# On-Board Satellite "Split TCP" Proxy

M. Luglio, *Member, IEEE*, M. Y. Sanadidi, *Senior Member*, M. Gerla, *Senior Member, IEEE*, and J. Stepanek, *Student Member, IEEE*

***Abstract*—Several satellite systems currently in operation or under development claim to support broadband Internet applications. In these scenarios, transmission control protocol (TCP) plays a critical role. Unfortunately, when used with satellite links, TCP suffers from a number of well-known performance problems, especially for higher data rates and high altitude satellites with longer delays. In response to these difficulties, the satellite and Internet research communities have developed a large gamut of solutions ranging from architectural modifications to changes in the TCP protocol. Among these, one approach requiring minimal modifications involves splitting the TCP connection in two or more segments with one segment connecting terrestrial nodes across the satellite network. In this paper, we consider an evolution of this idea: placing a TCP proxy on board the satellite that further subdivides the end-to-end connection into separate TCP connections between ground and space. We focus upon the efficient use of buffer resources on board the satellite, while at the same time enhancing TCP performance. We evaluate two TCP protocol versions, TCP NewReno and TCP Westwood. We consider various geosynchronous earth orbit satellite scenarios, with and without the split proxy, and with different channel error conditions (random errors, shadowing, etc). Using simulation, we show that an on-board proxy provides a number of distinct advantages and can enhance throughput up to threefold for both TCP New Reno and TCP Westwood, in some scenarios, with relatively modest on-board buffering requirements. The main contributions of this paper are: the on-board split proxy concept, the buffer management strategy, the use of a realistic "urban shadowing" model in the evaluation, and the extensive comparison of the recently announced TCP Westwood with the traditional TCP New Reno.**

***Index Terms*—Broadband, Internet, satellite, splitting.**

## I. Introduction

While the interest in ubiquitous broadband Internet access continues to expand, service providers struggle to deploy and operate broadband services because of limitations on technology, the size of target markets and of course, the costs involved with deploying and operating new advanced services.

Satellites offer a viable approach for delivering broadband service. Several satellite services either currently operating or in various stages of development [1]–[4] are targeting broadband applications and services. Such systems can in fact augment terrestrial services especially in remote areas, where terrestrial infrastructure remains prohibitively costly to deploy and operate. Alternatively, direct-broadcast satellites have achieved great success by providing direct broadband access to users requiring multicast and multiparty applications developed using the Internet protocol (IP)-based digital video broadcasting (DVB) IP and digital video subcommittee (DVS) return channel via satellite (RCS) standards [5], [6].

Of course, these systems must provide Internet services, which implies supporting IP in general and TCP in particular because of the world-wide-web and many other important client-server applications. Unfortunately, TCP suffers from a number of well-documented performance problems when used with satellite links. The problem grows more severe with longer delay, frequent errors, and large bandwidth.

In this paper, we extend the classical "split" TCP concept to a solution where the split occurs on board of satellite [7], [8]. Here, a *forwarding (proxy) agent* on the satellite maintains two separate split TCP connections for each end point of the TCP session. By splitting the TCP connection on board, we reap two very basic benefits: 1) we increase the speed of error recovery and 2) we reduce the propagation delay on each link. In the sequel, we will discuss the impact of these two features on performance in more detail.

## II. Satellites Characteristics Affecting TCP Performance

TCP provides a reliable stream of data when using any communication media, including satellites. Understanding some of TCP's features will help in assessing TCP performance in satellite environments. Here, we take a closer look at several properties of satellite links that can degrade TCP performance [9].

### A. Large Delays

The large propagation delay characteristic of satellites can severely impact TCP performance in several ways. First of all, large delay lengthens the duration of the "slow start" interval because the slow start process depends upon the round-trip time (RTT) of the connection [10], [11]. The TCP source sends up to one window of packets; TCP controls the sending packet rate by adjusting the size of the sender's window. Upon entering "slow start," TCP begins with a window of one segment and doubles this window approximately once for every RTT (assuming the receiver acknowledges each packet) up to a dynamically computed threshold. In fact, most implementations of TCP do not send a separate acknowledgment for every received packet and, thus, the window grows more slowly. In any case, the rate at which the window increases depends upon RTT. The larger RTT, the longer it takes to reach the threshold (see Fig. 1). Shortening this phase (by using a split connection) clearly improves performance. Moreover, the optimal size of the window, or the *pipe size*, is proportional to both RTT and the maximum sustainable

Manuscript received December 15, 2002; revised July 1, 2003 and September 20, 2003.

M. Luglio is with the Dipartimento di Ingegneria Elettronica, Università di Roma Tor Vergata, Rome 00133, Italy (e-mail: luglio@uniroma2.it).

M. Y. Sanadidi, M. Gerla, and J. Stepanek are with the Computer Science Department, University of California, Los Angeles, CA 90095 USA (e-mail: medy@ucla.edu; gerla@cs.ucla.edu; stepanek@cs.ucla.edu).

Digital Object Identifier 10.1109/JSAC.2003.819987

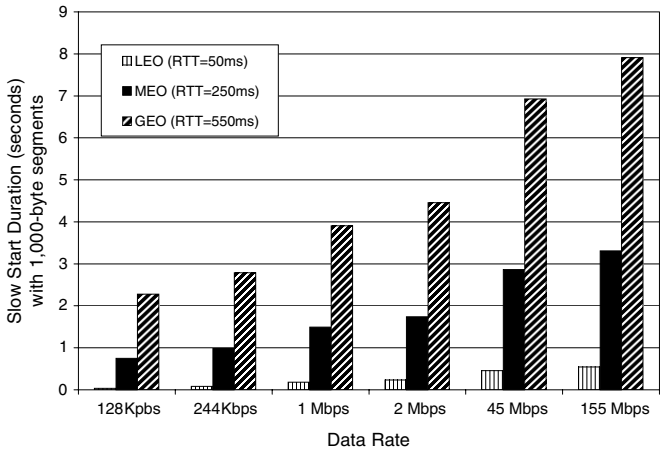
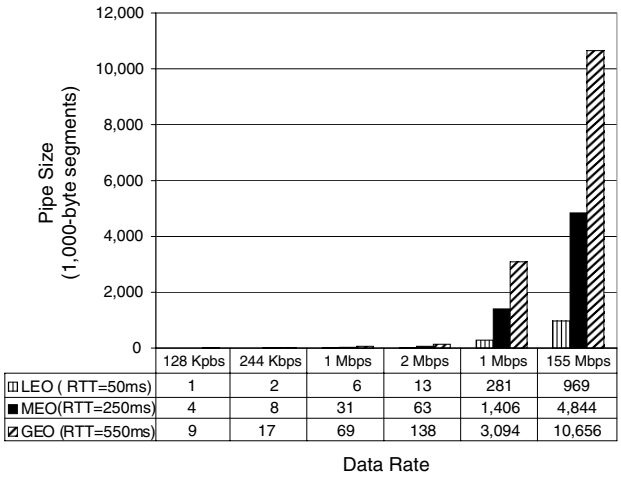



Fig. 1. Approximate duration of the slow start phase of TCP for various network conditions.




| | 128 Kpbs | 244 Kbps | 1 Mbps | 2 Mbps | 1 Mbps | 155 Mbps |
|---|---|---|---|---|---|---|
| LEO ( RTT=50ms) | 1 | 2 | 6 | 13 | 281 | 969 |
| MEO(RTT=250ms) | 4 | 8 | 31 | 63 | 1,406 | 4,844 |
| GEO (RTT=550ms) | 9 | 17 | 69 | 138 | 3,094 | 10,656 |

Fig. 2. Pipe size of TCP for various networks conditions.

data rate of the connection. As shown in Fig. 2, for high RTT and high data rate, the pipe size becomes very large. Larger windows further increase the length of slow start. Thus, scenarios requiring the provision of broadband service exacerbate this problem, especially when using geostationary constellations, as a consequence of both large bandwidth and large delays.

Larger RTTs also affect the rate at which the window grows during congestion avoidance. During this phase, the window increases by approximately one segment for every RTT. In the error-free case, this has little impact on performance. But in cases of more frequent losses, this behavior tends to decrease performance. Also, for these reasons connections with large RTTs may suffer when competing with connections with smaller RTTs. Competing connections with larger RTTs open their window more slowly and, thus, receive a smaller share of the available capacity.

### *B. Lossy Links*

The congestion-control mechanism of TCP relies upon lost packets as indicators of congestion. TCP congestion-control assumes packets are normally lost as a result of overflowing queues serving bottleneck links and infrequently lost for noncongestion reasons such as link errors and other types of corruption. While this assumption holds true for wired networks, it fails for many wireless links including satellites. Thus, when a TCP connection experiences relatively frequent losses from link-level errors, TCP performance suffers as a result. In fact, the impact of random errors upon performance increases with the size of the window. This occurs both as a consequence of and increased probability of multiple losses in one RTT—often leading to timeouts—and the increased duration of slow start. So for satellite links, with both higher losses and delays, random losses severely impact performance.

### *C. Previous Work*

The attractiveness of satellite access for broadband service has lead many in both the satellite and Internet communities to explore solutions to these problems. While carefully tuning TCP parameters, including the size of receive buffers (i.e., the advertised window) and the granularity of timers, improves performance in some cases [12], tuning fails to fully address the problems facing broadband networks with both high bandwidths and large delays. Solutions that go beyond tuning may involve changes to the protocol mechanisms [13]. In this case, modifications change the basic error-control and congestion-control strategies for improved performance, and often these solutions prove beneficial in a wider range of environments beyond satellite. Another class of solutions involves modifications and enhancement to the network infrastructure. In this case, intermediaries perform processing on behalf of TCP endpoints to the greater benefit of performance. One such technique involves subdividing connections into terrestrial and space segments, or splitting the connection.

As examples of the first type of scheme, consider changes to the TCP congestions-control mechanisms like NewReno [14] and Peach [15], or changes to error-control like SACK [16]. All have yielded promising results. Reference [17] summarizes many of these efforts in the context of satellites, but some of these approaches apply to nonsatellite environments as well. Along these lines, UCLA has developed a modification to the fast recovery algorithm called TCP Westwood [18]. A TCP Westwood sender continuously monitors the effective rate of its connection using ACK interarrival times. When this estimation falls below the current transmission rate as a result of packets lost at the bottleneck link, the Westwood sender corrects to match the estimated rate by adjusting its window and, thus, eliminates losses from congestion. By using estimated bandwidth instead of packet loss feedback, Westwood represents a significant departure from other congestion-control schemes. As previously discussed, this proves important for satellite links where lost packets are not a reliable indication of path congestion.

Since most congestion-control enhancements only involve modifications to TCP software at the source, destination or both, they remain transparent to routers as far as they do not require modifications in the software. Explicit congestion notification (ECN) [19] is one counter example to this rule. In ECN, intermediate routers provide TCP senders an alternative to divining congestion conditions from packet losses. This allows TCP senders to distinguish between losses caused by channel propagation effects, in which case the sender sees

no ECN feedback, and losses caused by congestion, in which case routers send positive ECN feedback. However, the cost of deploying ECN may overcome the practicality of using it since ECN requires expansive modification of routers—perhaps across the wired Internet. In contrast, a bandwidth estimation scheme like TCP Westwood requires a single TCP source to implement the change.

TCP Peach is a more recent TCP proposal that addresses satellite problems directly with protocol changes. TCP Peach replaces slow start and fast recovery with sudden start and rapid recovery. Both sudden start and rapid recovery use expendable "dummy" packets to probe the characteristics of the network; these packets are sent with a low priority so they do not create congestion. While the dummy packets are not recovered when lost, they are treated like regular data packets when the TCP sender receives acknowledgments for them, that is, the TCP sender increases the window in the manner of acknowledged data. As a result, the rate of window expansion increases when compared with slow start and fast recovery. When compared with ECN, TCP Peach has a weaker dependency upon routers, but to prevent dummy packets from introducing congestion, it requires that all routers process the precedence field of IP packets. Even so, the dummy packets may interfere with competing traffic at the link-level and routers must still process dummy packets if even to just drop them.

Beyond protocol enhancements, several architectural enhancements, or *middleware* techniques, address TCP's satellite problems. This approach has been generalized to the so-called "performance enhancing proxy" or PEP [20]. One of the PEP schemes involves segmenting, or "splitting" the TCP connection into satellite and nonsatellite segments [21]. Splitting makes use of TCP gateways located on the ground that maintain both TCP connections with other satellites gateways via satellite and TCP connections with other grounded nodes via ground networks. To communicate with each other, satellite gateways use separate connections that operate exclusively on the satellite segment. Between satellite gateways, splitting uses a transport protocol optimized for satellites. This transport might be a TCP protocol suitably modified for satellites or perhaps an altogether different protocol, for example, a reliable user datagram protocol (UDP)-based transport protocol again suitable for high performance in satellite environments.

Another PEP technique involves surreptitiously "capturing" a TCP connection and intelligently processing data packets and acknowledgments. Because the gateway in this case acknowledges data on behalf of the TCP receiver, this approach is commonly referred to as *spoofing* [22], [23]. Like splitting, spoofing divides the connection but in this case without the explicit knowledge of the endpoints. So spoofing forward and retransmits TCP packets introducing acknowledgments in a way that is not noticed by the endpoints.

None of the prescribed approaches, whether incorporating new protocol mechanisms or requiring enhanced infrastructure, represent a "best" overarching solution to TCP's problems. In fact, many of the approaches appear to complement one another. In reality, the utility of each approach depends upon many factors including the system's design and its intended use. The final systems may adopt a mixture of approaches depending upon their requirements, design, and cost limitations.

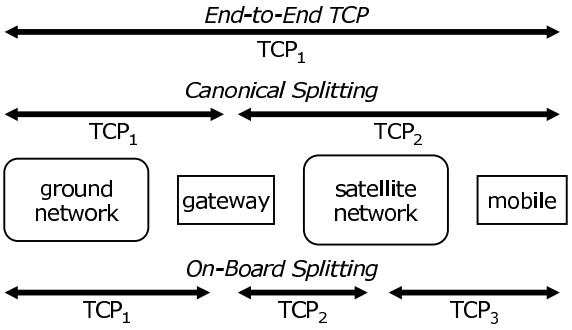



Fig. 3. An illustration of splitting TCP on board the satellite.

In this paper, we consider an architectural solution along the lines of splitting. Unlike the splitting described above, in this case, we propose using resources on board of the satellite in addition to grounded satellite gateways. We will show how this may both help with TCP performance and relieve some of the burden of ground nodes. Like normal splitting, this approach makes use of satellite gateways on the ground. But in this case, each satellite gateway operates separate TCP connections with the satellite instead of with each other and the satellite forward data between connections. Thus, the satellite itself acts as another application-level gateway between TCP endpoints.

The remainder of this paper is organized as follows. Section III describes the operation of the on-board forwarding agent. Following this, we then describe the simulation environment used to measure performance and memory requirements in Section IV and discuss the results of these experiments in Section V.

## III. Designing an On-Board TCP Forwarding Agent

As previously stated, the split TCP proxy architecture uses satellite gateways located on the ground in a manner consistent with traditional splitting. But in this case, each satellite gateway maintains a separate connection with the satellite rather than connections with each other (see Fig. 3). In the sequel, we will refer to the connection carrying data from the ground to the satellite as the *uplink connection* and the connection carrying data from the satellite to the ground as the *downlink connection*. In practice, each connection can send data in either direction and, thus, act as both uplink and downlink connections. To simplify our analysis, we first consider only the cases in which data packets flow in one direction from earth to space and back to earth again and, thus, acknowledgments flow in the reverse direction.

Subdividing the connection into earth-to-space connections results in a further reduction in windows sizes by approximately one-half. But as a consequence, the satellite must include an implementation of TCP that forwards packets between connections. So the reduction in window size comes at the expense of memory and processing on board the satellite. Moreover, like traditional splitting and other PEP approaches, on-board splitting violates the end-to-end semantics of TCP, that is, packets corrupted while in the gateway's buffer cannot be recovered

through TCP means. The satellite can help with this problem by acknowledging packets received only after they have been transmitted for the first time, that is, without waiting for the acknowledgment from the receiving node on the ground. To verify the integrity of the packet, the satellite uses the TCP checksum or some other method. As a result of these actions, only packets both lost on the downlink and subsequently corrupted in memory result in unrecoverable packets. In the semantics of TCP, this in effect represents a connection failure.

While the idea of on-board splitting appears promising, allocating on-board processing and memory resources remains a prime consideration to satellite designers, and while we believe the ability to maintain and upgrade software on board will continue to improve with advancements in satellite design, maintaining software on the ground will always remain easier than in space. As a result, using an on-board proxy as described here deserves careful consideration. We address this issue by listing some of the potential design advantages of using an on-board transport service beyond those already stated.

*1) TCP Interoperability:* A traditional splitting approach normally employs either a specialized TCP transport optimized for satellites or a non-TCP transport protocol requiring conversion. In contrast, the on-board transport improves performance using a standard TCP implementation. Specialized transports use an "optimistic" approach to congestion-control, and while these protocols may also benefit from splitting on board, under certain circumstances the satellite network may benefit from the more conservative TCP approach to congestion control. It is clear that scenarios involving only one satellite hop with circuit-switched access do not have any congestion problems. But multisatellite systems and packet switching on board may result in legitimate congestion and fairness problems. A future satellite system incorporating multiple-hops (and orbits) with packet switching and routing may benefit from standard TCP congestion-control mechanisms.

*2) Flexible Terminal Design:* Another advantage of using a standard TCP relates to the design of ground terminals. By using separate connections for the earth-to-space path, different terminals might use different transports according to their available resources; each ground terminal may use different protocols while maintaining interoperability. Consider a scenario in which a large satellite gateway serves many different types of terminals on the other side of the satellite. Some of these terminals may be *advantaged* in that they have optimized transport while others may be *disadvantaged* in that they lack resources and use an unoptimized TCP stack. By splitting the connection in space, these different terminal types can still communicate even though they use different transports. In this case, the disadvantaged terminal may also use the same TCP stack for all connections, roaming between wired, wireless, and satellite segments, and the disadvantaged terminal benefits when served by the gateway's optimized earth-to-space transport. Note that the satellite must implement different versions of TCP in order to support terminals with disparate resources. While this adds complexity to the satellite, supporting at least two TCP versions should be quite feasible on board a large satellite in geosynchronous earth orbit (GEO).

*3) Shadowing:* The problem of shadowing arises with GEOs deployed at higher latitudes, especially if there are mobile terminals moving behind obstacles. By inducing packet losses, shadowing severely impacts TCP performance. To counteract shadowing, the on-board forwarding agent uses independent error-control mechanisms for each earth-to-space connection and, thus, enhances the robustness of the transport. Normally, when both the uplink and downlink are susceptible to shadowing, a successful transmission requires that both links not be shadowed during such transmission. But using an on-board transport increases opportunities for success when both uplink and downlink suffer from intermittent shadowing. And since loss-recovery occurs across only the shadowed link and not across both uplink and downlink, terminals and gateways recover for shadowed losses more quickly when using an on-board transport.

*4) Multicast:* Finally, using on-board transport allows for combining optimized, congestion controlled TCP unicast on the uplink to the satellite with multicast on the downlink. This way, the multicast downlink takes advantage of the satellite's broadcast capabilities to serve multiple terminals and additional satellites. This service requires a gateway to handle the transition from TCP to multicast. In this case, the multicast may take advantage of the additional reliability provided by the on-board splitting. This adds additional reliability when compared with an end-to-end multicast session.

One may also argue that separate up and down link reliability can more easily be achieved by introducing an automatic repeat request (ARQ) link layer on up and down links. For example, the IEEE 802.11 media access control (MAC) protocol used in wireless local area networks (LANs) includes an ACK. This does achieve many of the previously described advantages, but by operating at the link layer, an ARQ mechanism of this sort suffers from the major drawback of forcing ALL flows to undergo ARQ. This includes, for example, real-time UDP-based transport that may include forward error correction (FEC) redundancy and, thus, does not require retransmissions. In fact, for real-time applications like VoIP link-level retransmissions work against the goals of the application. For these reasons, we find split TCP on the satellite an attractive solution, and explore thus using an on-board TCP proxy to enhance performance.

## IV. Simulation Models

To analyze the performance and memory requirement of on-board splitting, we have used the well-known ns-2 (network simulator) package [24]. This tool already supports satellites, but we have added several enhancements. These include an improved satellite channel model incorporating mobility and shadowing, and a model of our on-board forwarding agent that provides the splitting capability.

### A. Land Mobile Satellite (LMS) Channel

Signal shadowing remains the dominant factor affecting availability and performance of LMS systems. While fade margins and advanced transmission techniques compensate for multipath fading, they fail to mitigate blockage caused by shadowing. As a result, shadowing causes protracted high bit-error rates (BERs) and even temporary unavailability, and the degree of shadowing increases with lower satellite elevations, that is, higher latitudes in the case of GEO systems.

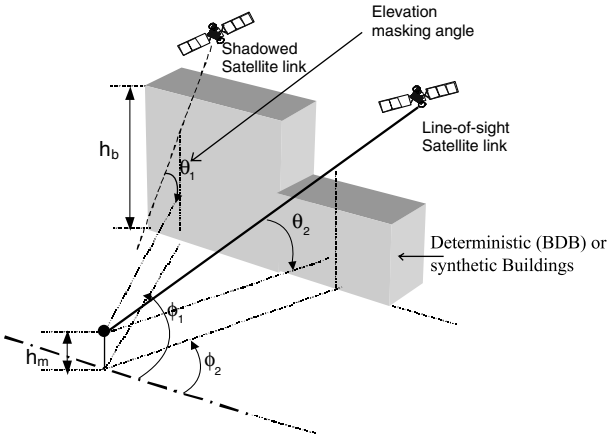



Fig. 4. Shadowed and LOS satellite links. Buildings can be obtained either from a BDB or through generating synthetic environment ($h_b$: building height, $h_m$: mobile height).

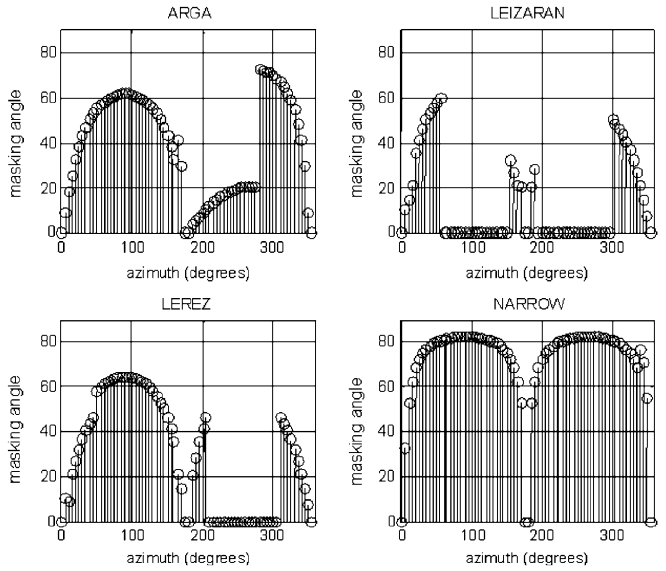



Fig. 5. Masking angles computed for four streets of Madrid.

Our simulation experiments use a physical-statistical LMS channel model (described in [25]) that incorporates geometrical projections of buildings surrounding the mobile terminal. In the model, height, and width statistical distributions describe these projections [26], [27]. The existence or absence of the direct ray defines the state of the channel as either line-of-sight (LOS), when a ray exists, or shadowed, when no ray exists. This model computation contains two parts as discussed below.

*1) Deterministic or Statistical Parameterization of Urban Environment:* The physical-statistical approach uses a canonical geometry for the environment traversed by a mobile receiver, typically a street as it is shown in Fig. 4. Depending on the satellite elevation, the buildings on both sides may block the view of the satellite as the terminal moves along the route. A building data base (BDB) stores the "canyon street" data characteristic of a particular type of urban environment. A receiver placed in a given position sees a particular skyline. For a fixed terminal or gateway, this skyline remains constant and shadowing occurs only if the satellite constellation moves; for GEO systems, this type of shadowing does not occur. For mobile terminals, however, the skyline continuously changes. In this case, a GEO becomes shadowed as the terminal moves behind buildings.

*2) Calculation of the Skyline Elevation Angles (Masking Angles):* Using the available building heights, the model incorporates computed elevation angles to the skyline. Generating the skyline requires a circular scan of 360° for each position along the mobile route in order to compute the elevation angle of the skyline, that is, an elevation-masking angle for every azimuth angle around the satellite terminal.

Fig. 5 shows an example of computed masking angles for four streets in Madrid. The channel model uses these elevation-masking angles to determine the link conditions for different azimuth angles of potentially visible satellites. If the satellite's elevation angle falls below the masking angle, the model assumes blockage. Otherwise, the terminal has LOS with the satellite.

Depending on its speed, the time the mobile needs to move along a given canyon street may be too short to obtain statistically significant TCP simulation times. In order to obtain significant simulation runs, the canyon streets were "stretched" by reversing and concatenating building information. Crossroads were also added as sections with no building obstructions. Changing the azimuth reference of the masking angle series, i.e., rotating the street relative to the satellites, allows simulating longer routes with different street orientations relative to the satellite.

Finally, while this canyon shaped street model does not consider the eventual presence of second-row buildings, we consider this effect negligible for the purposes of this study.

### B. On-Board Forwarding Agent

The primary focus of this study is the splitting of TCP connections using on-board satellite resources. In order to simulate this concept and measure performance and memory requirements, we enhanced the ns-2 simulator with a new TCP forwarding agent. Residing on the satellite, this agent receives data and acknowledgment from uplinks while sending, retransmitting, and acknowledging packets on downlinks.

More specifically, when the forwarding agent receives a valid packet on the uplink connection, it acknowledges this packet on the uplink connection and places the packet in the cache to be sent on the downlink. Consistent with the TCP sender, this packet remains in the cache until acknowledged by the downlink connection. At this point, the packet has successfully returned to earth and may be removed from the cache.

Note that an unrestrained sender can quickly overflow the cache space. As a result, the forwarding agent must control terrestrial senders. To accomplish this, the on-board forwarding agent uses TCP's advertised window mechanism. By progressively limiting the size of the window as the cache fills up, the forwarding agent also limits the rate of the sender. Our agent limits the advertised window on the uplink connection to one-third the total cache size, reducing this further as the cache fills. At the same time, the satellite limits the window on the downlink connection to one-third the total cache size, limiting the amount of unacknowledged data sent on the downlink connection. Thus, as packets become backlogged on the satellite,

the advertised window decreases, resulting in back pressure on the uplink. To support this type of back pressure, we modified the standard ns-2 implementations of TCP, allowing for an advertised window that changes over the life of the connection. Using a dynamic window allows for optimal use of the on-board cache, while ensuring that the cache does not overflow and result in dropped packets. By limiting windows to one-third the total cache space, the cache has room for one window of data from the uplink connection, while storing one window of unacknowledged data on the downlink connection and leaving one window of additional space for the feedback of back pressure to take effect.

## V. Simulation Results

The results of our simulations fall into three broad categories.

1) Single-Hop Unshadowed GEO—Two terminals connected via single GEO satellite.
2) Multihop Unshadowed Double GEO—Two terminals connected via two GEO satellites. The GEOs communicate using an intersatellite link (ISL).
3) Single-Hop Source-Shadowed GEO—As in the first case, but with a mobile TCP sender affected by shadowing.

In these experiments, the single-hop scenarios include two terminals with the TCP sender in New York and the TCP receiver in San Francisco. In this case, the GEO satellite resides at $-96^\circ$ longitude. In the multihop double-GEO case, the TCP sender lies in Rome with the TCP receiver in Los Angeles. In this case, one GEO resides at $-95^\circ$ (where the splitting occurs, if any) and the other at $0^\circ$ longitude.

The shadowed scenarios use the LMS channel model as described previously. Here, the mobile terminal moves at 2 m/s through an urban "canyon." In all scenarios, including shadowing, both uplinks and downlinks suffer from uniformly distributed BER of $10^{-6}$, $10^{-7}$, or $10^{-8}$. These bit errors were applied independently to both uplink and downlink for both data and acknowledgments.

In each scenario, we evaluate and compare the throughput of TCP NewReno and TCP Westwood, both with the NewReno enhancements. For the unshadowed cases, the throughput values were averaged over 100 file transfer protocol (FTP) sessions each lasting 31 s. Because shadowing introduces greater statistical variation, the throughput for the shadowed cases was averaged over 1000 FTP sessions each also lasting 31 s.

All of the TCP experiments used 1500-byte packets. TCP window sizes were not limited by the buffer space of the source or sink nodes.

*1) Single-Hop Unshadowed GEO:* Fig. 6 includes results for the GEO case without shadowing. $\text{Link capacity} = 1$ Mb/s. $\text{BER} = 10^{-6}$. Here, losses occur merely as a result of the uniformly distributed BER (no shadowing). In part due to the low capacity of the link, TCP achieves reasonable performance even when faced with the relatively high error rate of $10^{-6}$. TCP Westwood achieves 40% of channel capacity; TCP NewReno, less than 20%. These values are consistent with previously published results [18]. By introducing the split proxy on board, TCP performance increases significantly, more in NewReno than in Westwood (as the former has more to gain). Note that performance increases with the size of the forwarding cache until reaching a *saturation point*, beyond which the cache does not appear to present a limiting factor.

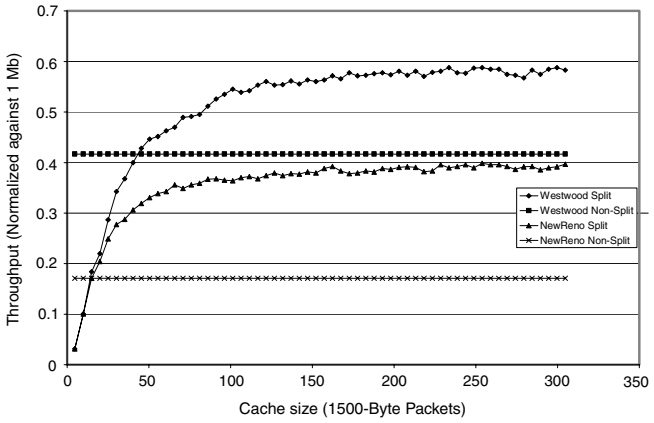



Fig. 6. TCP throughput versus cache size for the unshadowed GEO with $10^{-6}$ BER and 1 Mb/s capacity.

The relationship between performance and the size of the cache relates to the pipe size of the link. Note that in this scenario, both the uplink and downlink have 1 Mb/s capacity and use 1500-byte packets. Assuming a GEO one-way delay of 125 ms, the pipe size for each link becomes $1 \cdot 10^6$ b/s$/(8 \cdot 1500$ bytes/packet$) \cdot 2 \cdot 0.125$ s, or 21 packets. Recall that we use a buffer management strategy that advertises only one-third of the total cache space. In our case, performance levels off approximately after the size of the cache exceeds three times the pipe size, or 63 TCP segments.

The improvement of split TCP is due to two key factors 1) the halving of propagation delay RTT on each TCP segment and 2) the halving of loss probability on each segment relative the loss probability of the end-to-end connection. As loss and RTT combined have major impact on TCP performance, the improvement does not come as surprise. The improvement indeed is of the magnitude predicted by theory in [18].

The fact that performance is worse than the no-split case when the cache is much below the saturation value deserves some commentary. If there are not sufficient buffers in the cache, copies of the packets cannot be saved for retransmission in case of loss—thus performance rapidly deteriorates. From a practical standpoint, a 100-packet buffer is small enough to pose no implementation problems anyway. Note that the total cache memory required is independent of the number of TCP connections going through the satellite as the aggregate number of outstanding packets is a function of channel capacity only (it is independent of the number of flows).

Fig. 7 shows the TCP throughput versus cache size for a higher capacity link, i.e., 20 Mb/s. In this case, the increased capacity produces a higher pipe size. As shown in [18], increasing pipe size causes a higher degradation for the same BER. Splitting the connection on board reduces the pipe size, as well as BER and makes a much greater impact. This is again consistent with the results reported in [18].

Fig. 8 summarizes the simulations results for the unshadowed single-hop GEO case. For the connections split on board, these results report the average throughput when the forwarding agent used a "saturated" cache size, that is, a cache large

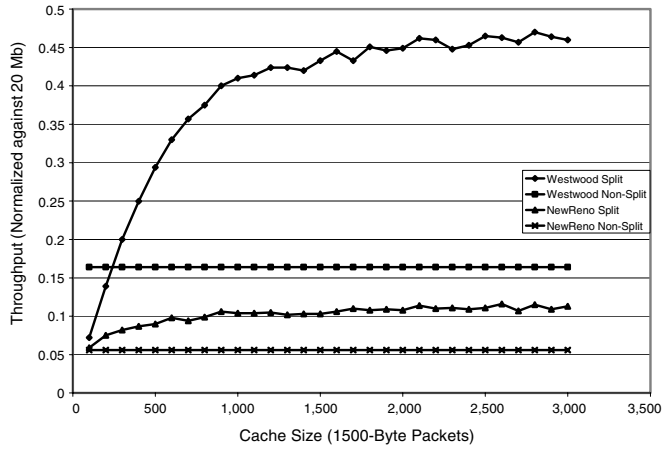



Fig. 7. TCP throughput versus cache size for the unshadowed GEO with $10^{-7}$ BER and 20 Mb/s capacity.

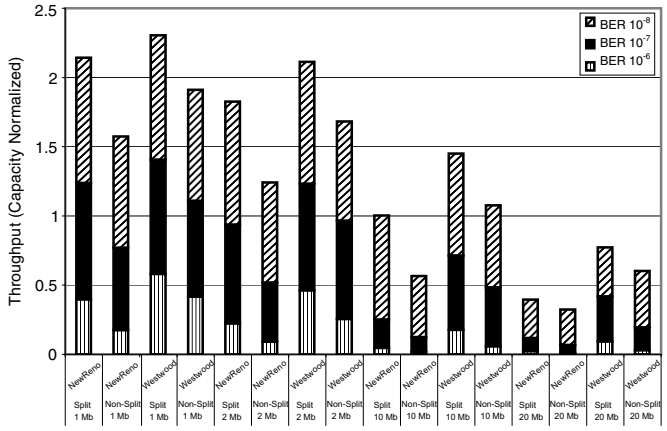



Fig. 8. Summary of TCP performance for the unshadowed GEO scenario.

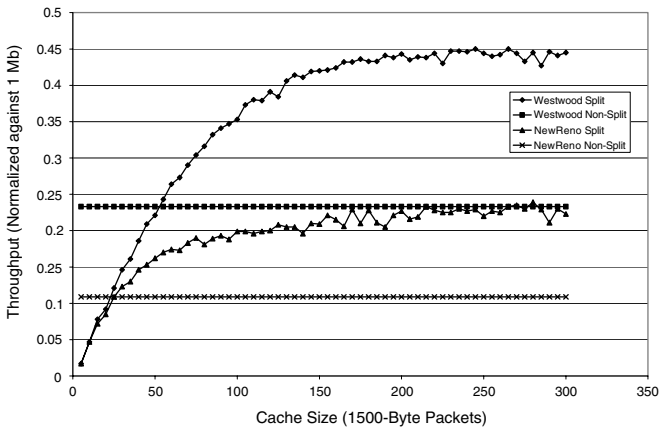



Fig. 9. TCP throughput versus cache size for the unshadowed double-GEO with $10^{-6}$ BER and 1 Mb/s capacity.

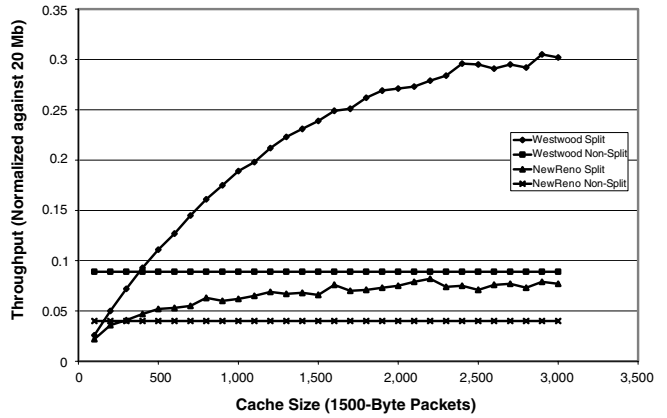



Fig. 10. TCP Throughput versus cache size for the unshadowed double-GEO with $10^{-7}$ BER and 20 Mb/s capacity.

enough for the performance to stabilize as seen in Fig. 6. From this chart, a clear trend emerges in which splitting generally enhances performance and greatly enhances performance with increasing bandwidth and error rates. In some cases, splitting achieves over a threefold increase. Overall, TCP Westwood benefits less from splitting (but consistently performs much better) than NewReno.

*2) Multihop Unshadowed Double GEO:* Fig. 9 shows TCP throughput for the double-GEO scenario. Here, due to the increased propagation delay, the connection achieves lower throughput overall. Much larger windows are required and, thus, performance only levels off with much larger caches. Splitting also dramatically increases performance for saturated cache sizes. Fig. 10 shows a similar picture for a higher data rate. Using 20 Mb/s links with a double-hop configuration results in the largest windows considered in this study. Overall performance is quite poor, but on-board splitting results in over a threefold improvement.

Fig. 11 summarizes the results for the double-hop GEO case. For longer delays, TCP Westwood increases performance relative to NewReno substantially more than in the single-GEO case. This holds especially true for large bandwidths.

*3) Single-Hop Source Shadowed GEO:* Figs. 12 and 13 show the throughput of TCP versus the buffer-size with 1 and 20 Mb/s, respectively. In this case, shadowing impacts the

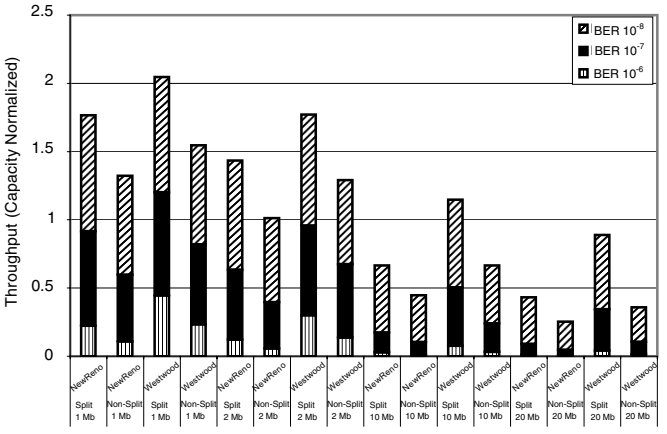



Fig. 11. Summary of TCP performance for the unshadowed double-GEO scenario.

mobile terminal acting as a TCP source. Shadowing results in a significant drop in overall throughput. For the lower link capacity, TCP Westwood appears more resilient to shadowing than NewReno, and while the high-capacity scenarios require significantly more buffer space to achieve optimal performance, splitting once again makes a significant impact. A somewhat surprising result is the fact that with 20 Mb/s channel speed,

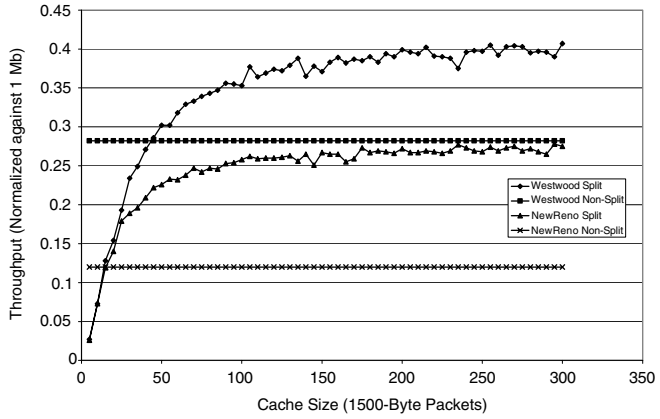



Fig. 12. TCP throughput versus cache size for the source-shadowed GEO with $10^{-6}$ BER and 1 Mb/s capacity.

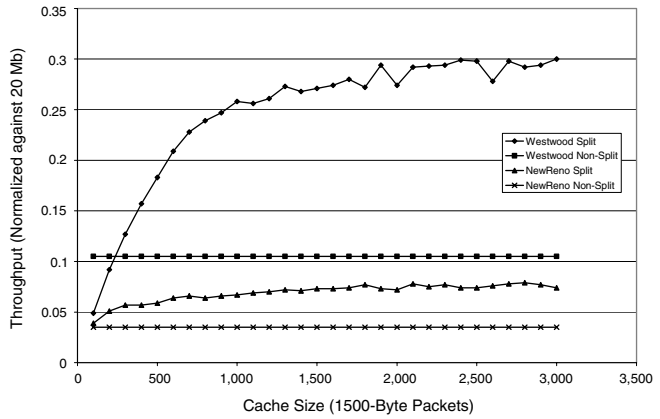



Fig. 13. TCP throughput versus cache size for the source-shadowed GEO with $10^{-7}$ BER and 20 Mb/s capacity.

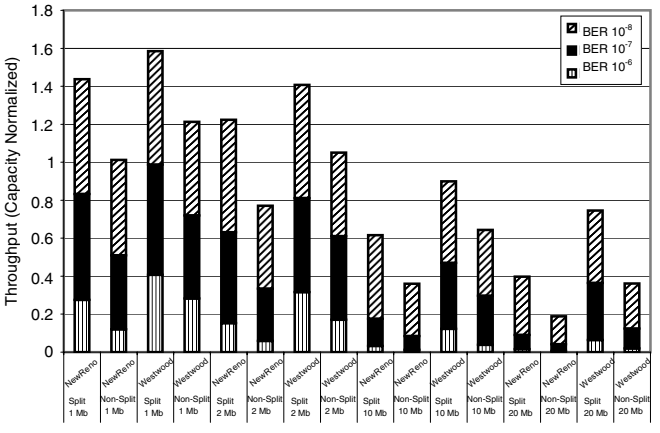



Fig. 14. Summary of TCP performance for the source-shadowed GEO.

TCP Westwood improves three times with splitting, while TCP NewReno improves only two times. This anomaly (with respect to prior observations) may be in part attributed to the higher statistical error and packet loss fluctuations due to shadowing induced by mobility.

Fig. 14 summarizes the results for the source-shadowed terminal, that is, when the TCP sender moves and is affected by shadowing.

## VI. Conclusion

In this paper, we have examined using a splitting TCP proxy on board the satellite in order to improve the performance of TCP. For this purpose, we have developed and studied (via simulation) a simple model of a TCP forwarding agent. This agent uses TCPs advertised window to perform backpressure on the TCP uplink segment, which results in efficient use of valuable satellite resources by reducing the chance of buffer overflow once the packet arrives at the satellite. Overall, we found that "splitting" the TCP connections into separate uplink and downlink component yields significant performance improvements. This is consistent with our expectations, as the subdivision of the path into two segments reduces both the relative error rate (BER) and propagation delay (RTT) for each segment. Such joint reduction is known to be highly beneficial to throughput efficiency. The price to pay is on-board store and forwarding—this is, however, a given in most contemporary satellites—and some extra cache buffers. The amount of buffering required, however, is fairly modest, and is independent of number of connections supported. If the connection is affected by mobility induced shadowing and goes over multiple hops, the buffering requirements in the cache typically increase; on the other hand, the on-board split TCP provides even greater improvement.

Research is now under way to compare the on-board "split TCP" solution with more conventional, gateway based split TCP solutions. Also, under study is the fair sharing across multiple TCP connections multiplexed on the same satellite channel.

## References

[1] [Online]. Available: http://www.spaceway.com/
[2] [Online]. Available: http://www.astra.lu/
[3] [Online]. Available: http: //www.eutelsat.com/
[4] [Online]. Available: http://www.alespazio.it/program/tlc/eurosk/eurosk.html
[5] M. R. M. Rizk and M. M. Ayoub, "MVDDS network layers adapted with DVB-RCS as a terrestrial RC solution," in *Proc. 19th National Radio Science Conf. (NRSC 2002)*, 2002, pp. 583–591.
[6] J. Neale, "Symposium on future satellite communications for global IP and ATM networking market trends and technological developments for DVB-RCS," in *Proc. IEEE Global Telecommunications Conf. (GLOBECOM'01)*, vol. 4, 2001, pp. 2789–2791.
[7] M. Luglio, J. Stepanek, and M. Gerla, "TCP performance using splitting over the satellite link," in *Proc. 8th Ka Band Utilization Conf.*, Baveno, Italy, Sept. 25–27, 2002, pp. 45–52.
[8] J. Stepanek, A. Razdan, A. Nandan, M. Gerla, and M. Luglio, "The use of a proxy on board the satellite to improve TCP performance," in *Proc. IEEE Global Telecommunications Conf. (GLOBECOM'02)*, Nov. 2002, pp. 2950–2954.
[9] C. Partridge and T. J. Shepard, "TCP/IP performance over satellite links," *IEEE Network*, vol. 11, pp. 44–49, Sept.–Oct. 1997.
[10] V. Jacobson, "Congestion avoidance and control," presented at the SIGCOMM'88, Stanford, CA, 1988.
[11] W. Stevens, "TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms," Internet Society, Reston, VA, RFC 2001, Jan. 1997.
[12] M. Emmelmann, "Effects of advertised receive buffer size and timer granularity on TCP performance over erroneous links in a LEO satellite network," in *Proc. IEEE Global Telecommunications Conf. (GLOBECOM'02)*, vol. 2002, Nov., pp. 2955–2958.
[13] M. Allman, D. Glover, and L. Sanchez, "Enhancing TCP over satellite channels using standard mechanism," Internet Society, Reston, VA, RFC 2488, Jan. 1999.
[14] S. Floyd and T. Henderson, "The NewReno modification to TCP's fast recovery algorithm," Internet Society, Reston, VA, RFC 2582, Apr. 1999.

[15] I. F. Akyildiz, G. Morabito, and S. Palazzo, "TCP-peach: A new congestion control scheme for satellite IP networks," *IEEE/ACM Trans. Networking*, vol. 9, no. 3, pp. 307–321, June 2001.
[16] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP selective acknowledgment options," Internet Society, Reston, VA, RFC 2018, Oct. 1996.
[17] M. Allman, Ed., "Ongoing TCP Research Related to Satellites," Internet Society, Reston, VA, RFC 2760, Feb. 2000.
[18] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP westwood: Bandwidth estimation for enhanced transport over wireless links," presented at the MOBICOM 2001, Rome, Italy, July 2001.
[19] K. Ramakrishnan and S. Floyd, "A Proposal to Add Explicit Congestion Notification (ECN) to IP,", RFC 2481, Jan. 1999.
[20] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations," Internet Society, Reston, VA, RFC 3135, June 2001.
[21] T. Henderson and R. Katz, "Transport protocols for internet-compatible satellite networks," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 345–359, Feb. 1999.
[22] J. Ishac and M. Allman, "On the performance of TCP spoofing in satellite networks," in *Proc. IEEE Military Communications Conf. (MILCOM 2001)*. McLean, VA, Oct. 28–31, 2001, pp. 700–704.
[23] A. Bakre and B. R. Badrinath, "I-TCP: Indirect TCP for mobile hosts," , Vancouver, BC, Canada, May 1995, pp. 136–143.
[24] Network Simulator (NS-2) [Online]. Available: http://www.isi.edu/nsnam/ns/
[25] P. Loreti, M. Luglio, R. Kapoor, J. Stepanek, M. Gerla, F. Vatalaro, and M. A. Vazquez-Castro, "LEO satellite systems performance with TCP-IP applications," presented at the IEEE Military Communications Conf. (MILCOM 2001), session U24, McLean, VA, Oct. 28–31, 2001.
[26] ——, "Throughput and delay performance of mobile internet applications using LEO satellite access," in *Proc. Int. Symp. 3G Infrastructure and Services*, Athens, Greece, July 2–3, 2001, pp. 68–73.
[27] M. Gerla, R. Kapoor, J. Stepanek, P. Loreti, M. Luglio, F. Vatalaro, and M. A. Vazquez-Castro, "Satellite systems performance with TCP-IP applications, tyrrhenian international workshop on digital communications," in *Proc. IWDC 2001*, Taormina, Italy, Sept. 17–20, 2001, pp. 76–90.

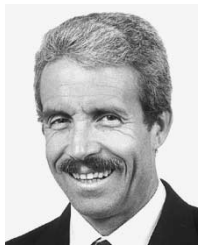
**Michele Luglio** (M'96) received the Laurea degree in electronic engineering from the University of Rome "Tor Vergata," Rome, Italy, and the Ph.D. degree in telecommunications in 1994.

From August to December 1992, he worked as a Visiting Staff Engineer at Microwave Technology and Systems Division of Comsat Laboratories, Clarksburg, MD. He is a Research and Teaching Assistant at the University of Rome "Tor Vergata,"since October 1995, where he works on designing satellite systems for multimedia services both mobile and fixed, in the frame of projects funded by EC, ESA, and ASI. He taught signal theory and collaborated in teaching digital signal processing and elements of telecommunications. In 2001 and 2002, he was Visiting Professor in the Computer Science Department, University of California, Los Angeles (UCLA), where he taught the satellite networks class. Now, he teaches satellite telecommunications and signals and transmission.

Dr. Luglio received the Young Scientist Award from the International Symposium on Signals, Systems, and Electronics (ISSSE) in 1995.

**M. Yahya "Medy" Sanadidi** (S'83–SM'03) was born in Damanhour, Egypt. He received his high school diploma from College Saint Marc, Alexandria, Egypt, the B.Sc. degree in electrical engineering (computer and automatic control section) from the University of Alexandria, Alexandria, Egypt, and the Ph.D. degree in computer science from the University of California, Los Angeles (UCLA), in 1982.

He is currently a Research Professor in the Computer Science Department, UCLA. As co-principal investigator on National Science Foundation (NSF) sponsored research, he is leading research in modeling and evaluation of high-performance IPs. He teaches undergraduate and graduate courses at UCLA on queueing systems and computer networks. He was a Manager and Senior Consulting Engineer with Teradata/AT&T/NCR, Segundo, CA, from 1991 to 1999 and led several groups responsible for performance modeling and analysis, operating systems, and parallel query optimization. From 1984 to 1991, he held the position of Computer Scientist with Citicorp, Santa Monica, CA, where he pursued R&D projects in wireless metropolitan area data communications and other networking technologies. In particular, from 1984 to 1987, he lead the design and prototyping of a wireless MAN for home banking and credit card verification applications. From 1981 to 1983, he was an Assistant Professor in the Computer Science Department, University of Maryland, College Park, where he taught performance modeling, computer architecture, and operating systems, and was Principal Investigator for NSA sponsored research in global data communications networks. He has consulted for industrial concerns, has coauthored conference and journal papers, and holds two patents in performance modeling. He participated as reviewer and as program committee member of professional conferences. His current research interests are focused on congestion control and adaptive multimedia streaming protocols in heterogeneous (wired/wireless) networks.



**Mario Gerla** (M'75–SM'01) received the graduate degree in electrical engineering from Politecnico di Milano, Milano, Italy, in 1966 and the M.S. and Ph.D. degrees in computer science from the University of California, Los Angeles (UCLA), in 1970 and 1973, respectively.

From 1973 to 1976, he was a Manager at the Network Analysis Corporation, Glen Cove, NY, where he was involved in several computer network design projects for both government and industry, including performance analysis and topological updating of the ARPANET under a contract from DoD. From 1976 to 1977, he was with Tran Telecommunication, Los Angeles, CA, where he participated in the development of an integrated packet and circuit network. Since 1977, he has been on the Faculty of the Department of Computer Science, UCLA. His research interests include the design, performance evaluation, and control of distributed computer communication systems and networks.

**James Stepanek** (S'03) received the B.S. degree in computer science from Harvey Mudd College, Claremont, CA, in 1994, and the M.S. degree in computer science in 2001 from the University of California, Los Angeles (UCLA), where he is currently working toward the Ph.D. degree.

Currently, he is member of Technical Staff in the Computer Systems Research Department, The Aerospace Corporation, El Segundo, CA. His research interests include wireless and satellite networks.