INTERNATIONAL JOURNAL OF SATELLITE COMMUNICATIONS AND NETWORKING
Int. J. Satell. Commun. Network. 2011; **29**:249–268
Published online 14 May 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sat.974

Transport challenges facing a next-generation hybrid satellite Internet

Gorry Fairhurst[1], Arjuna Sathiaseelan[1,*,†], Haitham Cruickshank[2] and Cedric Baudoin[3]

[1]*Electronics Research Group, University of Aberdeen, Aberdeen, U.K.*
[2]*University of Surrey, Surrey, U.K.*
[3]*Thales Alenia Space, Tolouse, France*

SUMMARY

This paper considers the transport layer implications by assuming a position where satellite networks form one integrated component of a hybrid Internet architecture. It reviews the key role of transport protocols in providing a reliable and robust end-to-end Internet service. A history of TCP protocol evolution from a satellite perspective is followed by focussing on the role of protocol-enhancing proxies in satellite systems and how these have impacted the introduction of new Internet transport techniques. Current transport research issues are identified and related to two new architectural approaches to highlight the expected performance benefits and derive the implications on the design of geostationary satellite Internet systems as the network evolves toward a next-generation Internet. Copyright © 2010 John Wiley & Sons, Ltd.

KEY WORDS: satellite; transport protocol; TCP; future Internet

1. INTRODUCTION

The Internet continues to grow, both in the number of connected systems and the number of packets exchanged. Traffic patterns with very different characteristics are becoming common (e.g. Voice over IP (VoIP) v. peer-to-peer download), as a direct consequence of the increasing range of services that are supported. The Internet is vital to the success of today's digital economy by supporting a portfolio of diverse applications (in the broader sense of a 'pool of technologies') including home-produced content, home networks, tele-education, tele-medicine and emergency management [1].

At the network layer, we see IPv4 as the dominant networking technology, although an increasing number of network operators are undertaking some form of IPv6 deployment and much of the technology required to deploy IPv6 in terrestrial environments is operationally ready. IPv6 offers some immediate advantages to enterprise, commercial and governmental environments, but requires changes to the design of the network interface and control protocols, with research and development needed to seamlessly integrate IPv6 with satellite systems. However, IPv6 is just one component needed to evolve away from the limitations of the current Internet and to open the path toward a more pervasive, more accessible, more flexible platform on which can be hosted a wide range of new applications and services.

*Correspondence to: Arjuna Sathiaseelan, Electronics Research Group, University of Aberdeen, Aberdeen, U.K.
†E-mail: arjuna@erg.abdn.ac.uk

Contract/grant sponsor: Satellite Network of Excellence (JA2130)
Contract/grant sponsor: EC 6th Research Framework Programme
Contract/grant sponsor: Thales-Alenia-Space

There is a common understanding of the need for the evolution of the Internet. However, although there are trends, there is no single vision for what will comprise the future Internet. Rather, the concept of the Internet of the future means many things to many people [1]:

- An Internet by and for people/communities (breaking the digital divide, addressing new and common user expectations and needs, facilitating everyday life of people, communities and organizations, breaking the barriers/boundaries between information producer and information consumer, allowing creation of any type of business regardless of size, domain and technology, etc.).
- The Internet as common substrate for knowledge and society, extending the role of digital communication, as a fundamental component of modern society.
- An Internet of things (interconnecting a wide variety of networked objects, devices and systems—much more than computing machines, e.g. ambient networks).
- An Internet of services (intelligent, proactive and not only reactive services).

Over the past decade, there has been significant development of new technology for broadband satellite access and broadband-related services and applications. Many users across the globe now rely on satellite technology to provide a part of their end-to-end Internet service and this is likely to continue for a variety of reasons [1]. Work is expected to continue to harmonize the interfaces and services of satellite systems with both wired and wireless terrestrial networks (e.g. WiMAX, WiFi, WiBro and TETRA). Hybrid satellite/terrestrial wireless will allow satellite to effectively contribute to providing cost-effective broadband to fixed, nomadic and mobile. Satellites are expected to have a specific role as gap-fillers complementing terrestrial ground networks to assure continuity of coverage across the service area.

The rapid advances in the network technology that underpin the future Internet have been accompanied by a significant change in individual user expectations. There is evidence of a growing dependency on the Internet for fundamental services, such as telephony, business transactions and other applications that under-pin the infrastructure of society. This, in turn, has fueled a demand for a more dependable and trustable network and a need to allow rapid composition of new services, in some cases personalized to individual users or enterprises.

We foresee satellite being an integral part of this future Internet, ensuring pervasive access to network capacity and services that communities use and rely upon [2]. Mobility will also become common. We expect that the current edge of the network (e.g. services to people's homes or companies) will often be just one hop in the Internet of the future.

1.1. The role of transport protocols

The Internet protocol suite has been extremely successful in allowing equipment at the network edge to evolve in response to changes in technology and patterns of use. In the present Internet architecture, transport protocols are responsible for many key functions that ultimately guarantee the required reliability of data transmitted across the network and at the same time are responsible for preventing traffic overwhelming the endpoint or the intervening network and importantly in preventing some users being deprived of capacity (starvation) or contribute to network overload (congestion collapse).

There are important changes in the way the transport service is being used. New markets for converged applications/services have already started to emerge: VoIP is now common, Video on Demand is offered in many places, Internet TV is emerging and social networking has been extremely successful. At one extreme, the future Internet is expected to transport bandwidth intensive applications such as tele-immersion and 3DTV and at the other extreme to connect vast numbers of tiny devices integrated into appliances, sensors, actuators and a range of previous independent systems [1]. These changes present new demands on the transport layer.

As the future Internet emerges, transports will need to change to address new network challenges such as: the impact routing scalability, the role of middleware in protocol stacks, the

DOI: 10.1002/sat

emergence of middleboxes within the end-to-end architecture, multi-homing (where systems connect to more than one network to improve service robustness), auto-configuration (where systems detect and configure interfaces to available networks), mobility (where systems move from one network to another to maintain connectivity), the need for a coherent security architecture and more. Future transports will also need to accommodate much greater heterogeneity in network transmission rate and a greater need to support service disruption, as increasing mobile nodes search for the best connectivity.

Another topic that impacts transport, especially multimedia transport, is the potential for convergence of existing satellite systems toward a next-generation network (NGN) infrastructure common with other wireless transmission networks. One of the main objectives of the NGN approach is to provide quantitative end-to-end Quality of Service (QoS) guarantees to individual multimedia services/applications over a multitude of network technologies on the end-to-end path (including both access and core networks). Methods such as the Internet Multimedia System (IMS) can be used to associate specific traffic flows with QoS policies. Where QoS can be explicitly signaled, it can reduce the need for transport protocols to control individual media flows, but it also introduces significant control and management complexity into satellite networks. In many cases QoS parameters are not known *a prior*, detailed classification of flows is not possible (e.g. due to flow encryption) or traffic is aggregated. There is still a need for transport protocols to support flow and congestion control (CC) for applications that do not use paths with tight QoS control.

The paper focuses on transport-related topics, relating this to the emerging vision for a future Internet, where users can seamlessly choose any bearer technology—including satellite—to match the needs of any intended Internet application. Other papers (e.g. [2, 3]) have already or are planned to address issues such as new opportunities and scenarios and the demands these place on spacecraft and ground equipment evolution.

The remainder describes the history of TCP protocol evolution over satellite links (Section 2) and examines methods that have been proposed to improve transport over satellite (Section 3). Section 4 introduces two new transport architectures that could enable satellite systems to take their place as a part of the future Internet. Section 5 concludes the paper.

2. THE TRANSMISSION CONTROL PROTOCOL OVER SATELLITE

Transmission Control Protocol (TCP) [4, 5] provides end-to-end communication across any underlying Internet path [6], offering reliable connection-oriented in-order delivery of data between a pair of Internet endpoints. Since its definition in 1981, TCP has been the dominant transport protocol in the Internet for the decades and continues to be so.

The two basic concepts behind the design of TCP were the ‘end-to-end principle’ [6] (which suggested that complexity within the network should be minimized and that the two end systems should assume responsibility for reliability of data transfer) and ‘fate sharing’ [7] (which dictated that TCP should continue to operate as long as there was any form of connectivity and irrespective of the path taken by packets).

A summary of TCP design goals are:

- It is reliable (recovers loss, preserves sequence).
- It is robust (transfer independent of network path).
- It is adaptive (transmission rate is dependent upon capacity).
- It is conservative (it does not lead to network ‘melt-down’).

2.1. Evolution of TCP

It is only right to question whether TCP is suited for use with applications, such as web 2.0, peer-to-peer, etc., which are quite unlike the applications of the 1970s. If TCP had remained the same since its creation, the answer would probably be ‘No’. However, TCP is not a simple

DOI: 10.1002/sat

protocol, modern TCP employs a family of techniques, which, when used in combination, continue to provide a stable and reliable transfer of packet data across the Internet.

Far from being fixed, TCP has evolved to meet the demands of new applications and an ever-increasing size of Internet. RFC 4614 [5] describes the current state of TCP evolution. TCP in its current form dates from 1988, when CC was added. A key role of modern TCP is therefore to ensure that a TCP session does not consume more than a 'fair share' of the available capacity along the network path over which it operates. The algorithm used to provide CC needs to consider a complex trade-off between utilization of capacity, varying traffic levels and a need for overall stability of the network. Most current vendor implementations of TCP are based on the (New) Reno reference implementation. Many vendors also support the Long Fat Network [8] and Selective Acknowledgment (SACK) [9] options, both can be desirable for satellite use.

Figure 1 depicts the standards that form the current TCP specification.

2.2. Challenges imposed by network heterogeneity

In designing a transport protocol, such as TCP, for use in the general Internet, it is essential to be able to deliver acceptable performance across a wide range of network characteristics. Current Internet paths already present substantial heterogeneity:

- Link rates from <1 kbps to 40 Gbps.
- One-way delay from $<1\,\mu s$ to $\gg 1$ s.
- Multiplexing from 1 flow/path to infinity.
- Supported packet sizes from 68 B to greater than 9 kB.
- Path characteristics that are stable for months or change dynamically over short timescales.

Transport protocols in the future Internet not only need to assure the performance of applications across this increasingly diverse Internet and ensure equitable sharing of available

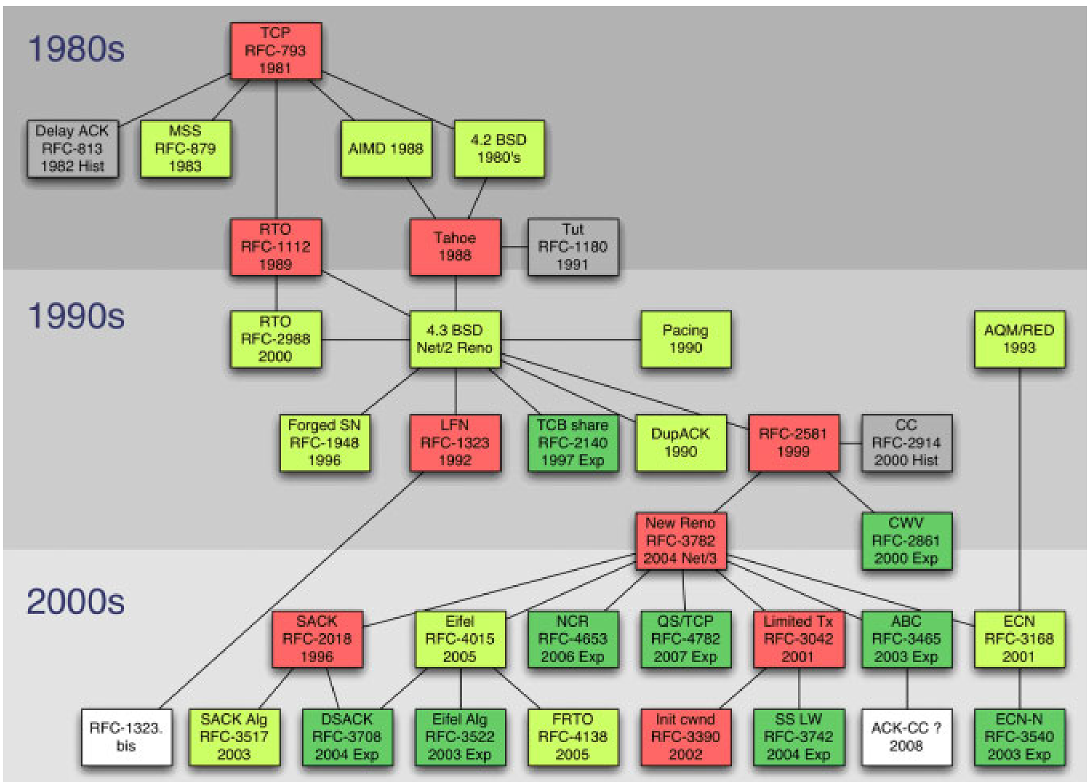


Figure 1. TCP road map.

DOI: 10.1002/sat

network capacity, they must also accommodate abrupt changes in characteristics that can result from changes in the chosen path (due to re-routing, mobility, etc.).

2.3. Challenges imposed by satellite

One of the projects that piloted the Internet approach was a satellite project [10]. Hence, it is no surprise that TCP worked over links with a satellite delay, and satellite links continue to form an integral part of today's Internet infrastructure. Any communication system requiring ubiquitous access to the Internet must consider satellite as a part of the portfolio of bearer technologies needed to support the network. No true worldwide connectivity can be achieved in the absence of a satellite component [2].

However, an increase in the transmission rate, the increasing sophistication of satellite equipment and the increasing demands of new applications have led to a set of performance constraints that are a direct result of the specific characteristics of modern satellite networks [11]. These characteristics are highlighted in the following sections.

2.3.1. The satellite impact on Internet path delay. This paper focuses on satellite networks that use a Geostationary Earth Orbit (GEO) satellite, which introduces a one-way delay of at least 250 ms. Although Low Earth Orbit satellite links can reduce delay, they present other effects such as fluctuations in delay due to handover, inter-satellite links, etc. The GEO satellite delay gives rise to two effects that limit the performance of TCP.

The performance of TCP reduces over paths when the receiver-advertised window is much smaller than the bandwidth delay product of the end-to-end network path [11]. One early problem was related to the limited receiver window size available in early stacks (often 8 kB, limiting throughput to less than 100 kbps). The larger window sizes required to transmit higher speeds over satellite required updates and configuration, but have since become standard features of protocol stacks (e.g. a default window of around 64 kB offers a limit of about 1 Mbps per flow at GEO delay). Still higher rates are achievable using window scaling [8], with a maximum rate faster than any currently deployed satellite link. Window Scaling is widely supported, although not normally enabled by default.

For any long delay path, selecting an appropriate window size is only a part of the problem. TCP also defines a congestion window (cwnd) that constrains the usable window, especially in the first part of any transfer. On reception of each ACK for a transmitted packet, the cwnd is increased by one segment, during the slow start phase and by a fraction of a segment (equivalent to one segment per round trip delay or approximately one segment per second for a satellite link) during the linear increase phase. A long path delay therefore significantly restricts the rate of growth of the cwnd. This is important for satellite users, since most TCP sessions in today's Internet are short-lived (e.g. www transfers, e-mail), and the satellite performance would be significantly improved if applications were designed to take advantage of longer-lived sessions.

In many satellite systems, capacity is a shared and managed resource. Unless appropriate QoS techniques are introduced at the network layer, this can result in significant jitter—which disrupt the timing of multimedia transport and introduce unwanted interactions with transport flow and CC. From a satellite perspective, it is vital that future end-to-end transport mechanisms do not make inappropriate assumptions (such as assuming the measured round trip time (RTT) or RTT variation is linked to the level of congestion).

2.3.2. Satellite channel impairments. In common with other wireless technologies, the satellite link budget determines the signal-to-noise ratio of the receiver signal, and therefore the probability of an error within a packet received over a satellite link. Modern satellite links use power control to accommodate a fall in signal strength jointly with adaptive coding and modulation to ensure a constant bit error ratio (BER). As a result, DVB-RCS provides a BER better than 10^{-7} on the return link and DVB-S2 guarantees a BER equal to 10^{-11} on the forward link, which results in a low packet loss rate. Fade mitigation techniques may also be used by advanced satellite systems to adapt the coding, modulation and the symbol rate to

DOI: 10.1002/sat

mitigate rain fading. The low packet loss rate after applying forward error correction (FEC) ensures that this does not impact the TCP transport performance [11]. However, schemes that adapt the link transmission rate can negatively impact the performance of TCP CC.
In some cases, terminals using a small antenna and/or being mobile may experience a poorer (or varying) BER requiring error correction methods such as FEC or automatic repeat request (ARQ) to prevent end-to-end retransmission of corrupted packets and false trigger of CC [11].

2.3.3. The impact of path changes. Disruption or the momentary suspension of the transmission of data between endpoints is a common feature in many hybrid and wireless network scenarios (e.g. as a result of a change to/from a satellite network segment after re-routing or handover). The effect of the disruption depends on whether the restored connectivity has the same or different path characteristics. The impact is generally a function of the duration of the disruption, compared with the current TCP retransmission timeout (RTO) period.

- For short disruptions (< RTO) standard TCP implicitly assumes that path characteristics remain unchanged by performing Fast Retransmit. If the path characteristics have changed, then TCP may be too aggressive in its behavior.
- For long disruptions (> RTO), standard TCP behaves more conservatively and performs slowstart, re-probing the path characteristics leading to the sub-optimal performance of TCP.

The performance of end-to-end mechanisms reduces over paths that experience gaps in connectivity (e.g. due to a link outages or intermittent connectivity), since a transport endpoint is unable to distinguish between the effects of congestion and loss/handover of a radio link. This can lead to unwanted activation of TCP CC mechanisms, or timeouts, significantly reducing the performance. Current TCP specifications deduce disruption due to lack of acknowledgement. Disruption events could be signaled to the local router or even to the active endpoints using cross-layer signaling, e.g. derived from Internet control message protocol reports [12] or using media-independent handover messages [13]. While some explicit response to disruption events seems attractive for mobility between different network technologies, it is also an area where it has been hard to find safe methods (from a security or CC perspective)—a side effect of 'fate sharing'. Nevertheless in a future Internet, where mobility and multi-homing is common, these effects will need to be addressed.

2.3.4. Implications of radio resource management (RRM). TCP is generally used in a shared Internet environment, where the available throughput to a session only changes as a result of network traffic sharing the same path or when the path through the network changes. Path changes are typically infrequent under normal conditions in the terrestrial Internet.

Layer 2 radio resource management (RRM) techniques (e.g. as defined in DVB-RCS [14]) are required to achieve effective sharing of the cost of satellite links. RRM controls and regulates the share of the radio resource based on a user profile (service level agreement, attenuation condition, etc.) and the requirements for bandwidth or volume. Since this is a dynamic method, it requires capacity requests to be sent over the air interface. The impact of introducing RRM is that the capacity and the delay can change abruptly over time [15].

In the future, such mechanisms will increasingly be adaptive (introducing variable coding and modulation as a response to degradation in channel conditions). Such methods are common in mobile communications and are being discussed for the next generation of DVB-RCS. Although these increase system performance, they can also lead to more significant changes in delay and capacity.

There can be significant interaction between the RRM and TCP CC mechanisms, especially over satellite paths, leading to very slow cwnd growth in the initial part of slow-start and possible RTO expiry [15].

2.3.5. Implications of path asymmetry. Many Internet transports are able to accommodate appreciable network path asymmetry, in that they can achieve good performance, where the transmission rates for download are higher than that for upload. Many satellite access technologies, however, present very significant path asymmetry. Asymmetry is often introduced to reduce the bandwidth of the return link, a key component of the overall system cost. A path with high asymmetry may result in the lower capacity satellite return link 'filling' with acknowledgments [16], causing ACK-clocking to throttle the forward rate of the forward path.

2.4. Improving the TCP performance using standard mechanisms

In 1999, RFC 2488 [11] identified a set of standard mechanisms and extensions that improve the performance of TCP over a satellite network path. The document recommended the use of standard mechanisms such as FEC, large windows, Path MTU discovery and SACK. These mechanisms enable TCP to more effectively utilize the available capacity of the satellite network path. RFC 2760 [17] supplemented this by identifying further research issues and since then new mechanisms have been introduced. Table I summarizes the set of current mechanisms that benefit the TCP performance over satellite. The table identifies which operating systems (Windows (Windows Server 2008 and Windows Vista) [18], Linux 2.6 [19], FreeBSD 8.0 [19], Solaris 10 [19] and Mac OS X [20]) currently support each mechanism.

Most standard TCP mechanisms require the TCP/IP stack to be updated at the sender and receiver, and do not require changes to the routers within the network. Two exceptions are ECN and PMTUD, both of which are widely implemented in routers but have yet to see widespread deployment in operational networks. Middleboxes, such as firewalls can and occasionally are also be impacted by 'new' TCP mechanisms, such as RFC 1323 and [22], motivating some users to disable these functions.

3. TRANSPORTS APPROPRIATE TO HYBRID SATELLITE NETWORKS

There have been many proposals that further tune the performance of the transport layer over a satellite network by using non-standardized mechanisms specifically adapted to a satellite environment. The methods can be divided into window-based methods and rate-based methods. Although such methods were generally not designed to be fair when used in a general Internet environment, they can offer much better performance when used over a path that only includes a satellite link. Table II presents examples of satellite-adapted TCP flavors.

Mechanisms such as those in Table II can improve the performance within a specific environment (such as when used directly over the satellite segment), but most will send more aggressively than standard TCP under some conditions. For example, the correct behavior of TCP Vegas depends on the accurate calculation of a base RTT measurement. If the BaseRTT is too small, then the throughput of the connection will be less than the available capacity; if the BaseRTT is too large, then it will contribute to congestion. However, accurate packet spacing and RTT measurements may not be possible in the general Internet where traffic aggregation, differentiated queuing (QoS) and other effects can lead to unpredictable measurements, affecting the transport behavior.

This trend to mutate TCP and produce new flavors has not only occurred in wireless and satellite networks, some emergent non-standardized flavors of TCP have been deployed in the general Internet. Examples include: TCP BIC [45] and CUBIC [46]—deployed as default transport protocols in various versions of Linux stacks and Compound TCP (C-TCP) [47] in Microsoft Vista. These are currently being evaluated by the IRTF Internet CC Research Group to determine whether they are viable long-term solutions, considering:

- Congestion Control dynamics (or at the very least overload prevention).
- Approximate fairness (or at the very least starvation prevention).

The widespread availability of a range of TCP-like transports is in itself a danger to Internet stability. It could lead to a proliferation of non-standardized CC enhancements, which do not

DOI: 10.1002/sat

Table I. Standard mechanisms that benefit satellite.

Problem	Method	Operating system support	Standards
Link utilization (large bandwidth delay product)	Large Initial Window: Enables a connection to start with a larger cwnd of up to 4 segments saving up to 3 RTTs, improving the performance for short flows over satellite	Enabled by default in Windows, Linux, FreeBSD, Solaris, OS X	RFC 3390 [21]
	Window Scaling: Larger scalable receiver window size allows a better performance over satellite for bulk flows when capacity is available	Enabled by default in Windows, FreeBSD, Linux, OS X; supported in Solaris	RFC 1323 [8] RFC1323.bis [22]
	Appropriate Byte Counting (ABC): ABC reduces the amount of time needed to increase the value of cwnd to an appropriate size in the satellite	Supported in FreeBSD, Linux	RFC 3465 [23]
	TCP Control Block Interdependence (CBI): This allows part of the TCP state to be shared among similar concurrent connections, improving the transient transport performance for short flows over satellite	Supported in Linux	RFC 2140 [24]
	Path MTU Discovery (PMTUD): Allows a sender to determine the maximum packet size for a network path avoiding IP fragmentation. A larger size also increase the rate of cwnd growth, improving the bulk transfer performance for satellite	Enabled by default in Windows, Linux, FreeBSD, Solaris, OS X	RFC 1191 [25]
Loss recovery	Selective ACK (SACK): SACK provides faster recovery after multiple lost segments in a window of data without relying on an RTO, improving the performance after congestion or loss over satellite	Enabled by default in Windows, Linux, FreeBSD, Solaris, OS X	RFC 2018 [26] RFC 2883 [27] RFC 3517 [28]
	Limited Transmit: This improves the speed of recovery for lost segments when a cwnd is small, or when a large number of segments are lost in a single transmission window	Enabled by default in FreeBSD; supported in Windows Linux	RFC3042 [29]
	Forward RTO Recovery (F-RTO): This allows a TCP sender to detect and recover from spurious RTOs caused by delay variation rather than packet loss, improving the performance for satellite	Supported in Windows, Linux, FreeBSD	RFC4015 [30] RFC4138 [31]
Congestion Management	Explicit Congestion Notification (ECN): ECN notifies the sender of congestion without dropping packets. This responds quicker to congestion and eliminates the delay associated with retransmission of dropped packets	Supported in Windows, Linux, FreeBSD, OS X	RFC 3168 [32] RFC 3540 [33]
Disruption	User Timeout Option (UTO): This option allows the user timeout to be increased, enabling a connection to survive extended periods without end-to-end connectivity	Supported in FreeBSD	RFC5482 [34]

Table I. *Continued*

Problem	Method	Operating system support	Standards
Asymmetry	Path MTU Discovery (PMTUD): A sender change enables a larger segment size, reducing the ACK rate	See above	RFC1191 [35] RFC4821 [36]
	ACK Modification and ACK Congestion Control: A modified delayed ACK algorithm allows a TCP receiver to reduce the ACK rate, improving the performance with congested return links	No current support in standard OS	RFC3449 [16, 37]

Table II. Examples of satellite-specific TCP flavours.

Type	Method	Summary
Window based	TCP Peach [38]	Uses low priority dummy segments to probe the network. Replaces slow start with sudden start: enabling sender to start sending a cwnd of segments one RTT after connection establishment using probe results. Replaces fast recovery with rapid recovery: uses a heuristic to allow sender to distinguish wireless errors from congestion loss
	TCP Westwood [39]	Proposes mechanisms to set slowstart threshold (ssthresh) and congestion window (cwnd) values intended to be consistent with the network capacity measured at the time of congestion. Updates faster recovery to avoid conservative reduction of cwnd after congestion using an end-to-end estimation of available capacity
Rate based	TCP Vegas [40]	Determines the sending rate by measuring packet delay, rather than packet loss. Detects congestion early by measuring an accurate RTT and checking for increases in the RTT
	TCP Hybla [41]	Provides a mechanism to enable long RTT connections to achieve the same transmission rate as a small RTT reference connection. It relies on SACK and uses packet spacing and timestamps to estimate capacity
	SCPS-TP [42, 43]	A space communications extension to TCP that used timestamps and window scaling. It can optionally use TCP Vegas for CC, provide periodic acknowledgments based on RTT and can provide selective negative acknowledgement (SNACK) to determine the cause of packet loss
	TCP Noordwijk [44]	A transport designed to operate over DVB-RCS and tailored to transmit short, but frequent, bursts of data. It seeks to adapt a burst transmission rate to capacity, while avoiding congestion collapse and seeks to not limit the throughput of competing TCP connections

fairly share capacity, lead to adverse interactions among different mechanisms or even worse seek to outperform each other. The use of non-standardized mechanisms may therefore be a threat to the stability of the future Internet.

3.1. Satellite performance enhancing proxy methods

Satellite systems have traditionally deployed protocol performance-enhancing proxies (PEPs) [48] to overcome constraints to Internet applications imposed by satellite networks. PEPs have primarily addressed the TCP performance issues for web-based applications, but can be optimized for any application. The term 'PEP' covers a wide range of protocol mechanisms and tuning methods performed at different protocol layers. Although there have been attempts to define a common satellite PEP architecture [49], as yet there is no standard PEP.

TCP-PEPs can release endpoint transport protocols from window size limitations as a result of local (of default) TCP/IP stack configuration, but the largest improvement is usually a result of overcoming the constraints of TCP Slow Start for short TCP connections. Other transport/network functions include:

- *ACK reduction*: Reducing unnecessary acknowledgements to improve efficiency or support asymmetry of path [16].
- *Flow control*: Providing feedback to control traffic flow in the satellite system and/or spoofing transport acknowledgments to control a sender.
- *Error recovery*: Link recovery of corrupted or lost packets [43] or spoofing of transport protocol functions to accelerate retransmission.
- *Traffic prioritization*: Providing cross-layer optimizations between session and radio-resource management (BoD) and QoS functions at lower layers [48].
- *Connection establishment spoofing*: Intelligently spoofing the TCP three-way handshake to speed up the establishment of a connection.
- *Header and/or payload compression*: Reducing protocol information or modifying protocols to match specific characteristics of a satellite channel.
- *Redirection*: Detecting application protocols and redirecting requests to the IP address of a content server, cache or application performance accelerator (see Section 3.2).
- *DNS acceleration*: Caching and other techniques to minimize delay, and eliminate DNS queries that resolve to a redirected content server [50].

It is common for deployed networks to combine more than one PEP method, and usually this includes both transport optimization and application-layer acceleration (e.g. by incorporating support for pre-fetching web content.

Most TCP PEPs use an architecture that splits the end-to-end TCP connection into multiple TCP connections, each independently using a non-standard transport session for the satellite segment, adapted to the characteristics of the link. PEPs using the split-TCP approach can be divided into two categories: Distributed PEPs, where the PEP client and server are located at each end of a satellite link and both end-points use standard mechanisms and Integrated PEPs, with only one PEP entity residing at the satellite gateway [49] using standard mechanisms to the endpoint beyond the gateway and a variant of TCP for the endpoint connected to the remote satellite terminal. The architecture of various PEPs is described in [50].

The Interoperable PEP (I-PEP) functional architecture assumes a split-distributed approach with the I-PEP server and a client both capable of supporting the I-PEP protocol (Figure 2). The transport protocol on the satellite segment is based on Space Communications Protocol Specification (SCPS-TP) [43]. It also adds a session protocol to control other enhancements above the transport layer. The Linux PEPSAL architecture [51] also uses a split approach, which maps standard TCP sessions to satellite networks using TCP Hybla [41, 50].

The key advantage of using integrated PEPs is that users can effectively use a satellite network with a PEP irrespective of their client platform. Many PEPs are integrated in satellite terminals were they can also introduce subtle cross-layer mechanisms coupled to the satellite lower layers. This design can lead to a tangle of complexity and complicated 'corner cases' such

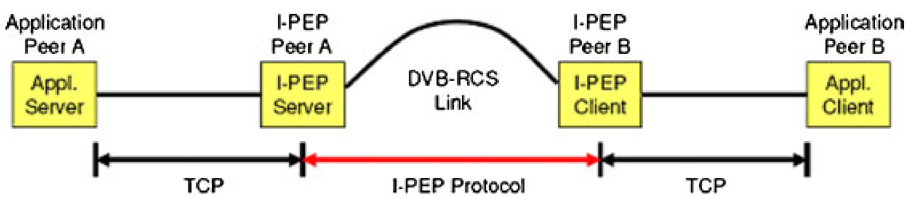


Figure 2. The split-TCP approach using I-PEP over a DVB-RCS link.

DOI: 10.1002/sat

as how to handle time-wait state in TCP, or how to provide flow-control for PEPs. However, the key advantage is that users can effectively use a satellite network with a PEP irrespective of their client platform. In addition, it does not require a user to use non-standard, additional hardware, or software to effectively use a satellite network.

One drawback of a PEP is that it cannot offer full benefit when IPsec is used in tunnel mode to provide a virtual private network service via satellite. In this case, some PEP functions can be provided remotely at the tunnel endpoints. However, it requires tunnel endpoints to be specifically configured to support specific types of satellite service, introducing more configuration complexity and requiring an awareness that the tunnel path includes a satellite segment.

A remote PEP (positioned at the ingress/egress of a tunnel) also cannot access satellite terminal state, and hence are typically unable to provide cross-layer interactions (e.g. to avoid congestion in the medium access control queues of the satellite terminals). Finally, some methods (error-recovery, lower layer header compression, etc.) cannot be performed remotely, although to some extent these drawbacks may be addressed by additionally using another PEP at the satellite terminal, and potentially defining new control information flows to the satellite PEP. These additional flows may also need to be secured, adding to the complexity of the approach.

3.2. Application protocol acceleration

Many of the most significant performance problems have their origin in the way in which applications use the transport service. This is a natural problem with a general purpose Internet where transports seek to support a wide range of application behaviors.

A long-term solution to the problems of TCP slow-start for short-lived connections is to modify applications to use fewer longer-lived connections in place of many short-lived connections. The primary protocol used for web browsing is http, which supports this style of operation (called ‘persistent http’, part of ‘http 1.1’). Adopting this as the default would have a very significant benefit to the satellite community. However, the incentive to change is not necessarily high, since this can also increase resources consumed within a server.

Another approach is to implement standard Internet methods to accelerate the application performance. As part of protocol enhancement, a transparent web proxy placed at the edge of a satellite network can map requests from a remote client to a session that uses a satellite-tuned transport protocol over the satellite network. A typical application proxy intercepts a DNS request, reassigning the IP address to an accelerator (or cache) that then assumes the role of the web server. Such proxies can also employ a range of methods to accelerate delivery, including use of caching, pre-fetching of related content and compression/recoding techniques to optimize transmission efficiency. Application acceleration methods are widely used in the Internet, especially for web traffic, and since these functions take place above the transport layer (in contrast to many methods in Section 3.1) they raise fewer issues than introducing transport enhancements.

However, it would be naïve to think that all traffic is Web content and that this content consists only of web traffic with simple web pages that can easily be redirected to proxy servers. Much current Internet content is marked as not cacheable, is dynamically generated or is protected by security protocols. This trend is increasing, with the emergence of mash-ups, increasing use of web-services, and real-time content. The http protocol is increasingly being used as a front-end or transport for applications very different to web browsing.

3.3. Transport challenges arising from the current Internet architecture

The end-to-end principle [6] identifies Internet transport functions as being between a pair of hosts, each bound to a globally unique IP address, which communicate using a locally meaningful transport port at each end host. This design principle remained relevant and powerful during much of the evolution of the Internet. However, in recent years this architecture has been progressively eroded [48] with the emergence of network address translation (NAT), firewalls and other middleboxes (in which we also now include PEPs in a satellite/wireless context).

Such middleboxes often intend to be ‘transparent’ to common applications, but operate by intercepting transport layer packets, interpreting higher-layer protocol information and

DOI: 10.1002/sat

applying heuristics to change protocol parameters or algorithms with the aim of improving the user performance. However, the current architecture makes it impossible to be transparent for all emergent applications. For example, NATs that offer an effective web-browsing experience can require significant 'work' to allow peer-to-peer communications (VoIP, P2P, etc.).

Not only is it now hard to deploy new applications, and extremely hard to deploy new network and transport mechanisms, many users now find that end hosts are often unable to connect to one another, even when security policies would otherwise allow such connections. This is a huge impediment to the evolution of the future Internet.

3.3.1. Challenges to the use of PEP methods in the future Internet. This section explores some of the middlebox issues from a satellite perspective, realizing that if satellite systems are to achieve their goal of integration in the future Internet, they will need to re-examine the traditional role of PEPs and consider the ways in which PEPs can evolve to support the challenges of the future Internet. Some specific challenges are:

1. *IPv6*: The introduction of IPv6 will require PEP vendors to update their systems, if they have not already done so. This could be as simple as changing the layer 3 service. However, IPv6 not only changes the address format, it also reintroduces the concept of network header extensions that can allow interaction both end-to-end and between-network entities along an Internet path. Middleboxes such as PEPs need to identify the presence of such extensions and where appropriate participate in the related protocol.
2. *Ossification of transport mechanisms*: Transport protocols continue to improve the performance of standard protocol stacks. When a new transport mechanism is introduced into standard stacks, it may not initially be compatible with a PEP mechanism. For example, a PEP may prevent endpoints successfully negotiating a new TCP option (such as UTO [48], Quick-Start [52], limited transmit [29]), even if this could offer significant performance gains in a satellite environment or would be needed to enable a new application. As such PEPs can contribute to ossification of the protocol stack, slowing the rate of deployment of end-to-end solutions.
3. *Mobility handoff*: Roaming among different communication networks will be common in the future Internet, with opportunities for combining service elements from a range of technologies to innovate new services and greater user choice. A key disadvantage of PEPs is that end hosts are typically unaware of the presence of a PEP along the Internet path. When mobility is supported in a heterogeneous environment, a mobile node may need to utilize a range of bearer networks to provide continuity of service. As mobility bindings change, systems using PEPs may need to also handover the PEP context associated with a flow. Work is in progress to integrate mobility support between different layer 2 technologies [13], although currently no specifications are available for handover to satellite networks. Such methods could provide knowledge of changes in path characteristics and QoS parameters to a mobile node. Layer 2 or Layer 3 mobility introduces transport challenges also. It is non-trivial to introduce a PEP into an existing path—if, for instance, a mobile node starts to tunnel over a satellite bearer to provide a remote service. Changes of network path can introduce momentary loss, a change of path RTT, a change of available capacity, etc.
4. *Multi-homing*: Simultaneous use of more than one interface to provides greater resilience to failure, to select the best network for a particular service or allow simultaneous transfer over multiple paths (e.g. recent work on multipath TCP) is especially important for corporate scenarios. This raises challenges to transport, in general, but the different path characteristics of satellite need to be accommodated and new methods found when working with PEPs.
5. *Multiple middleboxes*: There is a possibility that an end-to-end path will encounter not just a single PEP, but several other middleboxes (PEPs, QoS enforcers, etc.). There is a danger that different middleboxes will perform different piecewise optimizations, resulting in unwanted interactions and a reduction in the overall performance and/or

DOI: 10.1002/sat

reliability. Such problems are expected to become more common in the future Internet as applications run in a hybrid network environment that combine a range of different bearer technologies.

6. *Increased use of security mechanisms*: Most transport PEPs rely upon modification of the protocols/payload data to achieve their goals. This requires access to protocol header fields that are usually protected by security protocols. In general, the use of network-layer encryption prevents the use of TCP-PEP on the encrypted packets. Various studies have explored options to allow a trusted PEP to access encrypted flows and provide enhancement, by updating IPsec. However, currently suggested schemes either reduce the security or introduce significant additional complexity [50]. Security applied at or above the transport layer payload does not impact the transport operation of a split-PEP, but may still prevent application-level acceleration.
7. *Control-plane interactions*: To effectively optimize the performance, PEPs would seek access to, and in some cases modify the control-plane information of higher-layer protocols (such as session initiation protocol, real-time streaming protocol, peer-to-peer etc.). Since these protocols are primarily addressing higher layer functions (e.g. session control), the information may in future be authenticated or encrypted and may change substantially as technology or applications change. Systems need to avoid unnecessary dependencies: It is desirable that the service offered by a satellite path does not degrade when a new speech codec or session syntax is used in a next-generation application. The key issue here is a lack of a widely deployed end to middlebox control function.
8. *Evolution of network and transport architecture*: Middleboxes that tune behavior to particular network or transport protocols, not only result in ossification, they can impede the evolution of the transport and network layers, slowing or preventing the introduction of new protocols. The future Internet needs to allow change. As the network moves from the relatively fixed TCP/IP stack of the 1990s we see the need to allow deployment of new protocols. PEPs will need to tailor their support, based on how best to optimize specific stacks and applications, and be able to identify which sessions do not require/benefit from the PEP support. We see no reason why transport protocols will not continue to evolve, and this includes utilizing new approaches such as address-sharing and pre-congestion notification, and other features introduced at the network layer.

4. TRANSPORT IN THE FUTURE INTERNET

After many years of speculation, it seems that the Internet network layer is finally poised to change. IPv4 address exhaustion will drive operators either to adopt wider use of NAT or to finally make a transition to IPv6. The problem will only be exacerbated with the emerging need for IPv4–IPv6 translation, and the increased use of multiple network technologies together with mobility. These developments are not in response to the requirements of satellite systems, but they will nonetheless impact the way in which satellite systems need to be designed, and the performance perceived by the user.

Some applications do not want (or require) TCP's notion of reliability or strict packet ordering, e.g. applications that generally prefer time delivery of data. The IETF has defined a range of transport protocols (Table III).

The use of middleboxes (including NATs and PEPs) can be an obstacle to direct communication between hosts, to share files, access remote storage, upload and download multimedia content, etc. Many applications (adaptive multimedia, web and file sharing, background down-load and peer-to-peer) do not use TCP, yet need to work across satellite networks. In future, these are likely to be joined by a range of new and diverse applications.

Many multimedia services use UDP in preference to TCP. Applications that do not use QoS can take advantage of new transport methods, such as DCCP, which can efficiently share network capacity and provide CC. Work is needed to develop these advanced transport methods and ensure their widespread adoption across all network technologies, and specifically to ensure the favorable performance and efficiency over satellite.

DOI: 10.1002/sat

Table III. Internet standard transport protocols.

Protocol	Mode	Reliability	Notes	Base Reference
TCP	Connection-oriented	Reliable byte stream	Head of Line Blocking	RFC 793 [4]
SCTP	Connection-oriented	Reliable messages	Multi-homed Little NAT support	RFC 4960 [53]
SCTP-PR	Connection-oriented	Partially reliable messages	Same as SCTP	RFC 3758 [54]
DCCP	Connection-oriented	Non-reliable datagrams	Plugable CC Little NAT support	RFC 4340 [55]
UDP	Connectionless	Non-reliable datagrams	No CC	RFC 768 [56] RFC 5405 [57]
UDP-Lite	Connectionless	Non-reliable error tolerant	No CC Little NAT support	RFC 3828 [58] RFC 5405 [57]
RMT	Multicast	Reliable objects	A modular Building Block Framework Plugable CC	RFC 2357 [59, 60]

4.1. A transport architecture for the future Internet

Section 3.3.1 highlighted issues arising from the ossification of the Internet—in the transport context, impacting the ability to deploy mechanisms and new protocols as a result of the increasing presence of middleboxes. PEPs were also seen as having side-effects that can slow the evolution of Internet transport services.

Although the technical difficulties have been recognized for some time by the Internet community, there is now growing awareness at a governmental level that things will need to change. This has lead to many people contemplating the shape of the future Internet. From a transport perspective, this has lead some people to re-examine the architecture of the Internet protocol stack, and others exploring radically different protocol mechanisms that could be introduced into next-generation protocols. Initiatives, such as the Global Environment for Network Innovations from the US National Science Foundation, have been joined by research program across the world such as the European Commission's Framework program (e.g. the trilogy project (www.trilogy-project.org) and 4ward project (www.4ward-project.eu/)). Initial approaches focused on either a 'clean-slate' design—replacing the entire infrastructure with something much more flexible and maintainable, to something more 'evolutionary'—seeking to provide stepwise deployment of significant but incremental updates. Many people now recognize that the best approach could be a combination of these two.

The following sections examine alternate approaches to the design of Internet stacks that could offer benefits of the end-to-end model, while offering good performance when part of the Internet path is provided by a satellite link. We do not further discuss the evolutionary approach of deploying QoS to solve these problems since this is not a method that has been shown to scale to the global Internet.

4.2. A modular transport architecture

An Internet transport architecture defines a set of different components: transport semantics, endpoint addressing (e.g. port numbers), flow regulation (e.g. CC) and reliability functions (e.g. retransmission). Although these functions are distributed among a number of different specifications, the approach adopted by TCP integrates all these functions in a single protocol. More modern transports, such as the Datagram CC Protocol, DCCP [55] and the Reliable Multicast Transport, RMT [59] instead adopted a modular approach, dividing the layer into base functions and modular extensions. In the case of DCCP, a range of CC functions are provided—suited to different types of applications requirements, and it is envisaged that different CC functions will emerge as DCCP receives wider deployment.

The modular architecture could also be extended to other transport protocols. This could separate the coupling between endpoint addressing and flow regulation, allowing much easier transport evolution in the future Internet. For example, the deployment of many new transports (DCCP, UDP-Lite, SCTP, etc.) has been hampered by the need to update middleboxes to interpret the transport headers and extract the endpoint addressing and derive connection state. This design could also ease the introduction of new CC and reliability mechanisms.

A new transport design could divide protocol functions into hop-by-hop or end-to-end [61]. This approach allows a particular network to modify the hop-by-hop processing (e.g. adapting CC for a satellite case), but preserves important end-to-end functions. An enhancement called 'SaNTA' [62] introduced transport proxies inserted at the edge of a bearer network, making PEP part of the end-to-end architecture.

A new architecture that builds on previous work has been suggested as the basis for an alternative transport protocol that could in future replace TCP for many applications [63]. This architecture splits the transport layer into three separate sub-layers (Figure 3):

- *Endpoint layer*: This layer is responsible for multiplexing and flow identification and could be seen as a logical extension to the network layer that provides intra-host addressing and routing using port numbers. Such information could be of use by middleboxes.
- *Flow regulation layer*: This layer manages the performance of a flow between a pair of endpoints, and includes CC functions, multi-homing and multi-path functions, mobility functions, needed to regulate the rate of transmission in a shared Internet. Since this layer is distinct from the transport layer, in the future this layer could interact with specific middleboxes, such as PEP devices that modify CC behavior for specific types of links.
- *Transport layer*: This layer will provide functions needed to interface to the transport service user. Such functions include end-to-end flow control, the organization of data into reliable byte streams, reliable datagrams, media frames, multi-streams, or structured streams, and the handling of end-to-end network signaling.

This architecture permits an Internet path to be split into segments using Flow Middleboxes, in a way that resembles the split behavior of PEPs, where the Flow layer for each segment may use a CC method that is suited to the characteristics of the segment without impacting end-to-end transport semantics. This would be useful in a heterogeneous environment, e.g. a hybrid network that includes a satellite path (Figure 4). If end-to-end transport security functions were placed in the upper layer, then this could operate over the flow-layer without adversely impacting Flow Middeboxes.

This new transport architecture could evolve to address issues such as NAT traversal, buffering in flow middleboxes, APIs and interfaces between the new layers and cross-layer

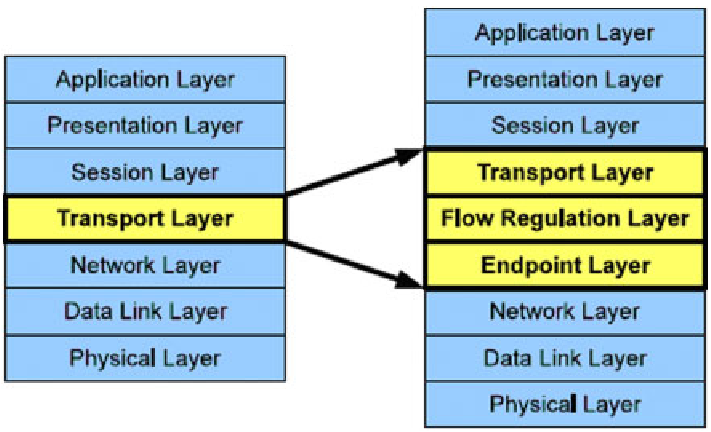


Figure 3. One proposal for reorganizing the transport layer [63].

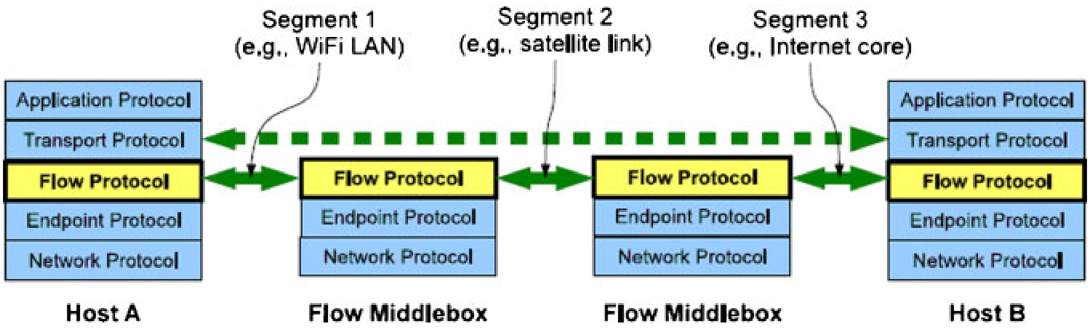


Figure 4. Example of path segmentation [63].

dependencies [63]. Deploying the architecture would require changes to end hosts and middleboxes and therefore significant agreement on the new way forward.

4.3. An adaptive transport architecture

The modular approach seeks a new design for the protocol stack. The future Internet may also present an opportunity to change the way protocol mechanisms work within the transport layer, and potentially introduce a way in which the transport layer can interact more explicitly with the network layer. Such methods could allow transports such as TCP to perform well in a heterogeneous environment. We propose a transport layer architecture that brings together a set of transport mechanisms to coordinate the transport connections in a host (Figure 5). This could be implemented with current mechanisms, without needing major architectural changes to the Internet, although new mechanisms could easily be introduced as they mature.

In this architecture, an individual transport connection, e.g. TCP, DCCP or the SCTP session, collects information about the current path it uses (as TCP holds connection state in a TCP control block). The connection-level information also contributes to a longer-term perspective of the path characteristics collected over all sessions. This information would be used to allow the transport protocol to choose appropriate mechanisms and initial parameters suited to the recent characteristics of the path. This would benefit transport protocols that need to operate over heterogeneous paths, for example, a connection across a satellite path may then use optimized initial parameters appropriate to the satellite path characteristics, without requiring PEPs to be placed in wireless/satellite segments. In future, this path history information could be shared between different hosts that perceive themselves to be behind the same bottleneck.

This architecture introduces two levels of state within a host:

- Network path state maintains information about the recent history of the network path(s) used by the host, e.g. congestion levels (methods such as Re-ECN can inform the amount of congestion being caused), reordering, existence of multiple paths, path MTU, RTT, etc.
- Connection state maintains per connection state for transport parameters determined by observing the performance of each session, e.g. ssthresh, cwnd, number of duplicate acknowledgements, etc. for a TCP or SCTP session and send rate, packet loss rate, timer history, etc. for a DCCP or RMT session.

In its simplest form, the new architecture only caches shared state, as in TCP control block interdependence, path MTU discovery, multipath TCP etc. However, we suggest the shared network path information could be used to make a tentative selection of appropriate protocol mechanisms and parameters (e.g. if a network path exhibits packet reordering, the TCP could be chosen to use the duplicate selective acknowledgement (DSACK) option with a higher duplicate threshold parameter based on the recent history).

The chosen parameters need to be subsequently confirmed by observing the progress of the connection [64]. If the chosen value is incorrect, then the mechanism reverts to a safe approach,

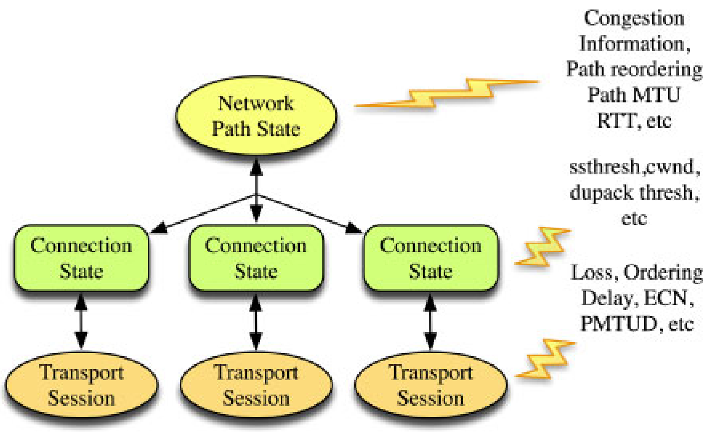


Figure 5. Integrating network awareness into the transport architecture.

and ensures that the network state is updated to prevent other new TCP connections from being affected.

This has the potential to significantly benefit paths that include satellite networks. An endpoint that successfully used a recent connection that operated at a specific rate using a satellite path would have stored path state that would allow the endpoint to start a subsequent connection at a higher rate than normal (reasonably assume that the capacity may still be available). It may continue at this higher rate for a short period, or until it receives any contrary feedback.

Key benefits are:

- It encourages the use of a single flavor of standard TCP together with standard mechanisms, but has the flexibility to adapt to a heterogeneous hybrid environment. It provides an evolutionary path that facilitates the addition of new standard methods that can enhance the performance of TCP.
- It is not limited to TCP sessions, and instead would allow shared state between any transport protocol, easing the deployment of new protocols.
- It provides a common framework that eases the introduction of new network-layer mechanisms, such as support for mobility and multi-homing and common approaches to transport security.

4.4. The role of security

Security issues were largely ignored in the design of the original Internet architecture. Satellite networks (professional and residential) have deployed middlebox networking devices (including PEPs, as described in Section 3.1), which have proved to be incompatible with many security solutions (including IPSec). The future Internet needs to seriously consider security issues early in the design process.

The model for security in the future Internet needs to address the challenges of the evolving network architecture, and in particular the increased use of wireless networks (including satellite), and roaming across networks. Security needs to be provided as a transport service (including encryption and authentication) to secure the communication between endpoints. To improve the robustness of the Internet, security mechanisms are required at lower layers, for instance, to protect the transport endpoints from being impacted by mis-delivered datagrams or from protocol attacks on the transport functions.

Transport security can also help to protect the network from many Denial of Service attacks targeted at network infrastructure, including middleboxes. Although the increased demand for security in the network and lower layers may also prevent the simple use of middlebox approaches.

Security models must provide acceptable security for users of the transport service, protection of the network, and address a raft of wider security concerns.

DOI: 10.1002/sat

5. CONCLUSION

This paper examines the role of transport protocols when used over Internet satellite systems. TCP is the most widely used transport protocol, but also one that has faced challenges when used in a satellite context. The paper reviews the history of TCP protocol evolution over satellite links. It explores the interactions between standard Internet transport mechanisms and satellite networks and how the particular characteristics of satellite links have lead to the widespread adoption of a range of TCP protocol-enhancing proxy (PEP) designs. In particular, it examines the different PEP mechanisms and whether such mechanisms are appropriate in a future Internet. This has revealed a set of challenges to the PEP approach when used in the evolving Internet.

The paper also examines proposals for a new generation of transport protocols. It suggests that a common modular and adaptive transport architecture currently being proposed for the general Internet also has the potential to enable richer interaction between the end hosts and the network. If widely adopted, this architecture could eliminate the need for PEPs in future IP-based satellite networks, allowing satellite systems to become one integrated component of a future Internet architecture that offers reliable, robust and pervasive networking and access to a wide range of network services.

A key challenge to the satellite community is to ensure that the next generation of Internet protocols are not adversely impacted by the effects of satellite delay or the transient variations in the performance resulting from BoD, satellite handover, etc. Common protocols that work over all types of network are vital as the Internet becomes the dominant network-layer, and satellite systems are increasingly integrated with other networking technologies to form a common hybrid bearer for the future Internet.

ACKNOWLEDGEMENTS

This work has been funded by the Satellite Network of Excellence (JA2130). The European Satellite Communications NoE (SatNEx II, IST-27393) is funded within the EC 6th Research Framework Programme. The editors of this paper gratefully acknowledge the contributions, feedback and encouragement of the many partners in the SatNEx JA21390, FT5 activity. Arjuna received funding from Thales-Alenia-Space for the study of advanced Transport mechanisms.

REFERENCES

1. Future Internet: The Cross-ETP Vision Document. *Version 1*, 8 January 2009. Available on-line at: http://www.future-internet.eu/publications/papers-reports.html.
2. Fairhurst G, Collini-Nocker B, Caviglione L. FIRST: Future Internet—a role for satellite technology. *IEEE International Conference on Space and Satellite Communications* (*IWSSC*), Toulouse, 2008.
3. *ISI Future Internet Position Paper*. Available on-line at: http://www.isi-initiative.eu.org/.
4. Postel J. Transmission control protocol. *IETF RFC 793*, September 1981.
5. Duke M, Braden R, Eddy W, Blanton E. A roadmap for transmission control protocol (TCP) specification documents. *IETF RFC 4614*, September 2006.
6. Saltzer J, Reed D, Clark DD. End-to-end arguments in system design. *ACM Transactions on Computer Systems* 1984; **2**(4):277–288.
7. Clark D. Design Philosophy of the DARPA Internet Protocols. *ACM Computer Communication Review* 1995; **25**(1):102–111.
8. Jacobson V, Braden R, Borman D. TCP extensions for high performance. *IETF RFC 1323*, May 1992.
9. Blanton E, Allman M, Fall K, Wang L. A conservative selective acknowledgment (SACK)-based loss recovery algorithm for TCP. *IETF RFC 3517*, April 2003.
10. Seo K, Crowcroft J, Spilling P, Laws J, Leddy J. Distributed testing and measurement across the Atlantic packet satellite network (SATNET). *ACM Computer Communication Review* 1988; **18**(4):235–246.
11. Allman M *et al*. Enhancing TCP over satellite channels using standard mechanisms. *IETF RFC 2488*, 1999.
12. Zimmermann A, Hannemann A. Make TCP more Robust to Long Connectivity Disruptions, IETF Work in Progress. *TCPM working group draft-zimmermann-tcp-lcd-00*, January 2009.
13. Yang O, Choi S, Choi J, Park J, Kim H. A handover framework for seamless service support between wired and wireless networks. *Advanced Communication Technology*, 2006.
14. ETSI EN 301 790. *DVB; Interaction Channel for Satellite Distribution Systems*, ETSI, 2003.
15. Sooriyabandara M, Fairhurst G. Dynamics of TCP over BoD satellite networks. *International Journal of Satellite Communications and Networking* 2003; **21**(4–5):427–449.

DOI: 10.1002/sat

16. Balakrishnan H, Padmanabhan V, Fairhurst G, Sooriyabandara M. TCP performance implications of network path asymmetry, BCP 69. *IETF RFC 3449*, December 2002.
17. Allman M *et al*. Ongoing TCP research related to satellites. *IETF RFC 2760*, February 2000.
18. New Networking Features in Windows Server 2008 and Windows Vista. Available from: http://technet.microsoft.com/en-us/library/bb726965.aspx.
19. Enabling High Performance Data Transfers. Available from: http://www.psc.edu/networking/projects/tcptune/.
20. MAC OS X Network Tuning Guide Revisited. Available from: http://www.macgeekery.com/tips/configuration/mac_os_x_network_tuning_guide_revisited.
21. Allman M, Floyd S, Partridge C. Increasing TCP's initial window. *IETF RFC 3390*, October 2002.
22. Jacobson V, Braden R, Borman D. TCP extensions for high performance. IETF work in progress, TCPM working group. *draft-ietf-tcpm-1323bis-01*, March 2009.
23. Allman M. TCP congestion control with appropriate byte counting. *IETF RFC 3645*, February 2009.
24. Touch J. TCP control block interdependence. *IETF RFC 2140*, April 1997.
25. Mogul J, Deering S. Path MTU discovery. *IETF RFC 1191*, November 1990.
26. Mathis M, Mahdavi J, Floyd S, Romanow A. TCP selective acknowledgment options. *IETF RFC 2018*, October 1996.
27. Floyd S, Mahdavi J, Mathis M, Podolsky M. an extension to the selective acknowledgement (SACK) option for TCP. *IETF RFC 2883*, July 2000.
28. Blanton E, Allman M, Fall K, Wang L. A conservative selective acknowledgment (SACK)-based loss recovery algorithm for TCP. *IETF RFC 3517*, April 2003.
29. Allman M, Balakrishnan H, Floyd S. Enhancing TCP's loss recovery using limited transmit. *IETF RFC 3042*, June 2001.
30. Ludwig R, Gurtov A. The Eifel response algorithm for TCP. *IETF RFC 4015*, February 2005.
31. Sarolahti P, Kojo M. Forward RTO-recovery (F-RTO): An algorithm for detecting spurious retransmission timeouts with TCP and the stream control transmission protocol (SCTP). *IETF RFC 4138*, August 2005.
32. Ramakrishnan K, Floyd S. A proposal to add explicit congestion notification (ECN) to IP. *IETF RFC 3168*, September 2001.
33. Spring N, Wetherall D, Ely D. Robust explicit congestion notification (ECN) signaling with nonces. *IETF RFC 3540*, June 2003.
34. Eggert L, Gont F. TCP user timeout option. *IETF RFC 5482*, March 2009.
35. Mogul J, Kent C, Partridge C, McCloghrie K. IP MTU discovery options. *IETF RFC 1191*, November 1990.
36. Mathis M, Heffner J. Packetization layer path MTU discovery. *IETF RFC 4821*, March 2007.
37. Floyd S, Arcia A, Ros D, Iyengar J. Adding acknowledgement congestion control to TCP, IETF Work in Progress, TCPM working group. *draft-floyd-tcpm-ackcc-06*, March 2009.
38. Akyildiz IF, Morabito G, Palazzo S. TCP Peach: A new congestion control scheme for satellite IP networks. *IEEE/ACM Transactions on Networking* 2001; **9**(3):307–321.
39. Casetti C, Gerla M, Mascolo S, Sanadidi M, Wang R. TCP Westwood: bandwidth estimation for enhanced transport over wireless links. *Proceedings of ACM Mobicom 2001*, Italy, July 2001; 287–297.
40. Brakmo L, Peterson L. TCP Vegas: end to end congestion avoidance on a global Internet. *IEEE Journal in Selected Areas in Communications* 1995; **13**:1465–1480.
41. Caini C, Firrincieli R. TCP Hybla: a TCP enhancement for heterogeneous networks. *International Journal of Satellite Communications and Networking* 2004; **22**(5):547–566.
42. Durst R, Miller J, Travis E. TCP extensions for space communications. *Second ACM International Conference on Mobile Computing and Networking* (*Mobicom*), New York, U.S.A., November 1996.
43. Space Communications Protocol Specification (SCPS) Transport Protocol (SCPS-TP). *CCSDS 714.0-B-2, MIL-STD-2045-44000, ISO 15893*, The Consultative Committee for Space Data Systems, 2006.
44. Roseti C, Kristiansen E, TCP Noordwijk: optimize TCP-based transport over DAMA in satellite networks. *IEEE International Workshop on Space and Satellite Communications* (*IWSSC*), Leganes, Spain, 2006.
45. Xu L, Harfoush K, Rhee I. Binary increase congestion control for fast, long distance networks. *IEEE Infocom*, Hong Kong, 2004.
46. Ha S, Rhee I, Xu L. CUBIC: a new TCP-friendly high-speed TCP variant. *ACM SIGOPS Operating System Review* 2008; **42**(5):64–74.
47. Tan K, Song J, Zhang Q, Sridharan M. Compound TCP: a scalable and tcp-friendly congestion control for high-speed networks. *Fourth International workshop on Protocols for Fast Long-Distance Networks* (*PFLDNet*), Nara, Japan, May 2006.
48. Eggert L, Gont F. TCP user timeout option. *IETF RFC 5482*, March 2009.
49. I-PEP specifications. Satlabs group recommendation, October 2005. Available from: http://www.satlabs.org.
50. Performance Enhancing Proxies (PEPs). *ETSI TR 102 676*.
51. Caini C, Firrincieli R, Lacamera D. PEPsal: a performance enhancing proxy designed for TCP satellite connections. *IEEE Vehicular Technology Conference* (*VTC*), Montreal, Canada, 2006.
52. Floyd S, Allman M, Jain A, Sarolahti P. Quick-start for TCP and IP. *IETF RFC 4782*, January 2007.
53. Stewart R (ed.). Stream control transmission protocol. *IETF RFC 4960*, September 2007.
54. Stewart R, Ramalho M, Xie Q, Tuexen M, Conrad P. Stream control transmission protocol (SCTP) partial reliability extension. *IETF RFC 3758*, May 2004.
55. Kohler E, Handley M, Floyd S. Datagram congestion control protocol (DCCP). *IETF RFC 4340*, March 2006.
56. Postel J. User datagram protocol. *STD 6*, *IETF RFC 768*, August 1980.
57. Eggert L, Fairhurst G. Unicast UDP usage guidelines for application designers. BCP 145, *IETF RFC 5405*, November 2008.
58. Larzon L-A, Degermark M, Pink S. 'The Lightweight User Datagram Protocol (UDP-Lite)', *IETF RFC 3828*, Jonsson L-E, Fairhurst G. (eds), July 2004.

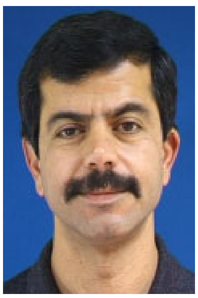
DOI: 10.1002/sat

59. Whetten B *et al*. Reliable multicast transport building blocks for one-to-many bulk-data transfer. *IETF RFC 3048*, January 2001.
60. IETF RMT Charter. Available from: http://www.ietf.org/dyn/wg/charter/rmt-charter.html.
61. Kristiansen E, Neves J, Brazio J, Pagano G, Palazzo S. Re-thinking the end-to-end paradigm—the missing protocol layer. *ICSSC*, Monterey, California, 2004.
62. Kristiansen E *et al*. Satellite network transport architecture (SaNTA). *Twenty-third AIAA International Communications Satellite Systems Conference* (*ICSSC*), Monterey, U.S.A., 2005.
63. Ford B, Iyengar J. Breaking up the transport Logjam. *Seventh ACM Workshop on Hot Topics in Networks* (*HotNets-VII*), Canada, October 2008.
64. Fairhurst G, Sathiaseelan A. Quick-start for the datagram congestion control protocol (DCCP). *IETF RFC 5634*, August 2009.

AUTHORS' BIOGRAPHIES

Prof. Gorry Fairhurst received his first degree in Applied Physics and Electronics from the University of Durham, U.K., and a PhD in Communications Engineering from the University of Aberdeen, U.K. He is a Professor in the School of Engineering at the University of Aberdeen with has over a decade of relevant experience in the development of new algorithms and protocols. His current research focuses on transport protocol design for broadband systems including IP over satellite. This includes link ARQ, link-specific tuning of protocols, TCP, development of multicast transport, and design of satellite Internet systems. He is an active participant in the specification and engineering of telecommunications standards, and contributes to ETSI and the IETF. He currently chairs the IETF IPDVB and TSVWG working groups.

Arjuna Sathiaseelan is a Research Fellow in the School of Engineering at the University of Aberdeen. His current research interests are in the integration of satellite broadband access in Next Generation Networks. He leads work on network simulation, design of multimedia transport protocols, TCP protocol optimization, and performance analysis. He contributes to the IETF DCCP working group.



Haitham Cruickshank is a Lecturer at the University of Surrey. He has worked there since January of 1996 on several European research projects in the ACTS, ESPRIT, TEN-TELECOM, and IST programs. His main research interests are network security, satellite network architectures, VoIP, and IP conferencing over satellites. He also teaches Data and Internet Networking and Satellite Communication courses. He is a member of the Satellite and Space Communications Committee of the IEEE ComSoc and a chartered engineer and corporate member of the IEEE in UK. He is an active member of the IETF and ETSI BSM working group in the security area.

Cédric Baudoin has a 13 years of experience in networking and access technologies. In 1996 he joined Thales Alenia Space as Access and Simulation Engineer in the Ground Systems Division. He is an expert in simulation and emulation for satellite multimedia broadband systems and has been involved in access and system definition (Radio Resource Management, connection control, access/network interfaces) on various programs and studies (Skybridge, IntelSat, PEA OBP EHF, Domino2, etc.). Since 2005, he is an access and network expert in the Research department dealing with QoS and IPv6 studies. He has been especially involved in the SatSix IST project as a technical manager. He has written or contributed to 35 papers. He is currently working on All-IP and constellation satellite systems, aeronautical communications, convergence and mobility in the frame of several projects (CNES NGN/IMS, IST NEWSKY, Iridium Next) and studies.

DOI: 10.1002/sat