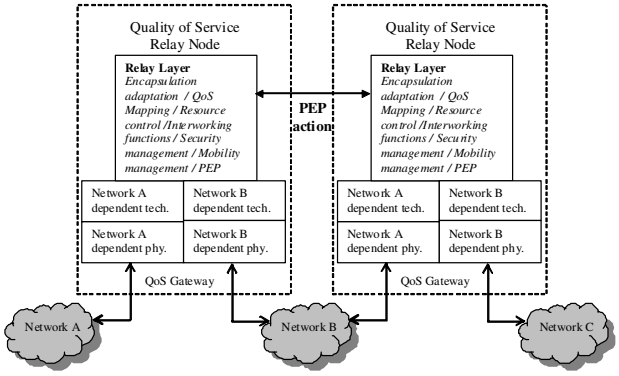

Figure 5. QoS Gateway action

As said in section II Bundle protocol is also suitable to act as overlay on top of heterogeneous networks as shown in Figure 1. The role of the Bundle layer as gateway to join different networks is mentioned also in [21] and in [10], where DTN architecture is presented also as a framework for dealing heterogeneity. Actually, the similarity of the architectures reported in Figure 1 and Figure 5 is immediate. The Bundle layer acts similarly to the Relay Layer, at least from the position in the stack. The idea may be merging the QoS Gateway with the DTN node from the functionalities viewpoint so to create a device that can provide the quality of service, mobility, and security capabilities of the QoS Gateways and the power of managing intermittent and disruptive links as well as large and variable delays of the DTN nodes. Interactions between QoS, mobility and security had been often ignored in the past and need further investigation. The idea of a new intelligent DTN gateway may be the object of future research activity

V. CONCLUSIONS

In the paper, we have examined the pros and cons of the DTN architecture when applied to satellite communications. The analysis has been carried out in a differential way, by highlighting both analogies and differences with PEPs, and focusing the attention on three aspects: network architecture, security and QoS. The analysis confirms that a DTN approach to satellite communications can be seen as an evolution and extension of the current PEP technologies, particular useful in the presence of intermittent and disruptive channels and/or a high level of network heterogeneity. In brief, the advantages offered by DTN are: a better resilience to long disruption, the ability to cope with intermittent channel availability, the possibility to implement both hop-by-hop and end-to-end security, and a greater flexibility in the design and implementation of advanced QoS techniques. On the other hand, there are some issues that still need to be addressed, such as flow and congestion control at bundle layer.

REFERENCES

[1] Y. Hu and V.O.H. Li, “Satellite-based internet: a tutorial,” IEEE Commun. Mag., vol. 39, no. 3, pp. 164-171, Mar. 2001.

[2] J. Border, M. Kojo, J. Griner, G. Montenegro, Z. Shelby, “Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations”, IETF RFC 3135, June 2001.

[3] Technical Report on Performance Enhancing Proxies (PEPs) for the European ETSI Broadband Satellite Multimedia (BSM) working group. ETSI Report TR 102 676 (Sept. 2009). http://portal.etsi.org.

[4] A. McMahon, S. Farrell, "Delay- and Disruption-Tolerant Networking", IEEE Internet Computing, vol. 13, no. 6, pp. 82-87, Nov./Dec. 2009

[5] V. Cerf , A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, H. Weiss “Delay-Tolerant Networking Architecture”, IETF RFC 4838, Apr. 2007.

[6] K. Scott and S. Burleigh, “Bundle Protocol Specification”, IETF RFC 5050, Nov. 2007;

[7] W. Ivancic, W.M. Eddy, D. Stewart, L. Wood, J. Northam, C. Jackson, , “Experience with Delay-Tolerant Networking from Orbit”, to be published on Int. J. Satell. Commun. Network.

[8] C. Caini, P. Cornice, R. Firrincieli, and D. Lacamera, “A DTN Approach to Satellite Communications”, IEEE J. Select. Areas in Commun., special issue on Delay and Disruption Tolerant Wireless Communication, vol. 26, no. 5, pp. 820-827, Jun. 2008.

[9] C. Caini, P. Cornice, R. Firrincieli, D. Lacamera, M. Livini, “TCP, PEP and DTN Performance on Disruptive Satellite Channels” in Proc. of IEEE IWSSC’09, Siena, Italy, Sept. 2009, pp. 371 - 375.

[10] K. Fall, S. Farrell, "DTN: an architectural retrospective", IEEE J. Select. Areas in Commun., vol.26, no.5, pp.828-836, June 2008.

[11] K. Fall, W. Hong, S. Madden, “Custody Transfer for Reliable Delivery in Delay Tolerant Networks”, Technical Report IRB-TR-03-030, Intel Research, Berkeley, July 2003, pp. 1-6. Available at DTNRG web site.

[12] C. Caini, R. Firrincieli, M. Livini, “DTN Bundle Layer over TCP: Retransmission Algorithms in the Presence of Channel Disruptions”, in J. of Commun. (JCM), Academy Publisher, special issue on Delay Tolerant Networks, Architecture, and Applications, Vol. 5, N. 2, pp. 106-116, Feb. 2010.

[13] S.F. Symington, et al, “Bundle Security Protocol Specification”, draft-irtf-dtnrg-bundle-security-08, IETF draft. March 2008

[14] S. Farrell, et al, Delay-Tolerant Networking Security Overview, draft-irtf-dtnrg-sec-overview-06, IETF draft. March 2009.

[15] N. Bhutta, G. Ansa, E. Johnson, N. Ahmad, M. Alsiyabi and H. Cruickshank, “Security analysis for Delay/Disruption Tolerant satellite and sensor Networks”. in Proc. of IEEE IWSSC, Sept. 2009, pp.385-389.

[16] ETSI. Digital Video Broadcasting (DVB); DVB specification for data broadcasting. ETSI EN 301 790 V1.4.1 (2005-04). Interaction channel for satellite distribution systems", 2005-04.

[17] H. Cruickshank, P. Pillai and M. Noisternig, “Security requirements for the Unidirectional Lightweight Encapsulation (ULE) protocol “, IETF Internet Draft (draft-ipdvb-sec-req-09.txt), Aug. 2008.

[18] M. Marchese, “Quality of Service over Heterogeneous Networks”, Wiley & Sons, Chichester, UK, 2007.

[19] Sensei web site: http://www.ict-sensei.org/.

[20] Eu-mesh web site: http://www.eu-mesh.eu/.

[21] Forrest Warthman, “Delay-Tolerant Networks (DTNs)-A tutorial”, May 2003, www.ipnsig.org/reports/DTN_Tutorial11.pdf