

# Effects on TCP of Routing Strategies in Satellite Constellations

Lloyd Wood, George Pavlou, and Barry Evans, University of Surrey

## ABSTRACT

A broadband satellite network uses a constellation of a number of similar satellites to provide wireless networking services to the Earth. A number of these constellation networks are under development. This article introduces the types of satellite constellation networks, and examines how overall performance of TCP communications carried across such a network can be affected by the choice of routing strategies used within the network. Constellations utilizing direct intersatellite links are capable of using multiple paths between satellites simultaneously as a strategy to spread network load. This allows more general routing strategies than shortest-path routing, but we show these strategies to be detrimental to the performance of individual TCP connections.

## INTRODUCTION

Broadband satellite constellation networks have been proposed and are now being developed. These will provide medium- and high-capacity wireless data services, and will interconnect with existing telco backbones and the Internet. Once launched and operational, proposed networks, such as Teledesic, Spaceway, and Skybridge, can be expected to transport significant amounts of TCP traffic between ground endhosts on the Internet.

Although the use of satellite constellations for delivering broadband IP services is new, satellite-constellation-based systems are already being used for applications such as providing:

- Navigation and position information: Global Positioning System (GPS) and GLONASS
- Voice telephony and low-bit-rate communication services (although not financially successful, the Iridium and Globalstar systems have made service available, and ICO Global Communications is to follow)
- Low-bit-rate messaging data services for a variety of applications (e.g., Orbcomm)

The altitude of the satellites in the constellation is a significant factor in determining the

number of satellites required to cover the Earth. A lower altitude decreases free space loss and propagation delay, but means that the service each satellite can offer is limited to users in a smaller visible area of the ground (the satellite's *footprint*). To fully cover the globe, more satellites are needed. This increases frequency reuse and overall system capacity, but will also increase overall system construction and maintenance costs. Satellites at lower altitudes must move faster relative to the ground to stay in their orbits, increasing the rate of handoff and Doppler effects between terminals and satellites.

Most proposed systems use circular orbits with constant altitudes, since this means that satellite overhead pass times and power levels needed for communication are constant, and each satellite can be used for traffic throughout its orbit. Other systems have been proposed using multiple elliptical orbits, where satellites are only intended to be available for service while moving relatively slowly around high-altitude apogee (e.g., the commercial Ellipso and Pentriad proposals). In those elliptical systems the communication payloads are unused throughout the rest of the orbit.

Figure 1 compares the altitudes of the systems discussed above.

## GEOMETRY, TOPOLOGY, AND DELAY

From a networking viewpoint, there are two fundamentally different approaches taken by these broadband constellations. Each constellation consists of a number of similar satellites, where the satellite design is based on one of the following approaches:

- Each satellite is a space-based retransmitter of traffic received from user terminals and local gateways below it on the ground, returning the traffic to the ground. This allows isolated user terminals to exchange traffic with nearby ground stations that are gateways into the terrestrial network. The satellite forms a wireless last hop to an extensive ground network. Commercial sys-

tems taking this approach include Globalstar and many geostationary satellites, and the planned ICO and Skybridge.

- Each satellite is a network switch that is also able to communicate with neighboring satellites by using radio or laser *intersatellite links* (ISLs). This allows user terminals on the ground below the satellite to exchange traffic with gateways to the terrestrial network or users below distant satellites, without requiring a local gateway or significant terrestrial infrastructure to do so. Commercial systems utilizing ISLs include the deployed Iridium and the proposed Teledesic, Hughes Spaceway, and Astrolink networks.

Proposals with circular orbits are the most common, while proposals using ISLs are the most interesting and complex from a networking and systems viewpoint, so we will focus on those.

Geostationary satellites can be interconnected to form a simple ring network around the Earth's equator, as shown in Fig. 2a for the minimum three geostationary satellites needed to cover most of the Earth's surface. At low- and medium-earth-orbit altitudes the ISLs between satellites form a more complex dynamic mesh topology. That topology is a variation of the Manhattan network, named after the grid of streets in Manhattan, which is well-known in theoretical computer networking for use in parallel computer architectures. Depending on the specific orbital geometry chosen, this overall satellite topology will be either:

- A fully toroidal network, created from an inclined *Walker delta* or *Ballard rosette* constellation [1], where the ascending (moving northward) and descending (moving southward) planes overlap and span the full 360° of longitude; for example, the Hughes Spaceway NGSO MEO proposal (Fig. 2b). This geometry offers the best satellite visibility and highest degree of satellite capacity over the populated mid-latitudes.
- A form of cylindrical mesh network, from a near-polar *Walker star* constellation [2], where the interconnected ascending and descending orbital planes of satellites each cover around 180° of longitude and are separate. Systems using this approach include Iridium and the proposed Teledesic system (a proposed Teledesic design is shown in Fig. 2c). This geometry offers the best satellite visibility at the poles.

In a near-polar star constellation, the gap between the counter-rotating ascending and descending planes, shown in the Teledesic design in Fig. 2c, is known as the *counter-rotating seam*. Cross-seam ISLs, which interconnect satellites at the edges of the cylindrical mesh to form a seamless spherical network, must be handed off frequently as the planes of satellites move past each other. These cross-seam links have not yet been demonstrated in practice; Iridium, the only system constructed using ISLs to date, has none.

However, cross-seam ISLs do have a considerable effect on network performance. Figure 3 shows the total propagation delay experienced by traffic taking the shortest path across the constellation networks between two fixed

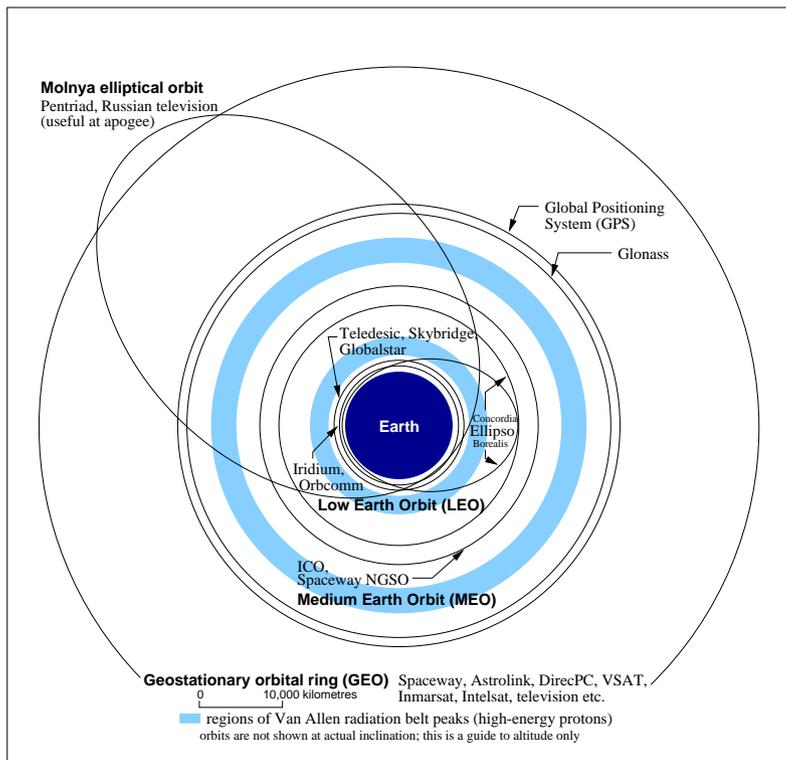


Figure 1. Orbits used by satellite constellations.

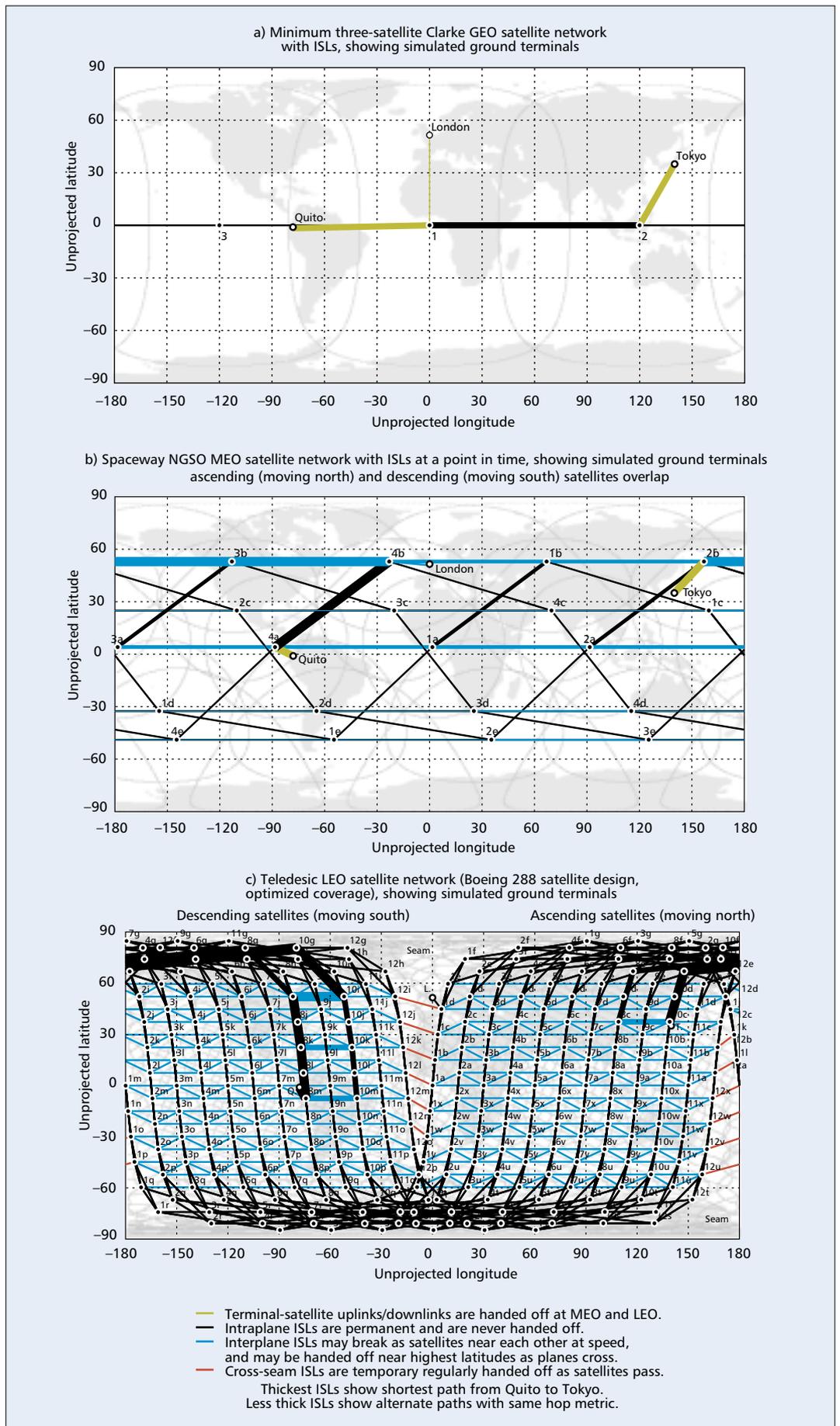
ground terminals over the course of a day, as the Earth rotates beneath the moving satellites. Here, we see smooth changes in delay due to the orbital motion of satellites, as well as abrupt changes due to handoffs and path changes; these changes are small compared to the granularity of a TCP timer.

Network latency results from queuing, switching, and link-layer medium access control (MAC) bandwidth allocation as well as from serialization and propagation delays. With the exception of the total path propagation delay shown in Fig. 3i, where the speed of light is a common constraint, these contributing delays will vary considerably according to the specific system design. As a first approximation, these delays can be thought of as proportional to the number of wireless links traversed, indicated in Fig. 3ii. (Teledesic's redundant "geodesic" eight-link design, shown in Fig. 2c, minimizes delay and the number of links traversed, while also providing the two ISLs needed for frequent smooth acquisition and handoff of a single link across the seam.)

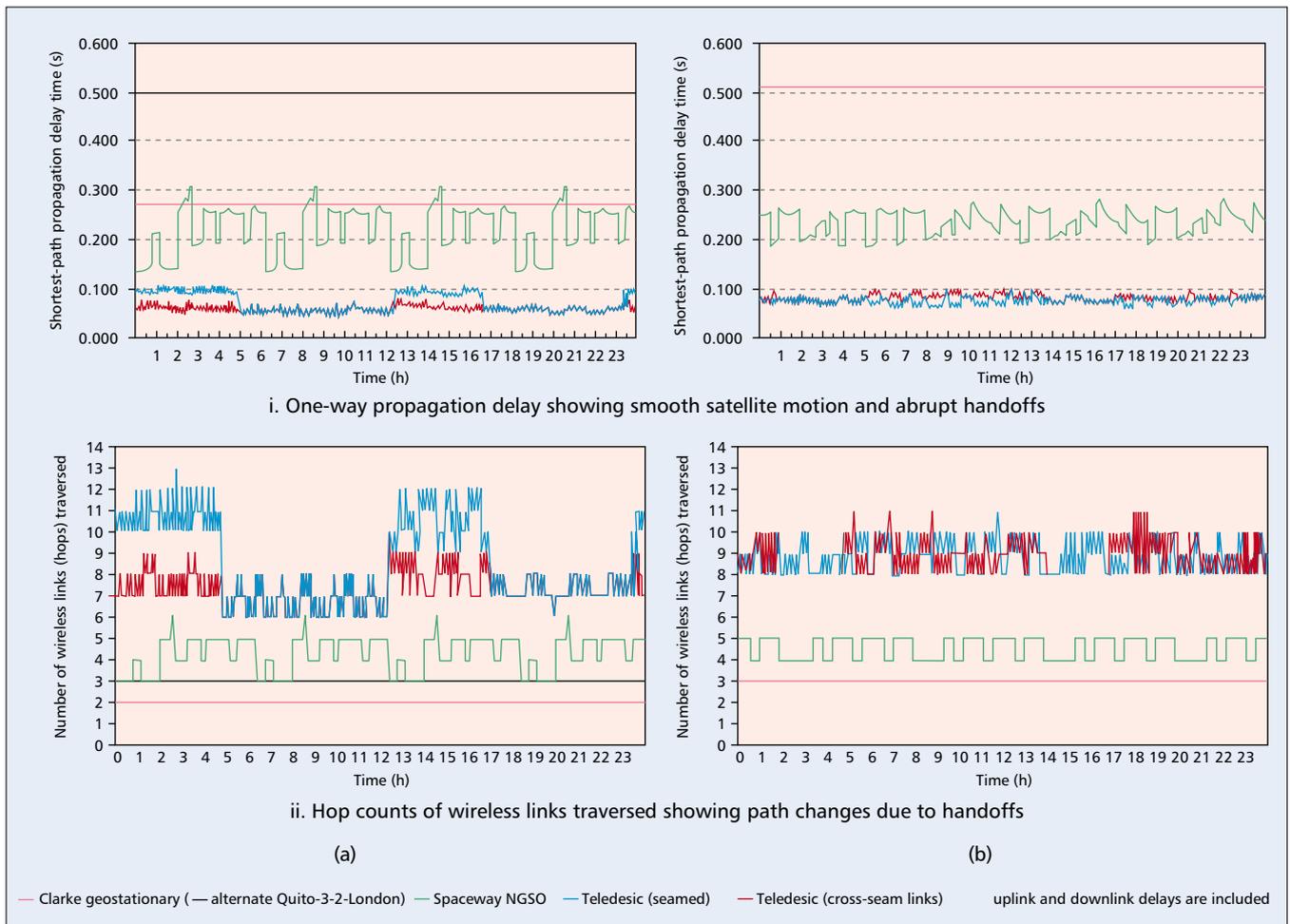
Examining the shortest-path propagation delay across a network provides a best-effort goal with which real-world performance could be compared. However, relying on a metric of minimum propagation delay for cost-based routing will tend to load interplane links at the highest latitudes, where distances between satellites are shortest.

Without cross-seam links, the seam increases both delay and the number of links traversed when it intersects the shortest path between ground terminals. The seam alters the shortest path between the terminals so that the path lengthens and lies over the highest latitudes. The duration of this disruption is proportional to separation in longitude between the terminals,

Unlike the “polar star” constellations, rosette constellations can also offer two distinct sets of paths between terminals, depending on whether the terminals’ uplink and downlink communications have been allocated to ascending or descending satellites.



■ Figure 2. Satellite constellation networks at different altitudes.



**Figure 3.** One-way total propagation delays of path across the constellations over 24 hours: a) routing packets between ground terminals from Quito to London; b) routing packets between ground terminals from Quito to Tokyo.

due to the Earth's rotation. Similarly, the size of the change in path is inversely related to the separation between terminals. As Quito and London are relatively close, traffic between them, shown in Fig. 3a, shows this disruption more clearly than traffic between Quito and Tokyo (Fig. 3b). Cross-seam links reduce the maximum propagation delay, path lengths, path changes and visible end-to-end disruption of network traffic in star constellations.

Rosette constellations with ISLs, such as the MEO Spaceway NGSO network shown in Fig. 2b, offer more regular repeating routing and delay patterns over time between terminals. Unlike the "polar star" constellations, rosette constellations can also offer two distinct sets of paths between terminals, depending on whether the terminals' uplink and downlink communications have been allocated to ascending or descending satellites. The satellites forming the shortest set of ISL paths for the traffic may not be selected by the terminal if the terminal is at a high latitude and does not have a wide choice of satellites, if demand for link capacity around a terminal is high, preventing a choice of the "optimum" satellite for that traffic, or if handoff is not coordinated for traffic across the entire network but is a local terminal decision. This is the case for our simulations, where handover

takes place only when the currently used satellite passes below the terminal's minimum elevation angle threshold. If the terminal at Quito in Fig. 2b was allocated to the descending satellites of 2c or 2d, it would be only one or two ISL hops from Tokyo, instead of three while using satellite 4a.

## TCP AND IP

TCP/IP is the protocol suite on which the Internet depends. IP provides a connectionless datagram service that provides enough header information for packets to be routed to their destinations. TCP makes reliable ordered communication of streams of data such as files possible by implementing a duplex protocol, based around sliding windows, on top of IP [3].

A number of intertwined algorithms determine the rate at which TCP puts new data into the network, and how losses are detected and handled. In particular, a TCP sender maintains a *congestion window* variable that determines how large the size of its sliding window of unacknowledged data given to the network can be at any point in time. This variable is increased over time from start, or after congestion losses, as the TCP sender probes the network and discovers that data can be sent with increasing fre-

Given the widespread use of TCP/IP-based applications and interconnection with the terrestrial Internet, it is likely that broadband satellite networks will be transporting large amounts of traffic generated by TCP's algorithms.

quency. It is decreased when losses due to congestion are inferred by the sender. Slowstart and other algorithms control the congestion window and other TCP settings. These are described in detail in [4].

## TCP IN THE SATELLITE ENVIRONMENT

Working across satellite links has been a motivation in the design of TCP since the mid-1970s. TCP treats all losses on the journey between source and destination as an indication of congestion, or overflowing packet buffers in router queues leading to discards. Any losses due to errors in transmission on a less-than-perfect satellite link, rather than congestion, cause TCP to reduce its overall sending rate to help alleviate perceived congestion.

Communication between terminals using a geostationary satellite has a comparatively long round-trip time (RTT), from sender to receiver and back, on the order of 0.5 s. This means that considerable time is needed in slowstart, on opening a new connection for TCP, to increase its throughput to what the link capacity will bear. After an errored packet is inferred as lost due to congestion, and the congestion window is reduced, the return to the previous high rate of throughput is slowed by the large RTT. In both cases, increasing the congestion window takes a number of slow 0.5-s roundtrips, and the capacity of an expensive satellite link will not be fully used as a result. Satellite considerations for TCP are detailed in [5, 6].

Given the widespread use of TCP/IP-based applications and interconnection with the terrestrial Internet, it is likely that broadband satellite networks will transport large amounts of traffic generated by TCP's algorithms, even if the satellite networks are not themselves based around the IP packet structure.

The desire to utilize satellite capacity efficiently has led to the development of a number of optimized-for-satellite TCP variants, with altered transmission control behavior. These run between ground terminals across the satellite link, while other TCP implementations complete the end-to-end communication across the terrestrial Internet. The end-to-end TCP connection across the congested but relatively link-error-free terrestrial Internet and the less congested but more error-prone satellite link is *split* into individual TCP connections tailored to each environment; this is easy to do with an HTTP proxy cache, but less elegant in implementation when other applications are concerned.

Improvements in coding on satellite links have altered the link error characteristics to be in one of two states: either error-free or extremely errored. With a correctly dimensioned satellite link, we can assume that the satellite link is as reliable as any terrestrial link.

## TCP AND MESH ROUTING

Unlike terrestrial networks, where there is often only one fixed route between sender and receiver at any point in time, MEO and LEO ISL

meshes, such as the Spaceway NGSO and Teledesic networks shown in Fig. 2, can offer more than one possible path for packets to take between two satellites communicating with ground terminals. When the path is longer than one ISL hop, there are likely to be a number of equivalent paths, of the same number of hops between the satellites, whose queuing and framing delays will be roughly equivalent. If a ground terminal has the opportunity to communicate with more than one satellite visible to it above its local horizon (using "diversity"), the number of possible paths between source and destination terminals can be increased still further.

This redundantly connected multipath ISL topology allows a change in behavior when network congestion occurs: rather than dropping packets at routers instead of queuing them on a congested path, the routers can instead divert those packets to another, less congested, path instead, introducing load balancing.

This gives us the opportunity to examine TCP's behavior when exposed to multipath routing in the satellite environment. We found that, of TCP's algorithms, two in particular affect TCP's performance over multipath routing: *fast retransmit and recovery* and *delayed acknowledgments*.

## THE EFFECTS OF FAST RETRANSMIT AND RECOVERY

A TCP sender can infer congestion in the network and losses of data transmitted from two mechanisms:

- A *timeout*, where no new acknowledgments are received for a calculated period of time, typically a multiple of 0.5 s. The sender believes that all traffic sent to the receiver is being lost due to congestion, and attempts to avoid contributing to this perceived congestion by reducing its congestion window to one segment. It then probes the network with an exponentially increasing flow of packets in slowstart, much as it did when first sending data when the connection was opened.
- Receipt of a number of duplicate acknowledgments, *dupacks*, where the receiver sends an acknowledgment which repeats the current position of the left edge of the TCP window. The fact that the receiver is issuing dupacks indicates it is receiving new data from the sender — but not the data necessary to be able to deliver previous data received to its application and move the left edge of the window along, allowing the sender to inject more new data into the network. A loss just after the repeatedly indicated window position can therefore be inferred by the sender, which can do a fast retransmission of the lost packet. The sender assumes that the loss is due to congestion in the network and reduces its congestion window to take this into account. Since the received dupacks indicate that data is still getting through to the receiver, the sender can be less conservative in reducing its window to avoid congestion than when encountering a timeout. The conges-

tion window, controlling the rate at which new data is sent, is halved before growing slowly to its previous size as new data is acknowledged. This is fast recovery.

Receipt of three dupacks, after the original ack of the last in-sequence packet received, is taken to indicate that a packet has been lost in the network. Three is an ideal value for an ordered flow of packets between sender and receiver, since it gives the fast warning of congestion-induced losses for which the retransmit and recovery processes were named, while coping with minor amounts of packet interleaving and reordering.

However, if more than one possible route between source and destination can be used simultaneously, as in a mesh of ISLs between satellites where congestion leads to dynamic rerouting rather than straightforward discards, packets can be received out of order due to slight differences in latency between paths. This will cause dupacks to be issued even when no losses have taken place. Any resulting fast retransmission and recovery will be entirely unnecessary and detrimental to TCP's performance.

To examine this, we used the network simulator *ns* [7] to simulate multipath routing of traffic between Quito and Tokyo across all available routes with the same number of wireless hops at a point in time. We simulated approximations of proposed Teledesic and Spaceway NGSO designs, based on details given in their applications to the U.S. FCC for frequency allocation.

We compared shortest-path routing, based on using the path with the smallest propagation delay, with arbitrary multipath routing, selecting any minimum-hop path to the destination at a point in time. This multipath routing approximates the behavior of temporarily congested satellites diverting newly arrived packets from a currently congested link onto a less congested alternate path, or simple "hot-potato" local routing decisions. The choice of path at each satellite was determined by a simple round-robin selection, updated on the arrival of each new packet. This made each multipath simulation deterministic and repeatable.

Since we wanted only to examine the effects of path diversion on TCP without confusing its effects with the effects of congestion, we presumed that the links were both error- and congestion-free, allowing any difference in propagation delays of the links to reorder the packets. We presumed high-capacity links — 5 Mb/s uplink and downlink, and 10 Mb/s ISL links — so that the time required to receive each 1000-byte packet was smaller than the delay introduced by each wireless hop, but imposed a small receiver window so that the TCP connection would not fill the channel. We simulated a bulk FTP transfer of a large file over a period of 100 s: a time short enough that terminal-satellite handoffs and resulting routing transients did not affect our simulations. We used abstractions of two widely implemented variations of TCP:

- New Reno, based on a stack implementation originally developed on a machine named Reno, and widely deployed since with some modifications [8]

- Selective acknowledgments (SACK), an enhancement to the Reno algorithms that provides more information in the TCP options field of acks sent from receiver to sender, allowing the sender to retransmit specific data indicated as unreceived [9]

The graphs in Fig. 4 show the effects of fast retransmit and recovery on overall TCP throughput, as a result of dupacks received due to out-of-order reception of both segments and acks over the multiple paths across the ISL meshes. In each case, TCP's throughput, or *goodput*, is degraded from its best performance over a single shortest path by the use of multiple paths, even though there are no losses due to transmission errors or congestion in the network.

By increasing the arbitrary requirement of three dupacks before entering fast retransmit and recovery to a higher threshold, we were gradually able to improve TCP's throughput to approach that when using the single shortest path. Similarly, lowering the dupack threshold decreased even further TCP's tolerance of misordering caused by interleaving packets sent over multiple paths and lowered its resulting throughput still further.

Although the mesh topologies of these satellite networks provide multiple paths between points and are clearly multipath, packet reordering has been observed in the terrestrial Internet due to parallelism or load balancing [10], and can be considered a natural state of affairs rather than something rare or easily avoided.

SACK's performance could be improved by using the dupacks' ack options field to provide more information to indicate what is causing each dupack to be generated, rather than just sending a dupack without extra information [11]. This can then be used by the sender to infer reordering in the network and to alter its behavior to suit, although this is still a research area.

## THE EFFECTS OF DELAYED ACKNOWLEDGMENTS

In most TCP implementations, not all segment-bearing packets are immediately acknowledged by the receiver as you might expect. Instead, a nominally optional, but widespread, delayed acknowledgment mechanism is used. This allows the receiver to skip acknowledging TCP segments before issuing a cumulative ack covering all the segments received since the last ack was sent.

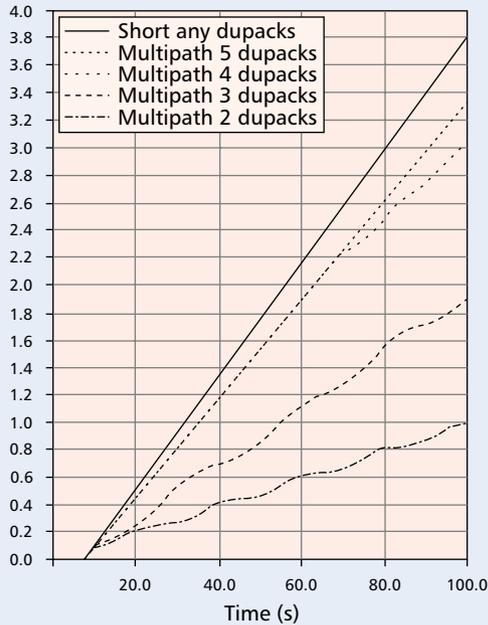
The receiver should acknowledge every second in-order TCP segment received, and should wait between 0.1 and 0.5 s for new packets containing TCP segments to arrive before issuing an ack. This allows acks to be piggybacked on any data segments that may be sent by the receiver (since TCP is duplex), reducing overall network traffic.

In practice, it is commonplace to wait for a second segment before sending an ack. The receiver can wait until a system timer goes off every 200 ms, when an ack must be sent. We included the common 200 ms delay when simulating the effect of the dupack threshold on multipath throughput, as shown in Fig. 4.

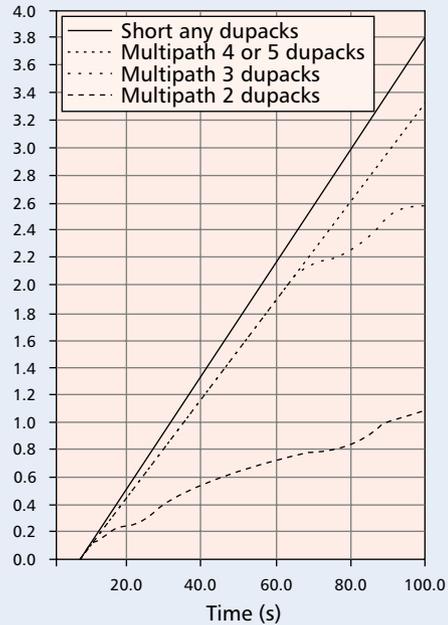
Only in-order segments are subject to this delay. Out-of-order segments are generally

*The choice of path at each satellite was determined by a simple round-robin selection, updated on the arrival of each new packet. This made each multipath simulation deterministic and repeatable.*

FTP transfer over Spaceway NGSO using TCP New Reno  
Amount of file transferred as seen by application (K)  $10^3$

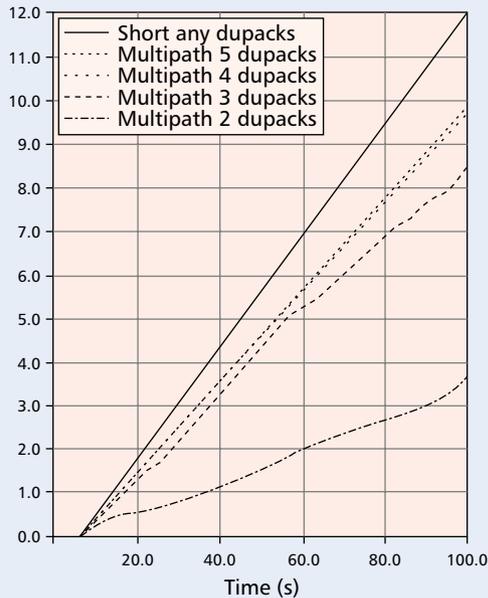


FTP transfer over Spaceway NGSO using TCP SACK  
Amount of file transferred as seen by application (K)  $10^3$

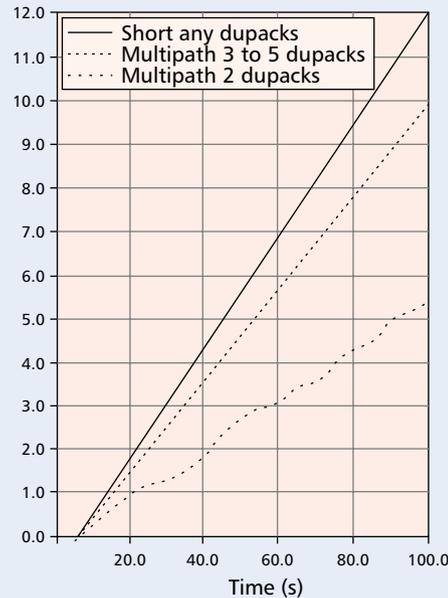


i. Progress of FTP transfer between terminals at Quito and Tokyo using Spaceway NGSO

FTP transfer over Teledesic using TCP New Reno  
Amount of file transferred as seen by application (K)  $10^3$



FTP transfer over Teledesic using TCP SACK  
Amount of file transferred as seen by application (K)  $10^3$



ii. Progress of FTP transfer between terminals at Quito and Tokyo using Teledesic

(a)

(b)

**Figure 4.** The effect of fast recovery dupack threshold on throughput over multiple paths: a) transfers using New Reno TCP; b) transfers using SACK TCP.

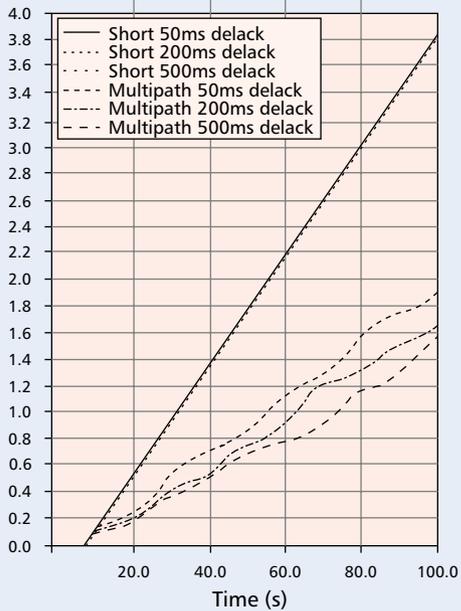
acknowledged immediately, on the principle that this dupack information is useful to the sender in determining what to resend in the case of losses. The effects of delayed acks are discussed in detail in [12].

Delayed acks have the advantages of conserving processing resources at the receiver, and decreasing back traffic and the number of packets and resulting load along the return path to

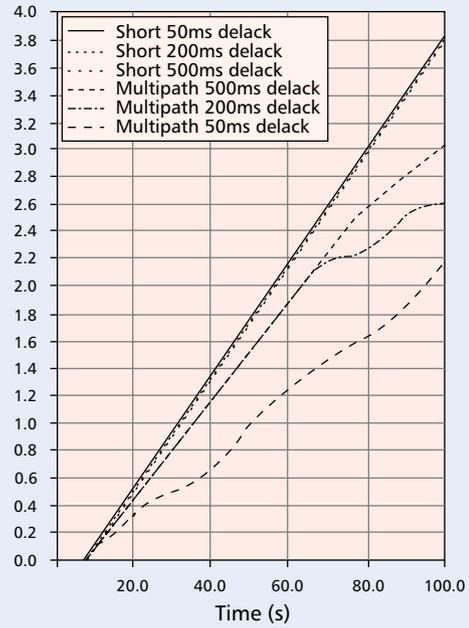
the sender. This is useful for asymmetric networks, such as DirecPC, where the downlink can be around 400 kb/s via broadcast from geostationary satellite, but the uplink is actually a terrestrial dialup modem of 56 kb/s or less.

However, delayed acks are widely recognized as degrading TCP throughput in some situations. Since TCP slowstart at the beginning of a transfer generally uses the number of acknowledg-

FTP transfer over Spaceway NGSO using TCP New Reno  
Amount of file transferred as seen by application (K)  $10^3$

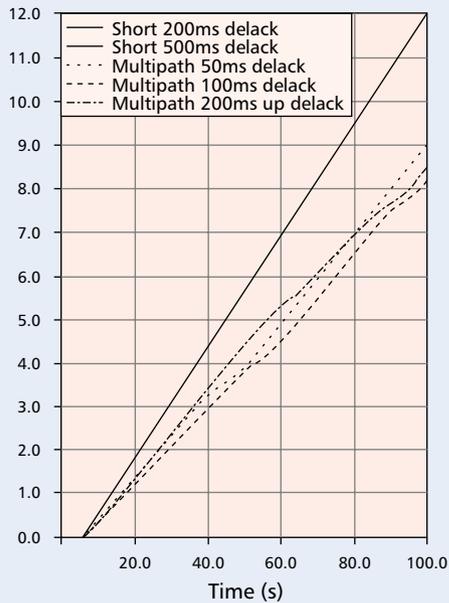


FTP transfer over Spaceway NGSO using TCP SACK  
Amount of file transferred as seen by application (K)  $10^3$

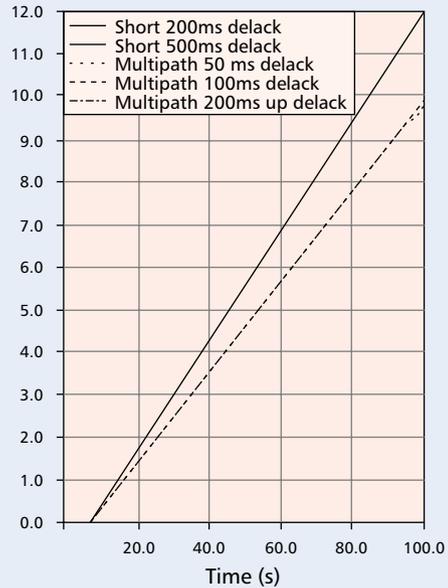


i. Progress of FTP transfer between terminals at Quito and Tokyo using Spaceway NGSO

FTP transfer over Teledesic using TCP New Reno  
Amount of file transferred as seen by application (K)  $10^3$



FTP transfer over Teledesic using TCP SACK  
Amount of file transferred as seen by application (K)  $10^3$



ii. Progress of FTP transfer between terminals at Quito and Tokyo using Teledesic

(a)

(b)

■ **Figure 5.** Delayed acks degrading the rate of file transfer over multiple paths; a) transfers using New Reno TCP; b) transfers using SACK TCP.

ments received as an indication of how much new traffic can be injected into the network, the initial slowstart phase is slowed further by the decreased number of acks sent by a delayed-ack receiver. Increase of the congestion window after a timeout or in fast recovery is similarly slowed by half. From the viewpoint of Internet traffic as a whole, these effects can be considered beneficial, since window growth is damped and individ-

ual TCP flows using delayed acks are less aggressive in gaining network capacity.

It is this damping which causes delayed acks to adversely affect TCP's sensitivity to out-of-order segments received over multiple paths. Although receivers with and without delayed acks should behave identically when receiving out-of-order segments — issuing acks immediately so that the sender can make the same deci-

*It is unlikely that any broadband satellite constellation will be a true IP packet-switching network based on the IP packet structure. Instead, the need for management of frequency allocation in the terminal uplink and downlink dictates the need for MAC with a fixed frame size.*

sion whether to do a fast retransmit after receiving three dupacks and enter fast recovery — the growth of the congestion window after entering fast recovery is slowed because the sender receives fewer acknowledgments to in-order packets.

If entering fast recovery becomes a regular occurrence, as we have seen happen in multipath environments, the delayed growth of the congestion window severely limits overall throughput, and considerably more time is spent in fast recovery with a low congestion window setting.

Figure 5, from our *ns* simulations as described earlier, shows that throughput in multipath environments can be affected considerably by alterations in receiver ack delays. The effect on shortest-path throughput, due to the change in slowstart when the connection is opened, is comparatively minimal. Delayed acks contribute to the degradation of TCP's performance in multipath environments. This degradation could be reduced by selectively avoiding use of delayed acknowledgements when the TCP sender is attempting to grow its congestion window. The sender would need to provide additional information to the receiver to make this possible. Such a change has already been suggested in [12] for the slowstart algorithm, and would benefit TCP traffic over geostationary satellites by speeding up the initial startup and post-timeout phases. However, the impact of such changes on the characteristics of Internet traffic as a whole is unknown.

Delayed-ack use has been tightened in specification so that only acks acknowledging receipt of *new* in-order packets received at the right edge of the window are delayed, rather than delaying all acks to all in-order packets. Acks to packets "filling in" gaps in window data should be sent immediately, even if the packet is "in order" relative to the previous packet. As a *should*, this is optional, although recommended. Following this recommendation, as the simulations presented here do, goes part of the way toward avoiding use of delayed acks when the TCP sender is attempting to grow its congestion window in reordering-induced fast recovery. This increases FTP throughput over that of receivers not following this recommendation, particularly for SACK's fast recovery algorithm.

## PRACTICAL CONSIDERATIONS

It is possible that dynamic routing changes due to link failures or handoff in a satellite constellation could result in dupacks and a drop into fast recovery at high throughputs, due to loss of packets being transmitted between nodes, or to later interleaving of packets and acknowledgements in flight along both original and new routes as routing information is propagated. Such transient effects are discussed further in [13].

It is unlikely that any broadband satellite constellation will be a true IP packet-switching network based on the IP packet structure. Instead, the need for management of frequency allocation in the terminal uplink and downlink dictates the need for MAC with a fixed frame size. A number of proposed schemes plan a MAC frame encapsulating ATM cells, so IP packets would be tunneled over an ATM-based network.

The recent advent of multiprotocol label switching (MPLS) can allow IP routing of ATM flows of IP traffic, and seems promising for satellite constellation networks since it allows support for IP multicast and quality of service (QoS) [14]. How such IP routing would be implemented (e.g., for optical ISLs, possibly using wavelength-division multiplexing), remains to be seen.

Although TCP is tolerant of out-of-order traffic, other non-IP-based traffic is less so and would be even more likely to be routed across a single path, with priority over best-effort Internet traffic. If flow-unaware multipath routing is used, it would be possible to have buffering and reordering of entire traffic flows at the edges of the satellite network, rather than just the minimal buffering needed to reconstruct higher-layer frames from MAC-level frames. However, this would add to overall latency, state held in the network, and system complexity.

It would be practical to implement split-TCP connections across the constellation network, where the TCP connection used across the satellites is optimized for local conditions. If the constellation uses multipath routing, the TCP implementation used in terminals and gateways for satellite communication would include modifications, such as a higher dupack threshold for fast recovery, to improve performance relative to end-to-end TCP traffic, while the TCP implementation used for communicating with the terrestrial Internet is configured differently for the terrestrial environment. This approach is reminiscent of many of the split-TCP implementations used with existing geostationary satellites today.

## CONCLUSIONS

Examining TCP over LEO and MEO satellite constellation networks with ISLs has clearly shown that using multiple paths to spread network load and avoid packet drops due to congestion can cause a TCP sender to behave just as if it was on a single congested path where packets are dropped. This is due to TCP's fast recovery algorithm, which is intended to handle losses in an ordered flow of segments. Delayed acknowledgments can damp window growth to impair the performance of TCP over multipath routing still further.

Although TCP tolerates the misordering that results from multipath routing and load balancing, its overall throughput suffers as a result of its assumptions about congestion. Performance considerations dictate a single ordered flow of traffic between source and destination. TCP is encouraging a circuit or flow paradigm in the underlying packet networks. The design of TCP is affecting and restricting the design of future networks. TCP should be examined with a view to improving its tolerance of load balancing and multipath routing.

As a result of the design of TCP, various Manhattan mesh or simple forwarding approaches to routing, based on local information, are less desirable than a global shortest-path routing or flow-based approach within the satellite constellation. The effects of simple flow-unaware routing approaches can be compensated for by reordering of packets at terrestrial gateways or implementing

a split-TCP approach, using a satellite-optimized TCP with more tolerance of misordering-induced duplicated acknowledgments.

Although examining bulk FTP transfers clearly shows the limits of TCP's tolerance for multipath routing, future work would examine the impact of multipath routing on the shorter, burstier, transactional data transfers seen using HTTP.

#### ACKNOWLEDGMENTS

We thank the maintainers of and contributors to the network simulator *ns*, used in our simulations, for its ongoing development. Tom Henderson's work in extending *ns* to simulate satellite constellation networks more realistically was particularly valuable. That work and exploration of split-TCP implementations are described in detail in [15].

#### REFERENCES

- [1] A. H. Ballard, "Rosette Constellations of Earth Satellites," *IEEE Trans. Aerospace and Elect. Sys.*, vol. 16, no. 5, Sept. 1980, pp. 656-73.
- [2] J. G. Walker, "Satellite Constellations," *J. Brit. Interplanetary Soc.*, vol. 37, 1984, pp. 559-71.
- [3] W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*, Addison-Wesley, 1994.
- [4] M. Allman, V. Paxson, and W. R. Stevens, "TCP Congestion Control," IETF RFC 2581, Apr. 1999.
- [5] C. Partridge and T. Shepard, "TCP Performance over Satellite Links," *IEEE Network*, vol. 11, no. 5, Sept./Oct. 1997, pp. 44-49.
- [6] M. Allman, D. Glover, and L. Sanchez, "Enhancing TCP over Satellite Channels Using Standard Mechanisms," IETF RFC 2488, Jan. 1999.
- [7] K. Fall and K. Varadhan, Eds., "ns manual/ns Notes and Documentation," available with ns from <http://www.isi.edu/nsnam/>.
- [8] S. Floyd and T. R. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm," IETF RFC 2582, Apr. 1999.
- [9] M. Mathis, J. Mahdavi, and S. Floyd, "TCP Selective Acknowledgment Options," IETF RFC 2018, Oct. 1996.
- [10] J. C. R. Bennett, C. Partridge and N. Shectman, "Packet Reordering Is Not Pathological Network Behavior," *IEEE/ACM Trans. Net.*, vol. 7, no. 6, Dec. 1999, pp. 789-98.
- [11] S. Floyd et al., "An Extension to the Selective Acknowledgment (SACK) Option for TCP," IETF RFC 2883, July 2000.

- [12] M. Allman, "On the Generation and Use of TCP Acknowledgments," *ACM Comp. Commun. Rev.*, vol. 28, no. 5, Oct. 1998, pp. 4-21.
- [13] K. Varadhan, D. Estrin, and S. Floyd, "Impact of Network Dynamics on End-to-End Protocols: Case Studies in TCP and Reliable Multicast," Tech. rep. 98-672, Dept. of Comp. Sci., USC, Mar. 1998.
- [14] L. Wood et al., "IP Routing Issues in Satellite Constellation Networks," Accepted to appear in Special Issue on Internet Protocols over Satellite, *Int'l. J. Sat. Commun.*
- [15] T. R. Henderson, "Networking over Next-Generation Satellite Systems," Ph.D. dissertation, Comp. Sci. Div., UC Berkeley, Fall 1999.

#### BIOGRAPHIES

LLOYD WOOD (L.Wood@surrey.ac.uk) received his masters in electronic and electrical engineering from Loughborough University. He gained an M.Sc. in satellite communication engineering from the University of Surrey in 1995, after studying constellation topology at ENST Toulouse. He then joined what is now the Centre for Communication Systems Research (CCSR), where he has contributed to a number of European ACTS, Esprit, and IST projects. He maintains the Lloyd's satellite constellations Web resource while pursuing his Ph.D.

GEORGE PAVLOU (G.Pavlou@surrey.ac.uk) gained a Diploma in electrical engineering from the National Technical University of Athens. He obtained M.Sc. and Ph.D. degrees in computer science from University College London, where he acted as a senior research fellow and lecturer. He is currently professor of information networking in the Centre for Communication Systems Research at the University of Surrey, where he leads the activities of the networks research group. Over the last 10 years he has led a number of collaborative research projects in networking, with emphasis on management and control. Routing and quality of service are among his key research areas of interest. He has contributed to ISO and ITU-T standardization work.

BARRY EVANS (B.Evans@surrey.ac.uk) received his B.Sc. and Ph.D. degrees in electrical engineering and in microwave systems from the University of Leeds. From 1969 to 1983 he was British Telecom Lecturer and later Reader in Telecommunication Systems at the University of Essex. He was then appointed to the Alec Harley Reeves Chair of Information Systems Engineering at the University of Surrey, and in 1990 became the founding director of the post-graduate Centre for Satellite Engineering Research and of a spinoff company, Surrey Satellite Technology Ltd. Since 1996 he has been director of the Centre for Communication Systems Research at Surrey. He is the chief editor of the *International Journal in Satellite Communications*.

Although examining bulk FTP transfers clearly shows the limits of TCP's tolerance for multipath routing, future work would examine the impact of multipath routing on the shorter, burstier, transactional data transfers seen using HTTP.