INVITED PAPER

Delay- and Disruption-Tolerant Networking (DTN): An Alternative Solution for Future Satellite Networking Applications

Applications of DTN for future satellite networks are discussed in this paper, as well as the relationship between DTN and quality of service (QoS).

By Carlo Caini, *Member IEEE*, Haitham Cruickshank, *Member IEEE*, Stephen Farrell, and Mario Marchese, *Senior Member IEEE*

ABSTRACT | Satellite communications are characterized by long delays, packet losses, and sometimes intermittent connectivity and link disruptions. The TCP/IP stack is ineffective against these impairments and even dedicated solutions, such as performance enhancing proxies (PEPs), can hardly tackle the most challenging environments, and create compatibility issues with current security protocols. An alternative solution arises from the delay- and disruption-tolerant networking (DTN) architecture, which specifies an overlay protocol, called bundle protocol (BP), on top of either transport protocols (TCP, UDP, etc.), or of lower layer protocols (Bluetooth, Ethernet, etc.). The DTN architecture provides long-term information storage on intermediate nodes, suitable for coping with disrupted links, long delays, and intermittent connectivity. By dividing the end-to-end path into multiple DTN hops, in a way that actually extends the TCP-splitting concept exploited in most PEPs, DTN allows the use of specialized protocols on the satellite (or space) links. This paper discusses the prospects for use of DTN in future satellite networks. We present a broad DTN overview, to make the reader familiar with the characteristics that differentiate DTN from ordinary TCP/IP networking, compare the DTN and PEP architectures and stacks, as a preliminary step for the subsequent DTN performance assessment carried out in practical LEO/GEO satellite scenarios. DTN security is studied next, examining the advantages over present satellite architectures, the threats faced in satellite scenarios, and also open issues. Finally, the relation between DTN and quality of service (QoS) is investigated, by focusing on QoS architectures and QoS tools and by discussing the state of the art of DTN research activity in modeling, routing, and congestion control.

KEYWORDS | Delay- and disruption-tolerant networking (DTN); performance enhancing proxies (PEPs); quality of service (QoS); satellite communications; security

Manuscript received October 15, 2010; revised March 4, 2011; accepted May 17, 2011. Date of publication July 22, 2011; date of current version October 19, 2011.

C. Caini is with the Department of Electronics, Computer Science, and Systems (DEIS), University of Bologna, 40125 Bologna, Italy (e-mail: carlo.caini@unibo.it).
H. Cruickshank is with the Centre for Communication Systems Research, University of Surrey, Guildford GU2 7XH, U.K. (e-mail: H.Cruickshank@surrey.ac.uk).
S. Farrell is with the Department of Computer Science, Trinity College Dublin, Dublin, Ireland (e-mail: stephen.farrell@cs.tcd.ie).
M. Marchese is with the Department of Communications, Computer and System Sciences, University of Genoa, 16145 Genova, Italy (e-mail: mario.marchese@unige.it).

Digital Object Identifier: 10.1109/JPROC.2011.2158378

I. INTRODUCTION

Satellite communications present some distinctive features that deserve to be briefly analyzed. On the positive side, they offer a very effective way to quickly provide coverage of large areas. Satellites can offer ubiquitous Internet access at reasonable cost in developing countries and in

sparsely populated areas, thus helping reduce the digital divide. Moreover, satellite communications are essential to support rescue teams in case of natural disasters, such as earthquakes or flooding, when the terrestrial communication infrastructure is seriously damaged. On the other hand, satellite systems, and in particular satellites in geosynchronous (GEO) orbits, have to cope with a series of challenges at different layers of the network stack. In particular, if we focus on transport and upper layers, performance is challenged by the following [1]: long round trip times (RTTs), especially for GEO systems (about 600 ms); the likelihood of segment losses due to residual errors on the satellite link; possible channel disruptions, especially for mobile terminals, due to obstructions (buildings, tunnels, etc.); and coverage issues at high latitudes and in challenging terrain. Even more challenging problems arise from the deep-space environment, as well as from other environments characterized by very long delays and intermittent connectivity.

A possible solution to these problems comes from the modification of the transport layer. In this view, although an end-to-end approach, i.e., the use of a specialized transport protocol on both end nodes, is possible, it is not practical for the general Internet. Since satellite clients represent a niche for general content providers, there is no real advantage for such providers in introducing a modification of the standard protocol stack just to offer a better quality of service (QoS) to the satellite using population.

In order to apply transport protocol variants suitable for the satellite link, the common solution is to use so-called performance enhancing proxies (PEPs), or protocol accelerators, based on the TCP-splitting technique [2], [3]. PEPs are intermediate nodes, inserted either at one end of a satellite link (integrated PEP), or more frequently, at both ends (distributed PEPs), to isolate the satellite link characteristics from the rest of the network. In short, PEPs split the original end-to-end TCP connection into two (integrated) or three (distributed) separate TCP connections, thus allowing the use of a suitable TCP variant on the satellite link. PEPs are an effective solution and have the important advantage of being transparent to end users. However, they violate the end-to-end semantics of transport protocols and have other serious disadvantages, for example, related to security, since TCP splitting is incompatible with standard Internet security mechanisms such as IPsec [4], which encrypts the TCP headers that the PEP must read to improve performance on the satellite link.

The delay- and disruption-tolerant networking (DTN) architecture [5]–[7] introduces an overlay protocol that interfaces with either the transport layer or lower layers. Each node of the DTN architecture can store information for a long time before forwarding it. Thanks to these features, DTN is particularly suited to cope with the intermittent connectivity provided by a single low earth orbit (LEO) satellite (e.g., for data sensing) or incomplete constellations (e.g., for vehicle and goods tracking) [8]. DTN can also represent a valid alternative to PEPs in GEO systems (as shown in [9] and [10] and further discussed below). Finally, DTN is essential in "data mule applications" characterized by the absence of a continuous path between the source and the destination.

This paper investigates how the DTN architecture can tackle the challenges of future satellite communications. For this purpose, we focus on the most relevant features of DTN, as applied to the satellite field. Sections cover DTN architecture and protocols, a comparative evaluation against PEP-based solutions, DTN security, and QoS, both of which are relatively open research areas. We highlight the novel aspects of the DTN approach and show the feasibility of using DTN for satellite networking.

Our overall aim is threefold: first, to introduce the DTN concept for the general readership; second, to make the satellite communications expert aware of the opportunities offered by DTN; and last, to convince the reader who is knowledgeable about DTN, but less familiar with satellite communications, that this represents a potentially important application field for DTN.

II. DTN OVERVIEW

The origin of the DTN concept lies in a generalization of requirements identified for interplanetary networking (IPN), in particular, for situations where Mars orbiting spacecraft could act as a data relay for landers [11]. In such situations, one faces latencies measured in tens of minutes, as well as limited and highly asymmetric bandwidth. While initially only the IPN use case was under consideration, Fall [12] effectively rechristened the IPN as the DTN by highlighting that the same approach had benefits when applied to challenging networking scenarios on Earth as well as in deep space. In addition to the deep-space use case, three main classes of terrestrial DTN applications have been widely studied: military tactical networking [13], sparse sensor networks [14], and networking in developing or otherwise communications-challenged regions [14]. The evolution of the DTN architecture and protocols has been the subject of recent journal articles [5], [15] and was also covered in a 2006 book [16].

Organizationally, the DTN architecture and protocols have been mainly developed by the Internet Research Task Force's (IRTF) DTN Research Group (DTNRG) [17], though there is also a partly overlapping Consultative Committee on Space Data Systems (CCSDS) DTN working group [18] developing specifications for the use of DTNRG protocols in deep-space missions. The DTNRG is an open research group with participants from many countries and disciplines, while contributors to CCSDS tend to be working for, or with, space agencies. DTNRG participants have developed a number of open- and closed-source implementations of DTN protocols that have been used in a wide variety of laboratory and real-world tests [19]. In the United States, between 2004 and 2009, DARPA funded a

disruption-tolerant networking program [20], with a focus on scenarios where links suffered frequent but usually short disruptions rather than the long light trip times (LTTs) involved in deep space. Researchers generally treat both DTN acronym expansions (delay- and disruption-tolerant networks) as synonyms. Within the European Union, some Framework 7 projects (e.g., N4C [21]) have more recently been funded as part of the FIRE activity on Future Internet [22].

In terms of applicability to satellite communications, the fact that DTN has a wide variety of other applications is a benefit, since this means that generic networking technologies can be used to handle satellite link challenges, rather than having to develop satellite-specific solutions, as has to date been done with PEPs. However, it must also be acknowledged that as with satellite communications, most current DTN use cases are also niches, so DTN is not (yet) a "mainstream" Internet technology, but the DTN architecture is designed to handle a very broad set of use cases and as will be seen can offer benefits for satellite networking.

A. DTN Architecture and the Bundle Protocol

The DTN architecture [6] is based on the introduction of an overlay above transport or other lower layer protocols. The essential point is that in such an overlay, delays and disruptions can be handled at each DTN "hop" in a path between a sender and a destination. Nodes on the path can then provide the storage necessary for application data before forwarding that to the next node on the path. For example, any required retransmissions in an Automatic Repeat re-Quest (ARQ) scheme [23] may come from an intermediate node, and no end-to-end connection is required between the sender and destination. Thus, the main benefit of protocols implementing the DTN architecture is that they do not require the contemporaneous end-to-end connectivity that TCP and other standard Internet transport protocols require in order to reliably transfer application data.

The bundle protocol (BP) [7] has been designed as an implementation of the DTN architecture and is by far the most broadly used DTN protocol. The basic unit of data in the BP is a "bundle" which is a message that carries application layer protocol data units (APDUs), sender and destination names, and any additional data required for end-to-end delivery.

The BP can interface with different lower layer (usually transport) protocols through convergence layer adapters (CLAs) as shown in Fig. 1 [6], [24]. Various CLAs have been defined, including for TCP [25], UDP [26], the Licklider transmission protocol (LTP) [27], [28]. Additional CLAs including NORM [29], DCCP [30], Bluetooth, and raw Ethernet have been implemented in the most commonly used open-source implementation of the BP, called DTN2 [31]. With the BP, each DTN node on a path may use whatever CLA is best suited for the next forwarding operation.

The DTN architecture has many novel aspects when compared to traditional TCP/IP-based networks. The most prominent, when dealing with satellite communications, are summarized below.

B. DTN as an Overlay

First, although the TCP protocol is not necessarily replaced, its role changes. In particular, the DTN architecture is suited for acting as overlay on top of a heterogeneous network consisting of different segments, such as wireless sensor/*ad hoc* networks, wired Internet, wireless local area networks (LANs), satellite links, etc. By installing a bundle protocol agent (BPA) on endpoints and nodes at the border of homogeneous segments, the end-to-end path can (if necessary) be divided into many DTN hops. On each DTN hop different CLAs can be used, or, when the same CLA is used for a bundle on both inbound and outbound hops, which is common, different variants of the same protocol (e.g., variants of TCP) can be used.

Readers familiar with satellite communications can easily see that the DTN multihop architecture can be seen as a generalization of the TCP-splitting concept widely used in satellite PEPs. This aspect is further investigated in Sections IV and VI.

C. Information Storage at Intermediate Nodes

A second, but no less important, difference between DTN and traditional TCP/IP networking is related to information storage. In standard networks, which assume continuous connectivity and short delays, routers perform nonpersistent (short-term) storage and information is persistently stored only at end nodes, i.e., outside the network core. This is because, dealing with reliable transmission, information is supposed to be easily retrieved directly from the source. Of course, this may not be the case in challenged networks. Therefore, to deal with long RTTs and channel disruptions, and to cope with the extreme case of the absence of end-to-end connectivity, in DTN networks information is persistently (long-term) stored at intermediate DTN nodes.

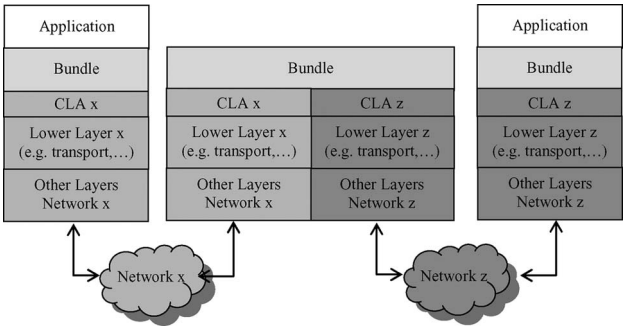

Fig. 1. *DTN architecture and protocol stack.*

This feature differentiates the DTN architecture also from PEPs. In PEPs, some segments are stored, but this storage is temporary and is aimed at synchronizing the incoming with the outgoing segment flows. In contrast, bundles (which are usually larger than segments) can be stored at intermediate nodes for extended durations, and, when the custody option (see Section II-D) is enabled, can be saved in persistent memory such as a local hard disk. This makes DTN much more robust against disruptions, disconnections, and temporary node failures (e.g., reboots). On the other hand, in-network bundle storage raises storage congestion issues that still need to be addressed. While the BP includes some "expiry" controls, so that expired bundles are eventually deleted from in-network storage, there may still be cases where a node does not have sufficient storage available and work on generic and scalable ways to handle this is still ongoing in the DTN community (see Section VI-D3).

D. Custody Transfer

In some DTN use cases, the original sender of a bundle will never have the opportunity to retransmit the application data, for example, due to physical movement away from the network, or for power management reasons (if the sender will be powered off until after the bundle expires). In order to handle this, the BP supports the concept of a node taking "custody" of a bundle [24] which essentially means that the custodian is taking responsibility for any required retransmissions. In this way, even if the sender is no longer attached to the network at all, a bundle can be retransmitted in order to handle disruption in the network. Locating custodians in proximity to links prone to disruption can also greatly reduce overall latency. In the BP, a sending node can request that other nodes on the path take custody by signaling this in the bundle header. When a node accepts custody, it signals back to the previous custodian (also reported in the bundle header) so that the previous custodian can release storage since it no longer needs to keep a copy of the bundle. The custody option increases reliability and is particularly useful whenever the sender has limited memory and/or power resources, as in sensor networks, or has good reasons not to keep in its memory sensitive information, such as in military applications.

E. Proactive and Reactive Bundle Fragmentation

An interesting feature of the BP is the possibility of fragmenting bundles. The DTN architecture and the BP define two types of fragmentation, namely, proactive and reactive. The former has been conceived to cope with intermittent periodic connectivity, where there may be a limit on the 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, for example, 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.

In contrast, reactive fragmentation works *a posteriori*, when disruptions interrupt an ongoing bundle transfer. In order not to retransmit successfully received data, the partially transmitted bundle is split into two "fragments." The first contains data already sent, the second the remaining data. At link reestablishment, only the second fragment is transmitted. Bundle fragments are treated as ordinary bundles and consecutive fragmentations are possible. Since fragments are bundles, they may be routed independently of one another. Reactive fragmentation is particularly useful when disruptions are relatively frequent, as in satellite communications with mobile terminals, when obstacles (buildings, tunnels, etc.) may prevent satellite signal reception and when large bundles are to be transmitted. Both proactive and reactive fragmentations are distinctive features of DTN.

F. Late Binding

In the BP, sources and destinations are named as endpoint identifiers (EIDs) and are syntactically represented as uniform resource identifiers (URIs). There is no concept of an address in the BP, and BP routing is based purely on EIDs. Clearly, CLAs do make use of both names and addresses, for example a TCP CLA might use the domain name system (DNS) to lookup an IP address in order to establish a contact, but the BP itself does not make direct use of IP addresses. This allows for so-called "late binding" where, for example, with a destination EID that includes a DNS name, only the CLA for the final DTN hop might have to resolve that DNS name to an IP address and routing for earlier hops can be purely name based. Late binding can be advantageous in networks where some nodes cannot access the kind of infrastructure offered by, for example, the DNS. A URI scheme "dtn:" has been registered for use with the BP, and an EID might look like "dtn://dtn.example.com/myApp" and the final forwarding step for a bundle destined for that DTN node might involve looking up the IP address for dtn.example.com and connecting to the standard TCP port (4556) [32] for the BP on that host.

G. Routing

As clearly stated in [33], "*the routing objective of traditional routing schemes has been to select a path which minimizes some simple metric (e.g. the number of hops). For DTN networks, however, the most desirable objective is not immediately obvious.*" So, metric definition is not trivial. Clearly, an important objective for DTN is to increase the probability of bundle delivery, but reducing the delivery delay is also usually important for applications. Storage management is also related to routing, as is energy efficiency. DTN routing schemes have to deal with the fact that nodes are not constantly connected, and the concept of the "contact" [11] has been defined as a duration during which one node can send to another with a certain bandwidth expectation.

For example, for a LEO satellite a contact would map to a pass over a ground station which will have a known duration and bandwidth. The contact volume or capacity is then the amount of data that can be transmitted in that contact and is essentially the product of the contact duration and bandwidth. Note that while the concept of a contact is very useful for routing schemes, contacts can fail to occur, or encounter disruption, in which case the BP's custodial retransmission may be used to recover, but some routing schemes may also recalculate routes as a result of such failure. The routing issue deserves great attention and is quite closely linked to QoS provision. For these motivations, Section VI contains a classification of routing schemes and a detailed analysis of the state of the art.

H. DTN Experiments

In addition to the many studies, simulations, and emulations, there have been a number of real-world experiments with DTN, for both terrestrial and space scenarios. On the ground, DTN has been investigated for military tactical networking [13] and for environmental monitoring [14], [16]. In space, DTN has been flown on the EPOXI spacecraft [34] in order to increase its technology readiness level (TRL) and on the International Space Station [35] and with the United Kingdom's part of the disaster monitoring satellite (DMC) constellation [8]. In all these cases, DTN has been found to be effective, and even more advanced DTN mechanisms such as reactive fragmentation have proved to be useful.

III. DTN AS AN EVOLUTION OF TCP-SPLITTING PEPS

Generally, two PEP configurations are possible: "distributed," with two PEPs at the edges of the satellite link [Fig. 2(a)], and, less frequently, "integrated," with just one PEP at the satellite gateway [Fig. 2(b)]. Although there are many kinds of PEPs, here we focus on TCP-splitting PEPs, which are the most common. Each TCP-splitting PEP splits the end-to-end connection into two parts. Therefore, in distributed PEPs, we have three TCP connections: from the sender to the gateway PEP; between the two PEPs; and from the satellite terminal PEP to the satellite receiver. The first and the last are usually on standard wired links and use ordinary TCP (e.g., NewReno). The intermediate satellite connection uses a different TCP version (or another transport protocol) specialized for the satellite link. For integrated PEPs, we have just two connections, one wired (using normal TCP), from satellite sender to integrated PEP, the other on the satellite link where a modified version of TCP is in order. Integrated PEPs have no need of a PEP at the user premises, which is a significant advantage considering that one satellite gateway may have hundreds or thousands of connected terminals. On the other hand, to keep an integrated PEP transparent to end users, the transport protocol on the satellite link is limited to enhanced versions of TCP, and more specifically to TCP variants that are compatible with standard TCP receivers.

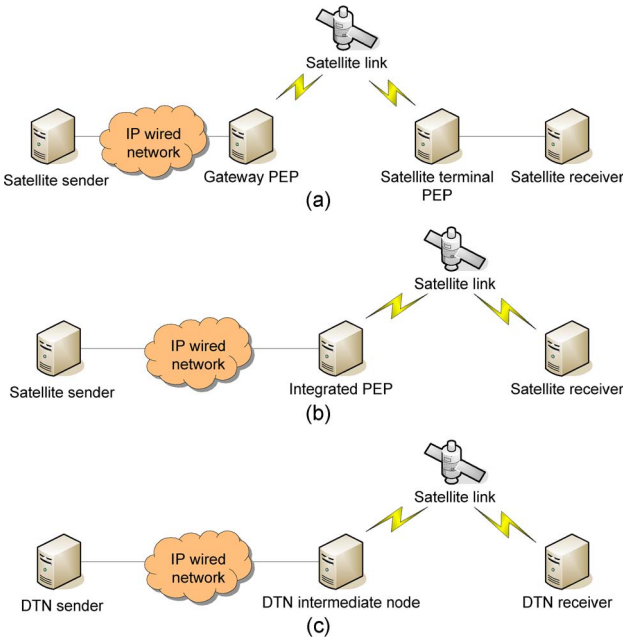

Fig. 2. ***PEP and DTN architecture comparison: (a) distributed PEP; (b) integrated PEP; (c) DTN network.***

For both distributed and integrated PEPs it is possible to show a corresponding DTN architecture which uses a CLA for TCP, included in the BP, in this case. We focus here on the integrated architecture, which is better suited for a direct comparison. A DTN network that corresponds to the integrated PEP in Fig. 2(b) is shown in Fig. 2(c). The corresponding protocol stacks are given in Fig. 3(a) and (b), respectively.

By comparing integrated PEPs and DTN, the following commonalities are apparent:

- both have two transport layer connections, the first wired and the second on the satellite link;
- both can use a TCP variant suitable for satellite links.

These characteristics are instrumental to offer good performance to satellite users, as shown in the next section.

There are, however, also some important differences.

- The DTN solution is not as transparent: the BP must be installed on end-nodes.
- TCP splitting violates end-to-end TCP semantics, because intermediate PEPs must operate at transport and application layers, while the protocol stack reserves these functionalities to end nodes only. In DTN, this drawback is avoided, as the role of TCP is redefined by the BP insertion.
- TCP splitting is incompatible with IPsec (see Section V).

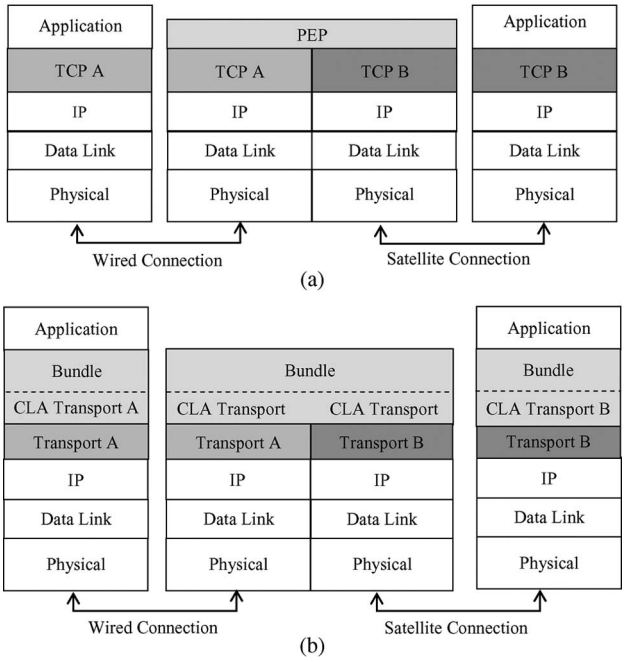

Fig. 3. ***PEP and DTN protocol stack comparison: (a) integrated PEP; (b) DTN.***

IV. DTN AND GEO/LEO SATELLITE COMMUNICATIONS: APPLICATION SCENARIOS

The DTN architecture can be used for GEO satellites with fixed terminals, GEO with mobile terminals, and for LEO satellites. These scenarios are separately examined below, however, we first consider the different levels of end-to-end connectivity in these scenarios. In GEO satellite with fixed terminals, continuous end-to-end connectivity is usually available, so alternatives to DTN, specifically PEPs, are available and offer good performance. In the second scenario, terminal mobility makes the environment more challenging, as channel disruptions can be caused by obstacles such as large buildings or tunnels. In this case, DTN can offer increased resilience, but the advantage over PEPs must be carefully evaluated, as it depends on the disruption characteristics of the channel. The third scenario is the most challenging, being characterized either by intermittent scheduled end-to-end connectivity, for example, for LEO observation satellites, or, by the total absence of end-to-end connectivity, for example, when data are transferred from a source to a destination by means of a LEO satellite that is never visible to both at the same time (the LEO satellite works as a "data mule" [36]). In these cases, the advantages of DTN are outstanding, as neither PEPs nor other end-to-end solutions are able to cope with such disruption.

Without loss of generality, we initially focus on the case of a fixed user connected to the general Internet through a GEO satellite. There are large delays (RTT of about 600 ms), possible congestion on the wired IP network and possible segment losses due to residual bit errors on the satellite channel (packets with one or more corrupted bits are discarded). This scenario is the least favorable to DTN because, without terminal mobility, we can reasonably assume that the satellite link is less disrupted and therefore there is almost always a continuous path between end nodes. For this scenario, it has been shown in [9] that the DTN approach is nonetheless competitive with current solutions, in particular PEPs. Here we present additional results, obtained with more recent versions of the same testbed and tools. Before proceeding, a brief description of these is necessary.

A. TATPA Testbed

The testbed on advanced transport protocols and architectures (TATPA) is shown in Fig. 4. It reproduces the characteristics of heterogeneous networks that include a satellite link and is based on a set of Linux PCs, whose kernel (version 2.26) has been patched with the multi-TCP package implementing TCP-Hybla [37].

A Linux implementation of an integrated TCP-splitting PEP, PEPsal [38], can be enabled on the router R2, in conformance with the topology given in Fig. 2(b). TCP A and TCP B in Fig. 3(a) are NewReno and Hybla, respectively. The corresponding DTN configuration [Fig. 2(c)] is obtained by installing the DTN2 BP reference implementation (version 2.7) [31] on the satellite sender and receiver, and on R2, which, in this case, acts as DTN intermediate node. For fairness in comparison with PEPsal, transports A and B in Fig. 3(b) are NewReno and Hybla. A satellite emulator adds the desired delay (287.5 ms one way for GEO) and packet error rate (PER, 0% or 1% in tests). Satellite link disruptions are emulated from real traces. DTN performance evaluation used the DTNperf_2 tool, included in the DTN2 package.

B. GEO Satellites With Fixed Terminals

Our first tests assess the performance achievable on a 180-s satellite data transfer in terms of goodput, i.e., the

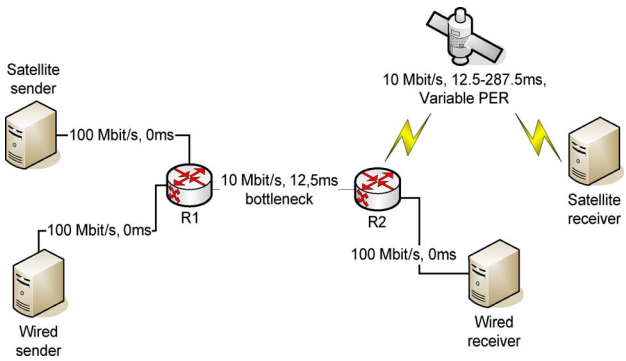

Fig. 4. ***Logical layout of the TATPA testbed.***

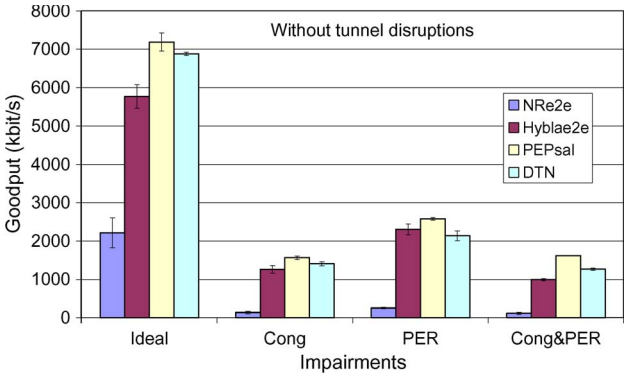

Fig. 5. ***GEO satellite with fixed users (no disruptions); goodput of a single satellite connection (RTT = 600 ms); averaged values, 90% confidence intervals.***

amount of application layer data transferred per time unit (Fig. 5). Four techniques are compared (end-to-end NewReno, end-to-end Hybla, PEPsal, and DTN) in four different environments with increasing challenges (ideal, congested, nonzero PER, and nonzero PER plus congestion). We examine the four scenarios individually.

1) Ideal Channel: This case is a baseline with just a long RTT (600 ms, 25 on R1-R2 link plus 575 on the satellite link). NewReno is not able to fully exploit the 10-Mb/s satellite bandwidth in the first 180 s. End-to-end Hybla performs better (5.8 Mb/s), as it was specifically designed to counteract long RTTs. The best performance is achieved by PEPsal, which takes advantage of both Hybla and the reduced RTT (575 ms) on the satellite connection. DTN reaches basically the same performance as PEPsal, due to the similarity between the two architectures.

2) Congestion: This environment shows the "RTT unfairness" problem, typical when encompassing both wired and satellite links [1]. Here, the satellite data transfer (RTT = 600 ms) is severely impaired by five short RTT (25 ms) connections competing on the "wired IP network" (i.e., on the R1-R2 link).[1] The impact on NewReno is dramatic, as goodput falls to almost zero. In contrast, the other techniques are quite close to the "maximum fair share," i.e., the bottleneck bandwidth divided by the total number of competing connections, which is 1.66 Mb/s (10 Mb/s divided by 6), and which can be considered as the ideal target. This should be ascribed for end-to-end Hybla to its improved congestion control, while for both PEPsal and DTN to the isolation of the satellite link from the wired part of the IP network.

[1]The ratio of one satellite over five wired connections is arbitrary, but aims at reproducing the disparity that can be found in realistic environments, where the amount of TCP traffic directed to a satellite receiver is largely dominated by competing terrestrial TCP traffic.

3) PER: This is as in the ideal case, but where the satellite link is affected by a very large PER (1%). This can happen with bad propagation conditions on the satellite link. Although the origin of the problem is different, results are qualitatively close to the congestion case: end-to-end NewReno performs very badly, while other techniques perform better (although goodput is far from the available bandwidth). In contrast to the congestion case, here the better performance of PEPsal and DTN is mainly due to the use of a TCP variant optimized to the satellite link, and not to the isolation of the satellite channel from the wired IP network.

4) PER and Congestion: This environment suffers from congestion and PER simultaneously. Results are close to the congestion-only case, which shows that congestion is the dominant impairment.

C. GEO Satellite With Mobile Terminals

In the second scenario, satellite terminal mobility introduces channel disruption as an additional impairment. In this case, DTN's possible advantage depends on the disruption duration and frequency. In particular, with the BP using a TCP CLA, and with the DTN2 reference implementation, we have to distinguish between "short" (≤ 30 s) and "long" (> 30 s) disruptions. The 30-s threshold is set in accordance with the DTN2 reference implementation default, even if DTN RFCs do not specify this value.

Short disruptions do not need a response from the BP and are directly counteracted by TCP retransmission. In contrast, longer disruptions trigger the BP to close the disrupted TCP connection and make a series of attempts to open a new TCP connection. This behavior may seem surprising at a first glance. However, it is exactly what a human user would do when dealing with a connection that seems frozen for a relatively long time. He would close it, and, immediately after, would try to open a new one. The DTN2 implementation of the BP does the same.

The forced TCP closure in turns triggers reactive bundle fragmentation—the bundle is divided into two fragments: the data already successfully transmitted being the first fragment, while the remaining data form the second fragment, which will be transmitted as soon as a new TCP connection is established after disruption ends.

Tunnels on railways or highways provide a practical example of disruption induced by terminal mobility. Here, we focus on railway tunnels, considering real data referring to the Bologna–Florence "Direttissima," one of the most important Italian railway lines [10]. The line is 96 km long, with 33 tunnels of different lengths alternating with open segments. The total length of the tunnels is 37 km, 39% of the line. Assuming a constant speed of 120 km/h, we have 30 "short" disruptions and three "long" disruptions (namely 214, 92, and 553 s).

Our test scenario is a file transfer from a well-connected server to a satellite terminal onboard a train moving from

Bologna to Florence with goodput averaged over the total trip time (2865 s at 120 km/h). The topology is as the same of the previous subsections, with the exception of terminal mobility. For the sake of comparison, we consider the same techniques as before. Results are shown in Fig. 6. Comparing this figure with the corresponding results presented in Fig. 5, two conclusions are worth noting. First, performance is reduced in all cases (note the different *y*-axis scale), which is obvious, considering that the satellite channel is unavailable for 39% of the time because of tunnels. Less evident is that, to this time, we need to add the "restart delay," only after which are the TCP and the BP able to restart transmission after end of the disruption, and also the time needed by TCP to reach a steady state. The longer the RTT is, the worse is this effect [39].

The second notable conclusion is that the qualitative behavior is the same as without disruptions, except that, in this case, it is DTN, and not PEPsal, that offers the best performance. The difference, however, is slight but real. Had we considered a 60-km/h train speed, the longest disruption would have become greater than 1200 s, which is roughly the "maximum tolerable disruption length" with Linux TCP defaults. In this case, all techniques but DTN would have aborted the data transfer [39].

In summary, with a mobile terminal, DTN can become advantageous also in terms of goodput, with its advantage depending on the duration and frequency of disruptions. For very long disruptions, DTN is definitively better. Some experiments applying DTN to GEO sat communications with mobile terminals have already been carried out in the military field [40].

D. LEO Satellites

LEO satellites are characterized by lower orbits with a reduced distance from Earth (160–2000 km). Compared to GEO they offer the advantage of reduced propagation loss and delay, due to the shorter propagation path. However, for an observer on the Earth they are not fixed in the sky, but appear to move very quickly; for example, at 520-km altitude the orbital period is about 90 min. As a result, a single satellite can only provide intermittent connectivity with a ground station, while continuous connectivity requires a constellation of satellites. We focus here on the case of a single satellite, which is the most interesting from a DTN point of view, considering two possible applications.

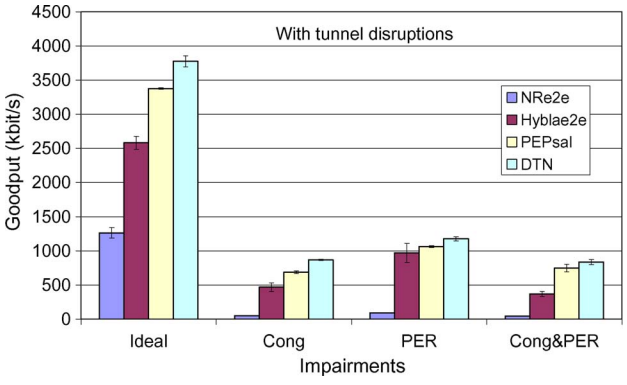

Fig. 6. *GEO satellite with mobile users (disruptions due to railway tunnels); goodput of a single satellite connection (RTT = 600 ms); averaged values, 90% confidence intervals.*

The first scenario applies to communications between one LEO satellite and its ground station. The satellite passes above the ground station on a scheduled basis and, as said before, a communication link (or "contact") can be established only for a short period ("contact window"), thus providing scheduled intermittent connectivity. The short contact window and the limited channel bandwidth impose a limit on the "contact volume," i.e., the total amount of data that can be transferred during a contact window. If very large files (e.g., images) cannot be transferred during a single pass, it would be necessary to divide them into multiple segments to be transmitted over consecutive passes. In this case, the "proactive fragmentation" feature of the BP can be used, automatically diving the large payload into multiple bundles of a predetermined size matching the known contact volume.

In fact, the first experiments of DTN technology on real LEO satellites match this scenario. In these experiments [8], images were downloaded as bundles to ground stations when in visibility. Despite some minor problems, these experiments demonstrated that DTN can facilitate automated routing of sensor data and increase the integration between terrestrial Internet and satellite observation networks.

In the second scenario, the data source and sink are both ground stations distant enough not to be in the LEO satellite coverage area at the same time. In other words, there is never a continuous connection between source and destination. This scenario is the most favorable to DTN applications, as the absence of end-to-end connectivity prevents the establishment of TCP (or TCP-like) connections between the ground stations. Moreover, the lack of end-to-end connectivity also makes UDP transfers impossible without some in-network storage. So the only possible approach is to store-and-forward large amount of data on the satellite. We can distinguish three phases. In the first phase, the source is connected to the satellite and data (e.g., images) are moved on board of the satellite. In the second phase, neither the source nor the sink ground stations are connected to the satellite and data are stored on local persistent memory (e.g., hard disks). In the third phase, the sink ground station connects and data are downloaded from the satellite. Although applied to a satellite environment, this is a typical example of "data-mule" communication, a task for which the BP was designed. The only alternative to DTN in this scenario would be either manual uploading/downloading of files or a specific application including the features of the BP.

V. DTN SECURITY

A. Internet Security Challenges When Using PEPs

When considering security, in addition to the transport layer PEPs (T-PEPs) we have already seen, we also need to consider application layer PEPs (A-PEPs), such as HTTP accelerators. T-PEPs and A-PEPs require access to either transport or application layer headers, respectively. This may be prevented by the use of Internet standard end-to-end security mechanisms, such as IPsec [41] or transport layer security (TLS) [42], which encrypt these headers and therefore prevent PEPs from functioning. More specifically, the use of end-to-end IPsec is incompatible with both kinds of PEPs, because the whole IP payload is encrypted, including both TCP and HTTP headers. By contrast, TLS, working only on the TCP payload, is compatible with T-PEPs (but not with A-PEPs), because TCP headers are not encrypted. It should be pointed out, however, that incompatibility here means that encrypted connections just cannot take advantage of PEPs.

To override the end-to-end incompatibility, one can make use of standard Internet security mechanisms on the satellite link only, which, being of course wireless, is the most prone to eavesdropping and other security attacks. This is illustrated in Fig. 7, where IPsec is applied between gateway PEP and satellite terminal PEP [Fig. 2(a)] [3], [43]. This is similar to using other techniques, such as DVB-RCS [44] or unidirectional link encapsulation (ULE) security [45], but IPsec provides modest additional security functions.

To make T-PEPs compatible with end-to-end security some satellite-specific solutions [4], such as transport friendly encapsulating security protocol (TF-ESP) or modified ESP (M-ESP), modify the encapsulated security protocol (ESP) headers used by IPsec, so as to leave TCP header outside the scope of encryption. Suggestions were also made to use TLS with IPsec in order to protect the TCP header. However, these methods all appear to expose the connection to security threats [46]. An alternative is the use of multilayer IP security (ML-IPsec), which is a more flexible (though complex) solution [47], [48]. It divides the IP datagram into zones, with different protection schemes. The user data part is protected end-to-end by keys shared only between the source and the destination, while the TCP header is protected by keys shared also by PEPs (see Fig. 8). A similar rule could be applied to the HTTP header zone. Despite its technical appeal, ML-IPsec also presents some drawbacks. In particular, it is not transparent, while most PEPs are, and its use is basically dictated by the need to overcome a problem, the incompatibility of IPsec with PEPs, which is urgent but also specific to satellite communications, which are a niche market. By contrast to the IPsec suite, ML-IPsec has not been standardized by IETF or widely deployed in general Internet.

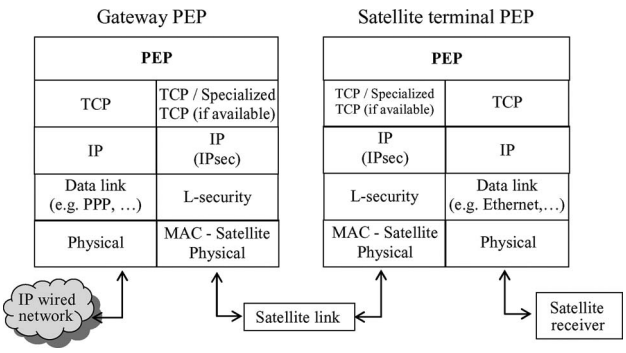

Fig. 7. ***Security solutions with PEPs.***

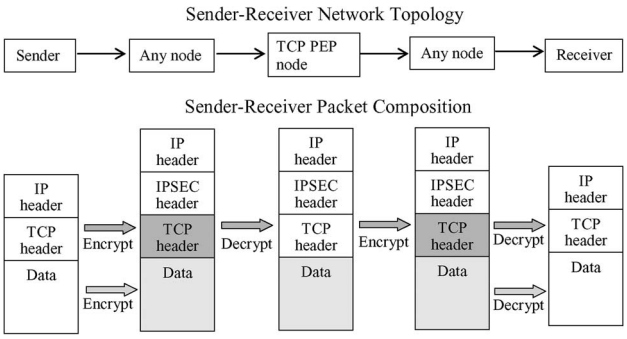

Fig. 8. ***ML-IPsec with PEPs.***

All security techniques have to deal with the key exchange problem. In IPsec, this is addressed by the IPsec Internet key exchange protocol (IKEv2), which is part of the IPsec suite. As an alternative in ML-IPsec, a multicast key management, such as the IETF secure multicast architecture, might be used, as suggested in [49].

In summary, while security on the satellite link is not an issue, end-to-end security in the presence of intermediate entities, like PEPs, is still problematic.

B. Current DTN Security

In challenged networks, standard Internet security mechanisms, such as TLS, IPsec and the variants presented in Section V-A do not perform well, or at all, because of long delays, possible disruptions, and the possible lack of a continuous end-to-end connectivity. The DTN architecture must therefore address this problem with new security tools. Before examining them, let us anticipate that DTN security solutions offer an interesting opportunity to cope with the typical security problems that are met in satellite [50] and space-based networks [51]. At present, the definition of BP security is still in progress but many specifications are already contained in the bundle security protocol (BSP) [52], which defines a set of BP extensions to support hop-by-hop and end-to-end authentication, integrity validation, and confidentiality. The bundle structure consists

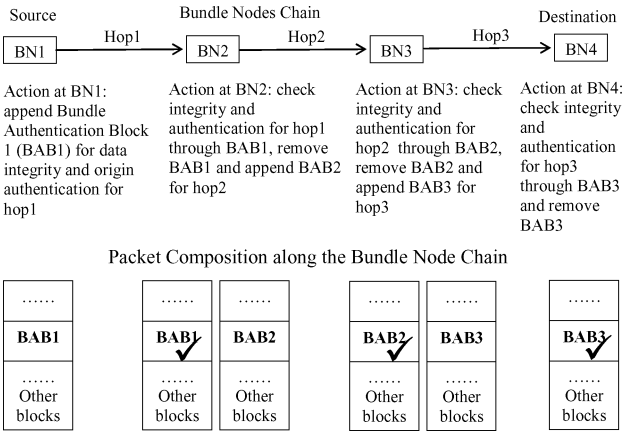

Fig. 9. ***BAB for hop-by-hop authentication and integrity check.***

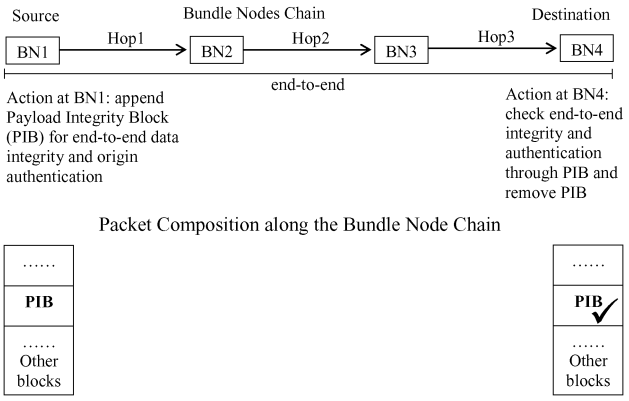

Fig. 10. ***PIB for end-to-end authentication and integrity check.***

of a series of elements called "blocks." In addition, the BSP defines [52] the following security blocks.

- Bundle authentication block (BAB): used to assure the authenticity and integrity of the bundle along a single hop from forwarder to intermediate receiver.
- Payload integrity block (PIB): used to assure the authenticity and integrity of the payload from the PIB security source, which creates the PIB, to the PIB security destination.
- Payload confidentiality block (PCB): indicates that the payload has been encrypted, in whole or in part, at the PCB security source in order to protect the bundle content while in transit to the PCB security destination.
- Extension security block (ESB): provides security for nonpayload blocks in a bundle.

As shown in Fig. 9, BAB is used for hop-by-hop authentication and integrity on one DTN hop. By contrast, PIB is used end to end or over multiple DTN hops (the security source may be the original source or an intermediary, and analogously for the security destination); see Fig. 10. Note that the authentication information in the PIB may be verified by any node in between the PIB security source and the PIB security destination that has access to the cryptographic keys and the revocation status information required to do so. Confidentiality between a security source and security destination is provided by PCB in a manner analogous to PIB; see Fig. 11. Finally, ESB can be used to provide security for nonpayload blocks, not protected by PIB and PCB, such as routing or other metadata. Further details on security architecture in DTN can be found in [52].

C. DTN Security Threats in Satellite Communications

Threats can, as usual, be passive or active. Due to the broadcast nature of satellites, passive threats such as eavesdropping and traffic analysis are major concerns. Active threats (or attacks) are more difficult to implement successfully, but must also be addressed. Examples of active attacks are masquerading, message modification, and denial-of-service (DoS) attacks. Noting that civil space missions ought now be more security aware, CCSDS have produced a "green book" [53] describing generic threats to space missions. A more specific analysis of threats arising from the use of DTN or other store-and-forward communications technologies in space missions has also recently been published [51].

Satellite link characteristics such as long delays, limited bandwidth, and link asymmetry may also make security provisioning and recovery from attack more difficult than in terrestrial networks [45]. Thus, satellite DTN service requirements can be summarized as follows.

- Due to satellite link delays, security processing should be kept to a minimum.
- Due to bandwidth limitation and link asymmetry, security overheads should be kept to minimum.
- As in satellite networks packet losses and link disruptions may be relatively frequent, key exchanges should include reliability mechanisms.

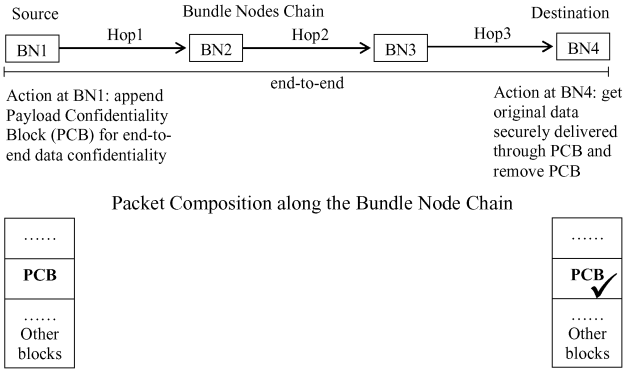

Fig. 11. ***PCB for end-to-end confidentiality.***

The considerations above give rise to research issues such as lightweight key management, lightweight authentication, authorization, and accounting (AAA)-like architecture for authentication and authorization, resilience to DoS attacks, and provision of anonymity to end users [15], [54], detailed in the following.

D. Key Management in DTN

Key management is one of the most difficult problems in DTN security [51] and essentially remains an open question. The reason is that key management generally requires multiple round trips in order to securely exchange or establish keys, and in a generic DTN round trips are always problematic because of long delays and possible disruption.

A DTN key management architecture must allow authentication between previously untrusting parties to establish the shared keys required for the BSP. To this end, various public key mechanisms have been proposed, such as traditional public key infrastructure (PKI) [12], identity-based cryptography (IBC) [55], and proxy-certificate-based schemes [54]. Security processing related to such schemes can be divided into the following phases [56]: entity registration, key establishment, re-keying, and key update. Of course, all phases must be ruled by suitable security policies. As an additional requirement in DTN, key management protocols must support heterogeneity to cater for nodes with varying capabilities including low-powered devices and sensors and should work with scheduled, predicted, and opportunistic contacts.

E. Access Control in DTN

Access control protects the network and its resources from unauthorized access. The relative resource scarcity of DTNs makes access control arguably more important than in the Internet. For example, in space communications link capacities and storage resources are definitively constrained. Standard access control mechanisms may be unworkable in DTN networks. In a seminal DTN paper [12] Fall proposed a public key-certificate-based approach where each participating entity is issued public/private key pairs. This approach uses access control lists though it is claimed to be partially susceptible to node compromise.

In traditional Internet, the AAA architecture [57] is used for access control. It tends to be relatively centralized [58], and can impose single points of failure in DTNs, if not carefully deployed. Distributed and hierarchical architectures may suit multiregional DTNs with region-specific policies. Thus, a workable access control solution for DTN should be simple and scalable, support offline processing, not impose too many communication overheads, and combine key and broader policy management. In [59], a hierarchical architecture is proposed.

F. Resilience to Denial of Service Attacks

DoS attacks can be launched at any layer of the protocol stack and at any network or security service [60], [61]. Since most DTN nodes are resource constrained in some sense, an attacker can usually find some kind of DoS that may damage any given DTN, e.g., the attacker could simply send many large bundles from a well-connected node, through some kind of gateway node into an area with low bandwidth or storage capacity. Compared to a host on the Internet, DoS recovery for a DTN node can be far more problematic, in particular, for remote nodes.

DoS mitigation techniques can often be directly applied in DTN; for example, the LTP protocol (a BP convergence layer) includes a cookie mechanism [62] which, when applied, should limit DoS attempts to on-path attacks. Similar schemes have also been proposed for the BP [63].

G. Identity Protection and Anonymity in DTN

Encryption hides transmitted data, but metadata that relate to user identity may still be visible. To achieve anonymity in DTN researchers defined "anonymous routing protocols" based on the idea of mixed networks or "onion routing" [64].

In some minor respects, the BP is privacy unfriendly, as there is no support for encryption of the primary bundle block which is the main header with the bundle source, creation timestamp, and destination. For this reason (and others), work has begun within the DTNRG on a method for bundle-in-bundle encapsulation [65].

VI. DTN AND QUALITY OF SERVICE

A. The Importance of QoS Over Generic Heterogeneous Networks

The importance of QoS increases with the heterogeneity of the network. As always, applications may require a specific level of performance from the network. Heterogeneous networks, however, may be managed by different service providers, may use different transmission means, such as cables and satellites, and may make use of different networking technologies, 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 use different methods to pay for them. The challenge in generic heterogeneous networks is to offer end-to-end QoS guarantees transparently.

As Marchese states in [66], the overall problem of QoS interworking may be structured into vertical QoS mapping and horizontal QoS mapping. Vertical QoS mapping regards a network as being composed of functional layers so that the overall achieved result depends on the QoS achieved at each layer. Horizontal QoS mapping relates to the need to transfer QoS requirements among network portions implementing different technologies and protocols.

Table 1 QOS Management Functions

Name	Intervention time
Flow/Traffic class identification	Packet time
Traffic shaping	Packet time
Scheduling	Packet time
Flow control	Round trip time
Call Admission Control (CAC)	Connection time
QoS routing	Connection time
Resource allocation and reservation	Connection time ->Long term

The implementation of vertical and horizontal mappings requires the use of QoS management functions. Table 1 contains a list of necessary QoS management functions along with an indication of the time interval at which they may be applied. Packet time is of the order of magnitude of packet arrival (e.g., tens of milliseconds and below), round trip time ranges from hundreds of milliseconds to seconds, and connection time is of the order of magnitude of connection requests arrival and may range from seconds to minutes. Long term indicates larger order of magnitude such as hours, days, up to months and years.

From the QoS architecture viewpoint, the idea is that each part of the network using a specific technology deserves a peculiar solution.

The features mentioned in Table 1 can be implemented within QoS gateways [66] (see Fig. 12). QoS gateways should be located among networks that implement different technological solutions. A similar approach is already applied in European Union projects like Sensei [67] and Eu-mesh [68].

B. QoS Architecture and DTN

We have outlined generic QoS requirements for possibly heterogeneous networks. We now describe the tools available within the DTN architecture and the BP to manage QoS. The connection between two DTN gateways that join different wide-area networks (WANs) is shown in Fig. 13. The similarity of the architectures reported in Figs. 12 and 13 is clear since the BP is also an overlay that can run on top of heterogeneous networks.

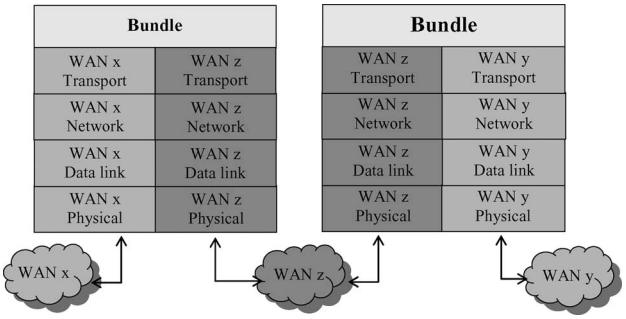

Fig. 13. *DTN gateways connection.*

The role of the BP as a gateway to join different networks is mentioned in a relatively old DTN tutorial [69], in [50], and in [15] where the DTN architecture is presented as a framework for dealing with heterogeneity. An idea introduced in [70] is to merge the QoS gateway with the DTN node to create a device that can provide the QoS, mobility, and security capabilities of QoS gateways and the power of managing intermittent and disrupted links as well as large and variable delays of the DTN nodes. Such a new intelligent DTN gateway may be the object of future research activity.

C. QoS Tools within the DTN architecture

1) Priority Class: Even if the BP does not provide a complete set of QoS management functions as outlined in Table 1, the BP and the DTN architecture contain important tools that can be used to implement QoS in DTNs. The

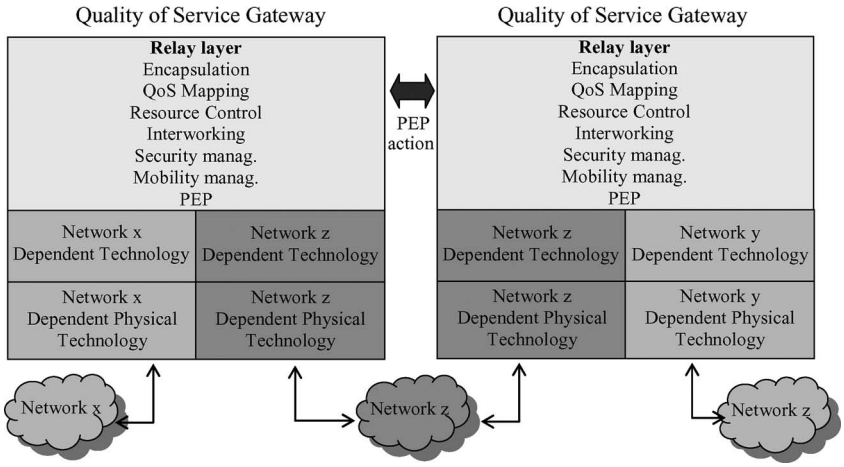

Fig. 12. *QoS gateway action.*

DTN architecture provides three priority levels for bundle delivery: low, medium, and high. Priority levels imply some form of priority-based scheduling within DTN node queues: bulk, which concerns lowest priority bundles; normal; and expedited, whose bundles should be shipped prior to bundles of the other classes. The concept of priority classes broadly matches flow/traffic class identification in Table 1.

2) Delivery Options: The DTN architecture offers a set of delivery options based on bundle status reports (BSR) [6], which can help QoS provision.

- Bundle reception—sent when a bundle arrives at a DTN node.
- Custody acceptance—sent when a node has accepted custody of a bundle.
- Bundle forwarded—sent when a bundle departs from a DTN node after having been forwarded.
- Bundle deletion—sent from a DTN node when a bundle is discarded.
- Bundle delivery—sent from a destination node when a bundle is received at a destination application. Bundle delivery report represents the return receipt.
- Acknowledged by application—sent by the application at a destination node when complete application packet comprising sent bundles has been processed by an application.
- Custody signal—which indicates that custody has been successfully transferred. It is a Boolean indicator and may signal either a successful or a failed custody transfer attempt.

Delivery options may help manage QoS related to scheduling, flow, and congestion control, and routing.

D. QoS and DTN: Research Activity State of the Art

1) Modeling a DTN Network: The first step is finding a proper model for a DTN. Describing a DTN through a graph where vertices (representing DTN nodes) may be interconnected with more than one edge (representing the outgoing links) is a widespread model [6]. This type of graph is called multigraph. Each edge is characterized by a bandwidth and by a delay, and both can be time varying. The peculiar feature of DTNs is that the bandwidth of an edge may be zero for a long period of time during which it is not possible to forward bundles over that edge. This feature does not exclude that link from the routing algorithm because each bundle can be stored in a node for a long time. As stated in Section II, the period of time when the capacity is positive and bundles can be forwarded is called "contact period." Contacts [6] may be persistent, if always available; on-demand, if they are instantiated on request; and intermittent, which are further classified as: scheduled, if established upon an agreement; opportunistic, if unexpected; and predicted, if based on predictions.

Routing and congestion control algorithms may vary heavily depending on the application environment [5] and the related model used to describe the network. Satellite and interplanetary DTN is often modeled through scheduled contacts that are known in advance as well as latencies, with approximations depending on factors such as fading and bandwidth. Emergency DTNs are mainly characterized by opportunistic contacts. Another important aspect in DTN nodes is represented by storage, which is finite and may be modeled through buffers of limited capacity.

2) Routing: Routing in DTNs deserves special attention. For this reason, as anticipated in Section II-G, a detailed classification of DTN routing schemes is presented below. Classifying DTN routing algorithms is somewhat complex. In [33], Jain *et al.* structure routing algorithms depending on the amount of knowledge they use to compute routes: "zero knowledge" algorithms, which do not use any knowledge; "partial knowledge" algorithms, which use partial information to make choices; and "complete knowledge" algorithms that have all information to route bundles. Knowledge is modeled through oracles, abstract entities that provide, separately, information about contacts, queue status, and traffic demands.

Moreover, in [33], Jain *et al.* also report a set of routing schemes, ranging from first contact (FC), in the "zero knowledge" category, to linear programming (LP)-based routing, which act using "complete knowledge" about contacts, queue status, and traffic demands. They also consider a group of schemes between the two extremes, all belonging to "partial knowledge" category: minimum expected delay (MED), which uses statistical information about contacts; earliest delivery (ED) and earliest delivery with local queuing (EDLQ), which use information about contacts, and, in EDLQ, local information about queues; and earliest delivery with all queues (EDAQ), which adds information about queues to the knowledge about contacts but still ignores traffic demands. This classification provides a clear framework that simplifies the comparison between different alternatives.

Another classification comes from [72]–[74]. In these schemes, routing algorithms are classified as ones that replicate bundles ("replication") or as ones that forward only a single bundle copy ("forwarding"). Both have advantages and disadvantages and their suitability depends on the application environment. "Forwarding" schemes are more suitable for resource optimization. By performing no replication they can make better use of bandwidth, storage, and energy. On the other hand, "replication" schemes can provide higher performance in terms of bundle delivery probability and lower delay. The schemes reported in [33] belong to the "forwarding" category, as does delay-tolerant routing for developing regions (DTLSR) [73], which proposes a modification of the classical link state algorithm (LSA) for networks characterized by intermittent contacts.

The scenario that gave rise to DTLSR is networking in developing regions but it may be applied also in other scenarios. Its basic idea is that route calculation should include links that are not currently available (totally ignored by classical LSA) but that could be available in future.

"Replication" schemes include all flooding-based algorithms [74]. Here, they are listed in the order of first appearance in the literature.

- Epidemic routing [75], which is highly reliable but heavily resource intensive as nodes continuously replicate and transmit bundles to new contacts that do not already possess a copy without any attempt to avoid replication.
- ProPHET [76], [77], which uses the nonrandomness of contacts, as often happens in real scenarios, to replicate bundles only if there is a given chance to deliver it.
- Spray-and-wait [78], which "sprays" a limited number of copies into the network, and then "waits" until one of these nodes meets ("contacts") the destination.
- MaxProp [79], evaluated on a real DTN bus network, is based on prioritizing both the schedule of bundles transmitted to DTN nodes and the schedule of bundles to be dropped. The priorities are based on path likelihoods according to historical data and on complementary mechanisms.
- RAPID [71], [72], also evaluated on the same DTN bus network, uses a random variable that represents the contact between two DTN nodes and replicates bundles in decreasing order of their marginal utility at each transfer opportunity. Utility is measured for three separate metrics aimed at minimizing either the average delivery delay or the missed bundle deadline beyond which the bundle is no longer useful, or the maximum delivery delay.

Both ProPHET and MaxProp have been designed for vehicular networks where contacts are relatively random. This randomness is unlikely in most satellite and in interplanetary networks where contacts are typically scheduled. This should be considered when choosing a suitable routing scheme even if some of the features proposed in these schemes appear very useful also for routing in space environments whose peculiarities are evidenced in [34]. One particular scheme that has been designed for the deterministic case of deep-space networking is contact graph routing (CGR) [80], where each node on the path calculates a route from itself to the bundle destination based on a graph.

As mentioned before, routing decisions, in particular in the satellite environment, may depend on different, often conflicting, factors, such as storage, bundle delivery, latency, and energy use. In this view, within the category of forwarding and partial knowledge schemes, Bisio *et al.* [81] suggest a congestion-aware routing approach for DTN interplanetary networks, based on a vector-optimization built on multiattribute decision making (MADM) and aimed at selecting routes by considering different performance indicators at the same time.

In contrast to these rather complex schemes, many DTNs [14] actually make use of static routing, where nodes match the bundle destination against a static table that usually supports wildcard matching. Generally, these tables map to contacts that may be configured in various ways, for example, as "opportunistic" or "always on." The model is usually event driven, so that when a bundle arrives, or a new contact is opened, the set of currently open contacts is checked and the bundle is forwarded if the static routing rule for one of those contacts matches the destination. Both forwarding and replication options may be used with static routing. DTN2 [31] is often used in this manner.

Finally, given the diversity inherent in the different kinds of DTNs that have been envisaged, and in the set of routing schemes that have been developed, it should be clear that no one routing scheme should assume that it will be used for the entire path from bundle source to destination. So there may be a need for some nodes to act as gateways between different routing domains. One relatively common case is where there are some well-connected nodes that make use of static routing and a gateway into a challenged region, inside which some subepidemic routing scheme, like the ones above, is used.

3) Congestion Control: Another topic of ongoing research, related to routing in DTN networks is congestion control, defined in the DTN architecture [6] "*as a means of assuring that the aggregate rate at which all traffic sources inject data into a network does not exceed the maximum aggregate rate at which the network can deliver data to destination nodes over time.*"

In a DTN, congestion can occur either at a CLA, for example, if a TCP session suffers congestion, or due to a shortage of persistent storage within a bundle agent.

Quoting [6]: "*When storage resources become scarce, a DTN node has only a certain degree of freedom in handling the situation. It can always discard bundles which have expired. . . If it ordinarily is willing to accept custody for bundles, it can cease doing so. If storage resources are available elsewhere in the network, it may be able to make use of them in some way for bundle storage. . . Determining when a node should engage in or cease to engage in custody transfers is a resource allocation and scheduling problem of current research interest.*" This suggests two possible reactions to congestion in DTNs: selectively discarding bundles and/or not accepting custody.

In [82], Bisio *et al.* focus on bundle discarding and on a congestion control scheme based on random early detection (RED) and explicit congestion notification (ECN) mechanisms within interplanetary DTN. Results show an improvement of RED-ECN application by considering a simulated 18 node test bed: the bundle loss rate (BLR) goes

from about 6% to about 2% with the simple application of RED-ECN congestion control, with a fixed routing scheme. This indicates that it is worthwhile investigating the effects of congestion handling techniques from traditional networking in DTNs.

In [83], Seligman *et al.* suggest handling congestion at DTN nodes by migrating stored bundles to neighboring nodes that then accept custody for the migrated bundles. The selection of bundles to migrate may depend on temporal, size, and priority factors. The choice of the new custodian may depend on bandwidth and storage availability. The metric used is the message completion rate (MCR), which is the ratio between the amount of traffic that emerges from the network and the amount of traffic sent into the network (in practice this is the percentage of delivered bundles). Results depend on node storage capacity but, just to report some values, MCR increases from about 45%, without bundle migration, to about 75%, by applying the proposed bundle migration scheme with node storage of 125 KB, from about 63% up to about 88%, with node storage of 250 KB, from about 70% to about 97%, with node storage of 500 KB. Further increases in node capacity provide reduced improvement in MCR.

The relationship between congestion control and routing is shown perhaps most clearly in the above-mentioned contact graph routing (CGR), where the routing scheme is designed so as to avoid congestion. While this may be effective for the kind of deterministic deep-space network envisaged in CGR, it may not apply in all satellite networking scenarios, especially when many sources are generating bundles or when changes to the contact graph occur frequently.

DTN congestion control is a promising research topic and deserves a great deal of attention in future research.

VII. CONCLUSION

In the paper, we have introduced the DTN concept, presented the opportunities offered by DTN, and shown how DTN-based communication may represent an opportunity for satellite networking.

In more detail, the paper underlines the main features of the DTN architecture and of the bundle protocol, such as its overlay function, long-term information storage at intermediate nodes, custody transfer, bundle fragmentation, late binding, and DTN routing peculiarities. After this overview, the paper shows a possible interpretation of DTN as an evolution of TCP-splitting PEP architectures and discusses commonalities and differences of the two solutions. To better assess the potential of the DTN architecture, the paper presents comparative results obtained through emulation within three real application scenarios: GEO with fixed terminals; GEO with mobile terminals; and LEO. Results show that DTN is competitive in the first scenario, the best choice in the second one, and the only choice in the most challenging applications (i.e., data mule) of the third one. The paper then examines security issues in satellite communications and shows why and how DTN can offer a valid solution to them. In particular, after presenting the key features of the bundle security protocol, the paper gives the state of the art of DTN key management, access control, resilience to denial of service attacks, and identity protection. The last part of the paper is devoted to DTN and QoS. In this context, the paper extends the interpretation of DTN as an evolution of a TCP-splitting PEP architecture and shows the DTN architecture as a possible QoS gateway with limited functionalities; presents the tools made available in DTN for QoS management; and suggests DTN present and future research topics such as modeling, routing, and congestion control.

In conclusion, the paper aims to show how DTN, first conceived for deep-space communication and then proposed for terrestrial, maritime, and also underwater sensor networks, can actually play a key role in satellite communications. The ability of DTN to meet the needs of challenged networks in both satellite and terrestrial networking, if proven, would mean that satellite networks might no longer need specific solutions. Through DTN, satellite networks might become just a component of the overall future Internet. This appears to be a goal definitely worth aiming for. ■

Acknowledgment

The authors would like to thank Dr. R. Firrincieli for his contribution to simulation tests.

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, and Z. Shelby, *Performance enhancing proxies intended to mitigate link-related degradations*, Internet RFC 3135, Jun. 2001. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3135.txt

[3] "Technical report on performance enhancing proxies (PEPs) for the European ETSI Broadband Satellite Multimedia (BSM) Working Group," ETSI Rep. TR 102 676. [Online]. Available: http://portal.etsi.org

[4] V. Obanaik, L. Jacob, and A. L. Ananda, "Secure performance enhancing proxy: To ensure end-to-end security and enhance TCP performance over IPv6 wireless networks," in *Proc. Elsevier Comput. Netw. 50*, 2006, pp. 2225–2238.

[5] A. McMahon and S. Farrell, "Delay- and disruption-tolerant networking," *IEEE Internet Comput.*, vol. 13, no. 6, pp. 82–87, Nov./Dec. 2009.

[6] V. Cerf, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, *Delay-tolerant networking architecture*, Internet RFC 4838, Apr. 2007. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4838.txt

[7] K. Scott and S. Burleigh, *Bundle protocol specification*, Internet RFC 5050, Nov. 2007. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5050.txt

[8] W. Ivancic, W. M. Eddy, D. Stewart, L. Wood, J. Northam, and C. Jackson, "Experience with delay-tolerant networking from orbit," *Int. J. Satellite Commun. Netw.*, vol. 28, no. 5–6, pp. 335–351, Sep.–Dec. 2010.

[9] C. Caini, P. Cornice, R. Firrincieli, and D. Lacamera, "A DTN approach to satellite communications," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 5, pp. 820–827, Jun. 2008.

[10] C. Caini, P. Cornice, R. Firrincieli, D. Lacamera, and M. Livini, "TCP, PEP and DTN performance on disruptive satellite channels," in *Proc. IEEE Int. Workshop Satellite Space Commun.*, Siena, Italy, 2009, pp. 371–375.

[11] S. Burleigh, V. Cerf, R. Durst, K. Fall, A. Hooke, K. Scott, and H. Weiss, "The interplanetary internet: A communication infrastructure for Mars exploration," in *Proc. 53rd Int. Astronaut. Fed. Congr./World Space Congr./Acta Astronautica*, Aug. 2003, vol. 53, no. 4–10, pp. 365–373.

[12] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proc. Conf. Appl. Technol. Architectures Protocols For Computer Commun.*, Karlsruhe, Germany, 2003, pp. 27–34.

[13] K. Scott, "Disruption tolerant networking proxies for on-the-move tactical networks," in *Proc. Military Commun. Conf.*, Atlantic City, NJ, 2005, pp. 3226–3231.

[14] P. McDonald, D. Geraghty, I. Humphries, S. Farrell, and V. Cahill, "Sensor networking with delay tolerance," in *Proc. 16th Int. Conf. Comput. Commun. Netw.*, Aug. 2007, pp. 1333–1338.

[15] K. Fall and S. Farrell, "DTN: An architectural retrospective," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 5, pp. 828–836, Jun. 2008.

[16] S. Farrell and V. Cahill, *Delay and Disruption Tolerant Networking*. Norwood, MA: Artech House, 2006, ISBN 1-59693-063-2.

[17] *Internet Research Task Force DTN Reseach Group (DTNRG)*. [Online]. Available: http://www.dtnrg.org/

[18] *CCSDS DTN Working Group*. [Online]. Available: http://cwe.ccsds.org/sis/

[19] E. Davies, *DTN: State of the Art*, N4C Project Deliverable, 2009. [Online]. Available: http://www.n4c.eu/Download/n4c-wp2-012-state-of-the-art-101.pdf

[20] U.S. DARPA, BAA04-13: *Disruption tolerant networking (DTN)*, May 2004.

[21] *Networking for Communication Challenged Communities—N4C, FP7 Project*. [Online]. Available: http://www.n4c.eu/

[22] *7th Framework FIRE Activity*. [Online]. Available: http://cordis.europa.eu/fp7/ict/fire/

[23] S. Lin, D. Costello, and M. Miller, "Automatic-repeat-request error-control schemes," *IEEE Commun. Mag.*, vol. 22, no. 12, pp. 5–17, Dec. 1984.

[24] K. Fall, W. Hong, and S. Madden, "Custody transfer for reliable delivery in delay tolerant networks," Intel Research, Berkeley, CA, Tech. Rep. IRB-TR-03-030, Jul. 2003, pp. 1–6.

[25] M. Demmer and J. Ott, *Delay Tolerant Networking TCP Convergence Layer Protocol*, Internet-Draft, Nov. 2008. [Online]. Available: http://tools.ietf.org/html/draft-irtf-dtnrg-tcp-clayer

[26] H. Kruse, *UDP Convergence Layers for the DTN Bundle and LTP Protocols*, Internet-Draft, Nov. 2008. [Online]. Available: http://tools.ietf.org/html/draft-irtf-dtnrg-udp-clayer

[27] S. Burleigh, *Delay Tolerant Networking LTP Convergence Layer (LTPCL) Adapter*, Internet-Draft, Aug. 2010. [Online]. Available: http://tools.ietf.org/html/draft-burleigh-dtnrg-ltpcl

[28] M. Ramadas, S. Burleigh, and S. Farrell, *Licklider Transmission Protocol—Specification*, Internet RFC 5326, Sep. 2008. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5326.txt

[29] B. Adamson, C. Bormann, M. Handley, and J. Macker, *NACK-Oriented Reliable Multicast (NORM) Transport Protocol*, Internet RFC 5740, Nov. 2009. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5790

[30] E. Kohler, M. Handley, and S. Floyd, *Datagram Congestion Control Protocol (DCCP)*, Internet RFC 4340, Mar. 2006. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4340

[31] *DTN2 Reference Implementation*. [Online]. Available: http://www.dtnrg.org/wiki/Code

[32] *IANA Port Allocations*. [Online]. Available: http://www.iana.org/assignments/port-numbers

[33] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proc. ACM SIGCOMM* Portland, Aug./Sep. 2004, pp. 145–157.

[34] J. Wyatt, S. Burleigh, R. Jones, L. Torgeson, and S. Wissler, "Disruption tolerant networking flight validation experiment on NASA's EPOXI mission," in *Proc. 1st Int. Conf. Adv. Satellite Space Commun.*, Colmar, France, 2009, pp. 187–196.

[35] K. Gifford, A. Jenkins, and S. Kuzminsky, "DTN experiments onboard the International Space Station," presented at the METERON Presentation to CCSDS, Spring Meeting, May 4, 2010. [Online]. Available: http://public.ccsds.org/meetings/2010Spring/Tech/lunchtime/Gifford-DTN%20ISS%20Activities%2004-May-2010.pdf

[36] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: Modeling a three-tier architecture for sparse sensor networks," *Ad Hoc Netw.*, vol. 1, no. 2–3, pp. 215–233, Sep. 2003, DOI: 10.1016/S1570-8705(03)00003-9.

[37] C. Caini and R. Firrincieli, "TCP Hybla: A TCP enhancement for heterogeneous networks," *Int. J. Satellite Commun. Netw.*, vol. 22, pp. 547–566, Sep.–Oct. 2004.

[38] C. Caini, R. Firrincieli, and D. Lacamera, "PEPsal: A performance enhancing proxy for TCP satellite connections," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 22, no. 8, pp. b-9–b-16, Aug. 2007.

[39] C. Caini, R. Firrincieli, and M. Livini, "DTN bundle layer over TCP: Retransmission algorithms in the presence of channel disruptions," *J. Commun.*, vol. 5, no. 2, pp. 106–116, Feb. 2010.

[40] C. Rigano, K. Scott, J. Bush, R. Edell, S. Parikh, and R. Wade, "Mitigating naval network instabilities with disruption tolerant networking," in *Proc. IEEE MILCOM*, San Diego, 2008, pp. 1–7.

[41] S. Kent and K. Seo, *Security Architecture for the Internet Protocol*, Internet RFC 4301, Dec. 2005. [Online]. Available: http://www.rfc-editor.org/rfc/rfc430.txt

[42] T. Dierks and E. Rescorla, *The TLS Protocol Version 1.2*, Internet RFC 5246, Aug. 2008. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5246.txt

[43] H. Cruickshank, R. Mort, and M. Berioli, "Broadband Satellite Multimedia (BSM) security architecture and interworking with performance enhancing proxies," in *Proc. Personal Satellite Services Conf.*, Mar. 2009, pp. 132–142.

[44] 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, Apr. 2005.

[45] H. Cruickshank, P. Pillai, M. Noisternig, and S. Iyengart, *Security Requirements for the Unidirectional Lightweight Encapsulation (ULE) Protocol*, Internet RFC 5458, May 2009. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5458.txt

[46] S. Bellovin, "Probable plaintext cryptanalysis of the IP security protocols," in *Proc. IEEE Symp. Netw. Distrib. Syst. Security*, San Diego, CA, 1997, pp. 52–59.

[47] J. Sing and B. Soh, "A critical analysis of multi-layer IP security protocol," in *Proc. 3rd Int. Conf. Inf. Technol. Appl.*, Sydney, Australia, 2005, pp. 683–688.

[48] Y. Zhang, "A multilayer IP security protocol for TCP performance enhancement in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 4, May 2004.

[49] M. Baugher, L. Dondeti, and F. Lindholm, *Multicast Security (MSEC) Group Key Management Architecture*, Internet RFC 4046, Apr. 2005. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4046.txt

[50] C. Caini, R. Firrincieli, H. Cruickshank, and M. Marchese, "Satellite communications: From PEPs to DTN," in *Proc. 5th Adv. Satellite Mobile Syst. Conf.*, Cagliari, Italy, 2010, pp. 62–67.

[51] W. Ivancic, "Security analysis of DTN architecture and bundle protocol specification for space-based networks," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, 2010, pp. 1–12.

[52] S. Symington, S. Farrell, H. Weiss, and P. Lovell, *Bundle Security Protocol Specification*, Internet RFC 6257, Mar. 2011. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6257.txt

[53] CCSDS, *Security Threats Against Space Missions*, CCSDS Green Book, CCSDS 350.1-G-1, Oct. 2006. [Online]. Available: http://public.ccsds.org/publications/archive/350x1g1.pdf

[54] 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. Int. Workshop Space Satellite Commun.*, Siena, Italy, 2009, pp. 385–389.

[55] A. Seth and S. Keshav, "Practical security for disconnected nodes," in *Proc. 1st IEEE ICNP Workshop Secure Netw. Protocols*, Boston, 2005, pp. 31–36.

[56] *ETSI Technical Specifications for Satellite Networks Multicast Security Architecture and Key Management for the European ETSI Broadband Satellite Multimedia (BSM) Working Group*, ETSI Specifications, ETSI TS 102 466, Dec. 2006. [Online]. Available: http://portal.etsi.org

[57] C. de Laat, G. Gross, L. Gommans, J. Vollbrecht, and D. Spence, *Generic AAA architecture*, Internet RFC 2903, Aug. 2000. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2903.txt

[58] V. Hu, D. Ferraiolo, and D. Kuhn, "Assessment of access control systems," U.S. Nat. Inst. Standards Technol., Interagency Rep. 7316, Sep. 2006.

[59] E. Johnson, G. Ansa, H. Cruickshank, and Z. Sun, "Access control framework for delay/disruption tolerant networks," in *Proc. Int. Conf. Personal Satellite Services*, Rome, Italy, Feb. 4–6, 2010, pp. 249–264.

[60] H. Samuel and W. Zhaung, "Preventing unauthorized messages and achieving end-to-end security in delay tolerant heterogeneous wireless networks," *J. Commun.*, vol. 5, no. 2, pp. 152–163, Feb. 2010.

[61] G. Loukas and G. Oke, "Protection against denial of service attacks: A survey," *Computer J.*, vol. 53, no. 7, pp. 1020–1037, 2010.

[62] S. Farrell, M. Ramadas, and S. Burleigh *Licklider Transmission Protocol—Security Extensions*, Internet RFC 5327, Sep. 2008. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5327.txt

[63] G. Ansa, E. Johnson, H. Cruickshank, and Z. Sun, "Mitigating denial of service attacks in delay-and disruption-tolerant networks," in *Proc. Int. Conf. Personal Satellite Services*, Rome, Italy, 2010, pp. 221–234.

[64] A. Kate, G. Zaverucha, and U. Hengartner, "Anonymity and security in delay tolerant networks," in *Proc. 3rd Int. Conf. Security Privacy*, Nice, France, 2007, pp. 504–513.

[65] S. Symmington, R. Durst, and K. Scott, *Delay-Tolerant Networking Bundle-in-Bundle Encapsulation*, Internet Draft, Aug. 2009.

[66] M. Marchese, *Quality of Service Over Heterogeneous Networks*. Chichester, U.K.: Wiley, 2007.

[67] *EU FP7 Project Sensei*. [Online]. Available: http://www.ict-sensei.org

[68] *EU FP7 Project Eu-Mesh*. [Online]. Available: http://www.eu-mesh.eu/

[69] F. Warthmann, *Delay-Tolerant networks (DTNs), A tutorial*, May 2003. [Online]. Available: http://www.dtnrg.org/docs/tutorials/warthman-1.1.pdf

[70] M. Marchese, "Comparative analysis of interplanetary and pervasive communication," in *Proc. Globecom Conf.*, Miami, FL, 2010, DOI: 10.1109/GLOBECOM.2010.5683499.

[71] A. Balasubramanian, B. Levine, and A. Venkataramani, "DTN Routing as a resource allocation problem," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 373–384, Oct. 2007.

[72] A. Balasubramanian, B. Levine, and A. Venkataramani, "Replication routing in DTNs: A resource allocation approach," *IEEE/ACM Trans. Netw.*, vol. 18, no. 2, pp. 596–609, Apr. 2010.

[73] M. Demmer and K. Fall, "DTLSR: Delay tolerant routing for developing regions," in *Proc. ACM SIGCOMM Workshop Netw. Syst. Develop. Regions*, Kyoto, Japan, 2007, DOI: 10.1145/1326571.1326579.

[74] Wikipedia, *Routing in Delay-Tolerant Networking*. [Online]. Available: http://en.wikipedia.org/wiki/Routing_in_delay-tolerant_networking

[75] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks, Duke Tech. Rep. CS-2000-06, Jul. 2000.

[76] A. Lindgren, A. Doria, and O. Scheln, "Probabilistic routing in intermittently connected networks," in *Proc. 4th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2003, pp. 19–20.

[77] A. Lindgren and A. Doria, *Probabilistic Routing Protocol for Intermittently Connected Networks*, Internet Draft, Aug. 2010. [Online]. Available: http://tools.ietf.org/html/draft-irtf-dtnrg-prophet

[78] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proc. ACM SIGCOMM Workshop Delay-Tolerant Netw.*, 2005, pp. 252–259.

[79] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for vehicle-based disruption-tolerant networks," in *Proc. IEEE INFOCOM*, Barcelona, Spain, DOI: 10.1109/INFOCOM.2006.228.

[80] S. Burleigh, *Contact Graph Routing*, Internet-Draft, Jul. 2010. [Online]. Available: http://tools.ietf.org/html/draft-burleigh-dtnrg-cgr

[81] I. Bisio, T. de Cola, and M. Marchese, "Congestion aware routing strategies for DTN-based interplanetary networks," in *Proc. IEEE Globecom*, New Orleans, LA, 2008, DOI: 10.1109/GLOCOM.2008.ECP.262.

[82] I. Bisio, M. Cello, T. de Cola, and M. Marchese, "Combined congestion control and link selection strategies for delay tolerant interplanetary networks," in *Proc. IEEE Globecom*, Honolulu, HI, Nov.–Dec. 2009, DOI: 10.1109/GLOCOM.2009.5426295.

[83] M. Seligman, K. Fall, and P. Mundur, "Alternative custodians for congestion control in delay tolerant networks," in *Proc. SIGCOMM Workshop*, Pisa, Italy, 2006, pp. 229–236.

ABOUT THE AUTHORS

Carlo Caini (Member, IEEE) received the Dr. Ing. degree (*summa cum laude*) in electronic engineering from the University of Bologna, Bologna, Italy, in 1986.

In 2001, he joined the Department of Electronics, Computer Science, and Systems (DEIS), University of Bologna, where he is currently an Associate Professor in Telecommunications. His main scientific interests are in the field of satellite systems and wireless networks, with a special emphasis in the last years on the study, design, and implementation of transport protocols and architectures for satellite and other "challenged networks," including enhanced versions of TCP, performance enhancing proxies (PEPs), and delay- and disruption-tolerant networking (DTN). He is an author of many international publications on these and other topics. He promoted and supervised the development of many free software programs, such as TCP Hybla, PEPsal, and DTNperf.

Prof. Caini is member of IEEE Communications Society and participates in several international research projects. He was corecipient of the Best Paper Award at the 2009 International Conference on Advances in Satellite and Space Communications (SPACOMM'09) and the 2009 IEEE International Workshop on Satellite and Space Communications (IEEE IWSSC'09) for works on DTN.

Haitham Cruickshank (Member, IEEE) was born in 1959. He received the B.Sc. degree in electrical engineering from the University of Baghdad, Baghdad, Iraq, in 1980, and the M.S. degree in telecommunications, and the Ph.D. degree in control systems from Cranfield Institute of Technology, Cranfield, U.K., in 1995.

He is a Senior Lecturer at the Centre for Communication Systems Research (CCSR), University of Surrey, Guildford, U.K. He has worked there since January 1996 on several European research projects in the ACTS, ESPRIT, TEN-TELECOM, IST, P6, and FP7 programs. His main research interests are network security, satellite network architectures, delay- and disruption-tolerant networking (DTN) security, and quality-of-service (QoS) provisioning. He also teaches the data and internet networking, network security, and satellite communication courses at the University of Surrey. He is the author and coauthor of about 110 papers including 20 refereed journals, over 80 conferences, three books chapters, and four IETF/ETSI standards.

Dr. Cruickshank is a member of the Satellite and Space Communications Committee of the IEEE Communications Society, and is also a Chartered Electrical Engineer and Institution of Electrical Engineers (IEE) corporate member in the United Kingdom.

Stephen Farrell received the joint honors degree in mathematics and computer science from the University College, Dublin, Ireland, in 1986 and the Ph.D. degree "A Delay- and Disruption-Tolerant Transport Protocol" from Trinity College, Dublin, in 2008

He is a Research Fellow in the Department of Computer Science, Trinity College Dublin, Dublin, Ireland, where he teaches and researches on security and delay- and disruption-tolerant networking (DTN). In 2006, he coauthored the first book on the latter topic. He has been involved in Internet standards for more than a decade. He is also a founder of Tolerant Networks Limited, a TCD campus company offering DTN consulting and support.

Dr. Farrell is currently serving as an Internet Engineering Task Force (IETF) security area director and also as Cochair of the Internet Research Task Force (IRTF) Delay Tolerant Networking Research Group.

Mario Marchese (Senior Member, IEEE) was born in Genoa, Italy, in 1967. He received the "Laurea" degree (*cum laude*) from the University of Genoa, Genova, Italy, in 1992, the Qualification as Professional Engineer in April 1992, and the Ph.D. (Italian "Dottorato di Ricerca") degree in telecommunications from the University of Genoa in 1996.

From 1999 to 2004, he worked with the Italian Consortium of Telecommunications (CNIT), the University of Genoa Research Unit, where he was Head of Research. Since February 2005, he has been an Associate Professor at the Department of Communication, Computer and Systems Science (DIST), University of Genoa. He is the founder and technically responsible engineer of Satellite Communications and Networking Laboratory (SCNL), University of Genoa. He is author and coauthor of more than 200 scientific works, including international magazines, international conferences, and book chapters and of the book *Quality of Service over Heterogeneous Networks* (New York: Wiley, 2007). His main research activity concerns: satellite and radio networks, transport layer over satellite and wireless networks, quality of service and data transport over heterogeneous networks, emulation and simulation of telecommunication networks, and satellite components.

Dr. Marchese chaired the IEEE Satellite and Space Communications Technical Committee from 2006 to 2008.