Satellite-3G Hybrid Networks: Impact of ACM and ARQ on TCP Performance

Nedo Celandroni, Erina Ferro, Giovanni Giambene, Mario Marandola

Original scientific paper

Abstract— **The adoption of satellite systems in providing broadband transmissions to mobile users such as trains, buses and vans is expected to be an interesting solution. The scenario we considered refers to a hybrid network architecture, where a geostationary satellite forward link and a terrestrial 3G return link are used in order to exploit both the high bandwidth of a satellite channel and the lower propagation delay of a terrestrial path. The resulting round-trip delay is much shorter than that one experienced by using both the forward and return link via satellite. This is particularly appealing for overcoming the TCP efficiency degradation in high delay-bandwidth product and error prone channels. In this hybrid scenario, we used simulation results to compare the goodput of four of the most popular TCP variants, in the presence of a GOOD-BAD satellite channel, as the one experienced by mobile users. We applied an *Adaptive Coding and Modulation* (ACM) technique as well, and studied its impact on TCP efficiency, when used both alone and in cooperation with an *Automatic Repeat reQuest* (ARQ) scheme of the *Selective Repeat* (SR) type with low persistency. Results obtained indicate that this hybrid architecture is advantageous for TCP transmissions in terms of average goodput, and that ACM is effective only if it is jointly used with ARQ schemes.**

Index Terms— **Hybrid networks, TCP, ACM, ARQ.**

I. Introduction

Satellite communication systems represent an adequate solution for providing high bit-rate services to mobile users over wide areas. In fact, satellites offer a set of advantages such as: (*i*) easy fruition of both broadcast and multicast high bit-rate multimedia services; (*ii*) backup communication services for third-generation (3G) cellular users on a global scale; (*iii*) efficient support of high-mobility cellular users (e.g., users on trains, planes, etc.). For many isolated areas on the earth, satellites are the only solution to be connected to local Internet service providers. Indeed, satellite communications represent an integrated component of 3G terrestrial cellular systems [1].

We assume operating with a *Geostationary Earth Orbit* (GEO) satellite. The high altitude of this type of satellite requires high transmission power levels and induces high propagation delays, which significantly affect the behavior of protocols at layers 2, 3, and 4 of the OSI stack. In addition, the *Effective Isotropic Radiated Power* (EIRP) levels of mobile users are limited by the reduced size of terminals' antennas. Thus, it is quite difficult to have a return link via satellite with adequate reliability. This is the reason why we investigated the potential of a hybrid system architecture, where a terrestrial cellular system and a satellite one co-operate in providing multimedia Web contents to users on trains, buses or vans. This is realized via FTP/HTTP applications, which make use of TCP. We refer to *forward path* as the main data flow (TCP segments) from the satellite *Earth Station* (ES), where the TCP sender is located, towards the *Mobile Terminal* (MT) that receives the TCP-based flow. The *reverse path* consists of acknowledgements and channel quality indications from the MT towards the ES. TCP data are routed from/to the Web server to/from the ES.

In rural areas, the *Line-of-Sight* (LoS) between satellite and MT is almost always uninterrupted; thus, we used the satellite link for forward path and the terrestrial 3G cellular system for reverse path. MT's receive high-bit-rate via satellite and send feedback traffic through the terrestrial cellular segment, which has a lower, but adequate, capacity. In addition to this, in both the urban and sub-urban scenarios, we exploited the possibility of addressing the forward path through the terrestrial segment as well, since the direct link to the satellite may be obstructed most of the time. An intelligent agent in the ES is able to switch the forward path through either the satellite or the terrestrial cellular network.

It is well-known that classic versions of the TCP protocol may have low performance in terms of throughput over GEO satellite links [2] due to: (*i*) high propagation delays, which entail a slow increase in the *congestion window* (*cwnd*), regulating the amount of data injected in the network; (*ii*) possible frequent errors in the radio channel, which cause halving *cwnd* [3], in that TCP assumes that packet losses are due to congestion. Excessive delays in correctly transmitting a packet (a "segment", in TCP terminology), due to bad channel conditions, may entail a TCP time-out, thus forcing TCP to reduce *cwnd* to the minimum value (one or two segments in most cases). References [4]–[9] present some of the many proposals that have been done in order to improve TCP throughput in terrestrial and satellite mobile communication systems.

In our scenario, forward path transmissions occur by means of a *Digital Video Broadcasting-Satellite* (DVB-S) [10] air interface, which we modified to better support adaptability to radio channel variations; reverse path employs the terrestrial

Manuscript received May 1, 200; revised September 15, 2007. This research was supported by the SatNEx network of excellence (FP6, IST-027393). This paper only in part derives from a paper presented by these authors at Globecom 2006, 27 November – 1 December 2006, San Francisco, USA ("TCP Performance in a Hybrid Satellite Network by using ACM and ARQ").

Nedo Celandroni and Erina Ferro are with ISTI – Consiglio Nazionale delle Ricerche (C.N.R.), Area della Ricerca di S. Cataldo, Pisa, Italy.

Giovanni Giambene and Mario Marandola are with CNIT – Università degli Studi di Siena, Siena, Italy. Correspondence to Giovanni Giambene (e-mail: giambene@unisi.it).

1845-6421/07/7026 © 2007 CCIS

Universal Mobile Telecommunication System (UMTS) [1] or a similar cellular technology. Our interest is in studying TCP performance in a hybrid architecture that reduces the *Round Trip Time* (RTT) with respect to a system that employs a return channel via satellite (DVB-RCS standard). Hence, this approach combines high forward path bit-rates with reduced RTT's, which allow us to adopt an *Automatic Repeat re-Quest* (ARQ) scheme of *Selective Repeat* (SR) type with low persistency. This feature is very effective in overcoming the TCP efficiency degradation in high *Delay-Bandwidth Product* (DBP) and error-prone channels. On the satellite air interface, we also adopt an *Adaptive Coding and Modulation* (ACM) scheme, which is a technique supported by DVB-S2 [11], in order to cope with fluctuations in channel conditions. In particular, when channel conditions are in BAD state, we need to adopt a powerful *Forward Error Correction* (FEC) code (with a consequent reduced information bit-rate) in order to avoid excessive errors; on the other hand, when the satellite channel is in GOOD state, a higher bit-rate is exploited with a lower FEC protection. The alternation of GOOD and BAD channel states introduces a significant fluctuation in the capacity available at the TCP level; this causes some problems with the congestion control mechanisms, which are interesting to investigate.

The literature is very rich in the analysis of TCP behavior in conjunction with ARQ. Indeed, only few papers deal with high DBP channels. One of the most recent papers [12] makes an interesting analysis of SR ARQ over channels characterized by hidden Markov models by using matrix signal flow graphs. However, the results presented in that paper are not relative to high DBP channels and refer to very high average packet loss rates. In our paper, we preferred using ACM in order to deal with not too high packet loss rates, which is demonstrated to be an efficient solution [13],[14]. At the best of the authors' knowledge no papers deal with both ACM and ARQ effects on TCP behavior in a satellite-based scenario. The interest of our work is in a *cross-layer approach* that combines physical layer, link control, and transport layer congestion control mechanisms. In particular, we considered various combinations of techniques, such as ACM, ARQ and different TCP versions, by analyzing the contribution given by each of them to the improvement in performance.

Section II describes the TCP variants we compared each other, and Section III details the characteristics of the system proposed. Simulation results are presented and discussed in Section IV, while our final considerations are summarized in Section V.

II. TCP Variants Considered

The classic TCP congestion control mechanism is relative to the Reno version, which is based on four phases [3]: Slow-Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery. With the Reno version, it is possible to detect a lost segment through duplicate *acknowledgements* (ACKs), thus avoiding the occurrence of a *Retransmission Time Out* (RTO), by entering fast recovery/retransmission phases. TCP Reno can only recover one packet loss per RTT. In case of multiple losses in the same RTT, the sender must wait for the RTO expiration in order to determine the segment to be retransmitted. In fact, TCP adopts a cumulative ACK scheme and the highest segment received in sequence is acknowledged. The fast recovery phase is ended upon the correct reception of the lost segment, only if all segments that follow the lost one are correctly received. After RTO expiration, the sender uses the slow start algorithm to restart the transmission with *cwnd* at the minimum value.

In our study, we considered the following TCP variants:

- NewReno – This version can recover multiple packets lost from one window of data by using the partial ACK mechanism, which attempts to avoid RTO expiration [4]. The *Impatient* (Imp) variant of NewReno resets RTO only after the first partial ACK is received. In this case, if a large number of packets is dropped from a window of data, the TCP RTO ultimately expires so that the TCP sender will invoke slow start. In contrast, the *Slow-but-Steady* (SS) variant of NewReno resets RTO after each partial ACK. In this case, for a window with a large number of dropped packets, the TCP sender retransmits at most one packet per RTT. Hence, the RTO expiration is avoided, but the recovery phase may last for a long time.
- SACK – This is an option used to obviate the possible poor TCP performance when multiple packets are lost from one window of data [5]. It consists of a *Selective Acknowledgment* (SACK) mechanism, combined with a selective repeat retransmission policy. The receiving TCP sends back SACK packets informing the sender about the data received. The sender can then retransmit the missing segments only. Errors are typically recovered before RTO occurs; thus, the slow start phase is avoided. Note that SACK also includes both fast recovery and fast retransmit algorithms, as the TCP Reno version.
- Westwood – This scheme introduces a modification to the fast recovery algorithm, called "Faster Recovery" [8]. Differently from the standard TCP version (i.e., NewReno), which halves the congestion window after three duplicate ACKs and blindly fixes the *slow start threshold* (*ssthresh*) equal to the congestion window, TCP Westwood attempts to select *ssthresh* and *cwnd* that are more consistent with the actual available bandwidth, which is estimated by means of the rate of arriving ACKs.

III. The Hybrid Scenario

In our system, each MT supports both terrestrial 3G and broadband (DVB-S) satellite radio interfaces. The system architecture is depicted in Fig. 1; a GEO satellite ES receives data from a terrestrial server, and re-transmits the received data by using TCP with high bit-rate directly to an MT that is displaced in a rural area. TCP ACKs (each one composed by 20-byte TCP header[1] and 20-byte IP header) are sent to the ES at a lower bit-rate (i.e., 144 kbit/s) via a terrestrial 3G segment.

[1]Note that in the SACK case, the TCP header is longer than 20 bytes.

TABLE I
ACM PARAMETERS.

State	Code rate (RS + convolutional code without interleaver)	Channel bit-rate, Information bit-rate, Mod. scheme	Minimum C/N_0 [dB]	Minimum E_b/N_0 [dB]	Max FER	Max SER
GOOD	(204, 188) and 3/4	2.048 Mbit/s, 1.416 Mbit/s, QPSK	66.9	5.4	10^{-5}	8×10^{-5}
BAD	(204, 188) and 1/2	1.024 Mbit/s, 0.472 Mbit/s, BPSK	59.7	2.95	10^{-4}	8×10^{-4}

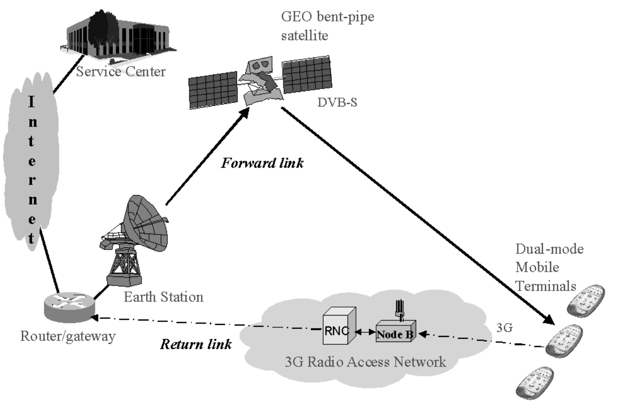

Fig. 1. The hybrid system scenario considered.

A. Physical Layer Characteristics

Transmissions on the forward link occur according to two different modalities that depend on the satellite *Channel Quality Estimation* (CQE), performed by the MT and sent back to the ES. In particular, we assume the radio channel is in a GOOD/BAD state according to the signal-to-noise ratio above/below a certain threshold. The GOOD-BAD channel model detailed in [15] seems an appropriate choice for our MT's; the state is related to good/bad propagation conditions (transmissions in LoS or not). Permanence times in each of the two states are exponentially distributed and their mean values, T_{good} and T_{bad}, depend on the MT speed and the mean traveled distances in LoS and non-LoS conditions. We adopted the channel characterization reported in [15] for a rural area. In particular, when MT is traveling on a highway and the satellite elevation angle is 13°, the mean traveled distance in the GOOD (BAD) state is 90 (29) m [15]. Assuming an MT speed of 60 km/h, T_{good} lasts 6 s and T_{bad} 2 s.

At the physical layer, DVB-S adopts the MPEG2-TS (*Transport Stream*) packet of 188 bytes; after the application of a shortened (204, 188) *Reed-Solomon* (RS) cyclic code [16],[17], derived from the RS (255, 239), the frame length becomes 204-byte long. Moreover, an inner convolutional encoder is used before data transmission. The DVB-S standard also considers a convolutional interleaver between the two coders together with the adoption of the QPSK modulation. We assume using a modified DVB-S transmission modality, in the sense that we adopt both BPSK and QPSK as possible modulation schemes, while the inner coder is convolutional with rates 1/2 or 3/4. In other words, we define an ACM technique that can be used on the satellite air interface in order to cope with fluctuations in channel conditions. Table 1 reports system parameters of our ACM scheme for both GOOD and BAD states; in the BAD state, we adopt a higher *Forward Error Correction* (FEC) protection, and the BPSK modulation instead of the QPSK one.

The round trip channel latency is assumed equal to 280 ms: 250 ms for the satellite hop plus 30 ms for the terrestrial segment; the minimum TCP RTT takes into account a segment plus a TCP ACK transfer time. We neglected the delay from the server to the ES. We call *High Bandwidth* (HBW) and *Low Bandwidth* (LBW) the information bit-rate in the GOOD and BAD state, respectively. With the values shown in Table 1, the TCP versions considered reach an efficiency (ratio between the end-to-end transfer rate –goodput– and the information rate) of about 99%, considering the channel in stationary conditions (GOOD or BAD). In the following, the terms ‘frame’ and ‘packet’ are used interchangeably when referring to layer 2.

The relation that links E_b/N_0 (information bit energy to one-sided noise spectral density ratio) to C/N_0, where C denotes the carrier power, is:

$$E_b/N_0 = C/N_0 - 10Log_{10}b_r \quad [\text{dB}], \tag{1}$$

where b_r is the information bit-rate in bit/s, i.e., the rate at which the 188-byte MPEG2-TS packets are sent. C/N_0 values that characterize GOOD and BAD states, reported in Table 1, are compatible with the downlink Ka transponder of the Hotbird VI satellite with 32 carriers and an MT antenna of 40 cm. Furthermore, the *Frame Error Rate* (FER) is related to 188-byte frames (DVB-S packets), while the *Segment Error Rate* (SER) refers to TCP segments (*Maximum Segment Size* – MSS) of 8 packets each for a total of 1504 bytes. In order to evaluate FER and SER, we used the characteristics of a convolutional coder/Viterbi decoder over BPSK/QPSK modulated symbols relative to the NASA standard code with 1/2 rate, constraint length 7, and the derived 3/4 punctured code [19].

The value of b_r depends on the channel state. Let $b_{r,good}$ ($b_{r,bad}$) denote the information bit-rate in the GOOD (BAD) state; considering that the channel evolves between GOOD and BAD states, the average information bit-rate at layer 2, $E\left[b_r\right]|_{layer2}$, experienced for the transmission of MPEG2-TS packets (frames), is:

$$E\left[b_r\right]|_{layer2} = \frac{T_{good}}{T_{bad}+T_{good}} b_{r,good} + \frac{T_{bad}}{T_{bad}+T_{good}} b_{r,bad}. \tag{2}$$

For the selected channel model and $b_{r,good}$ and $b_{r,bad}$ values, we have: $E\left[b_r\right]|_{layer2}$ = 1.180 Mbit/s.

B. Packet Losses and Recovery Techniques

We aimed at investigating the effect of ACM on the TCP efficiency in the two-state channel. In order to realize the ACM, the system needs to estimate the channel quality at the physical level; after each estimation time, the receiver transmits the current channel quality value to the sender. References [20] and [21] describe some possible channel quality estimators of PSK modulated signals. The estimation time depends on both the number of symbols inspected and the symbol rate, other than the accuracy of measurements required. In our case, reasonable estimation times are in the order of tens of ms. In a part of our simulation runs we assumed that the modulation and FEC types used in GOOD/BAD state permit to reach the FER and SER values specified in Table 1 when the channel is actually in GOOD/BAD conditions; in both states, the channel is assumed to be of AWGN type. In addition, we also present cases in which packets sent and received in BAD state (by using LBW) have an FER higher than that specified in Table 1.

Critical or inefficient cases occur when there is a misalignment between the transmission modality adopted and the actual radio channel conditions due to both the estimation time and the delay in communicating a channel state variation. In particular, when there is a transition from BAD to GOOD, during the misalignment time, the transmission is overprotected by FEC and a capacity lower than the allowed one is exploited; hence, all bits are correctly received. On the other hand, when a transition occurs from GOOD to BAD, during the misalignment time, FEC protection is insufficient; thus, we consider that all transmitted frames are lost.

In our scenario, the round trip propagation delay is close to one half of that typically experienced in GEO communications. Therefore, an ARQ scheme of the SR type with low persistency [18] can be used to recover those packets that FEC is unable to correct. Typically, ARQ introduces further delays that may be harmful for the TCP congestion control mechanism in that RTO timeouts may expire before the end of the ARQ recovery cycle. However, ACK messages are transmitted to both the ARQ and TCP senders through the terrestrial cellular network; thus, the reduced RTT and the low ARQ persistence adopted allow avoidance of most RTO expirations.

In evaluating the system performance, we neglected both layer 2 and layer 4 ACK errors on the return terrestrial channel. This choice is due to the fact that, since both ARQ and TCP ACKs are cumulative (ACKn means that the previous $n-1$ packets have been correctly received), a moderate ACK loss does not produce any significant degradation in TCP performance. Results of tests, reported in the next Section, show that a slight performance degradation begins only when ACK losses are higher than 1%. ARQ aims at recovering both residual errors after FEC decoding and losses due to the aforementioned misalignments. In our study, we evaluated the TCP performance with adaptive parameters at the physical layer (ACM), with and without ARQ; in Fig. 2 the protocol stack is reported when both ACM and ARQ techniques are used.

C. The ARQ Scheme Adopted

Three different ARQ classes ([22] and references therein) are defined in the literature, basically: *Stop and Wait*, *Go back N*, and *Selective Repeat*. We refer to the last one, which gives the best performance on a high delay-bandwidth product link. Since there are no rigid rules, different ARQ versions are possible even within the same class. In the following, we provide a description of the scheme we used in our simulations.

Each TCP segment is fragmented into MPEG2-TS frames that are stored by the sender in a buffer (*Sender Link Buffer*, SLB) until the arrival of an ACK message from the receiver. ARQ employs a sliding window, similarly to the TCP sending mechanism. Both ACK messages and frame headers contain four 16-bit fields: SN (*Sequence Number*), BoS (*Beginning of Segment*), EoS (*End of Segment*), and CRC (*Cyclic Redundancy Check*). In the ACK message, the receiver indicates the SN of the next expected frame, which denotes the correct reception of all preceding frames as well. When frames are lost, on reception of a valid frame, the receiver sends a duplicated ACK and indicates in both the BoS and EoS fields the first and the last of the lost frames, respectively. On reception of a duplicated ACK, the sender immediately retransmits all lost frames: ARQ retransmissions are prioritized with respect to packets to be transmitted for the first time; this is needed in order to limit the delay to recompose a segment to be delivered to higher layers at the receiver.

A maximum number (i.e., persistency level) of packet retransmissions, N_{max}, is adopted in order to avoid expiration of the TCP RTO in the presence of repeated errors on retransmissions. In our simulations, we set N_{max} = 1 just to keep the maximum delay to send a correct packet sufficiently lower than TCP RTO. Packets are re-sent with the maximum protection (i.e. in LBW like in the BAD channel state). These two choices, together with a suitable setting of ARQ timers (see next paragraph) allow us avoiding most of TCP RTO expirations.

D. TCP Segment Reassembly

The receiver uses a link layer buffer (*Receiver Link Buffer*, RLB) to assemble packets into TCP segments, which must be delivered in sequence to higher layers. Due to channel losses, packets need to be stored until losses are recovered by retransmissions. Only correctly received packets originate layer-2 ACKs (both original or duplicate); furthermore, resent packets may be lost as well. For the above reasons, a timeout mechanism, on both the sender and receiver side of ARQ, is necessary to limit TCP RTO expirations and to prevent deadlocks. We remind that a TCP RTO causes a slow start, while the loss of one or more segments causes a fast recovery, in TCP NewReno and more recent versions. TCP RTO expirations highly affect the goodput degradation in high DBP channels.

Let ASTO (*ARQ Sender Time Out*) be the timer set by the sender when a frame is transmitted; therefore, an individual ASTO is set for each frame and is reset when the ARQ ACK of the relative frame is received. If ASTO expires, the sender retransmits the relative frame.

As we consider unitary persistence, after that frames have been resent they are removed from the SLB. After the completion of each in-sequence segment, the receiver restarts the

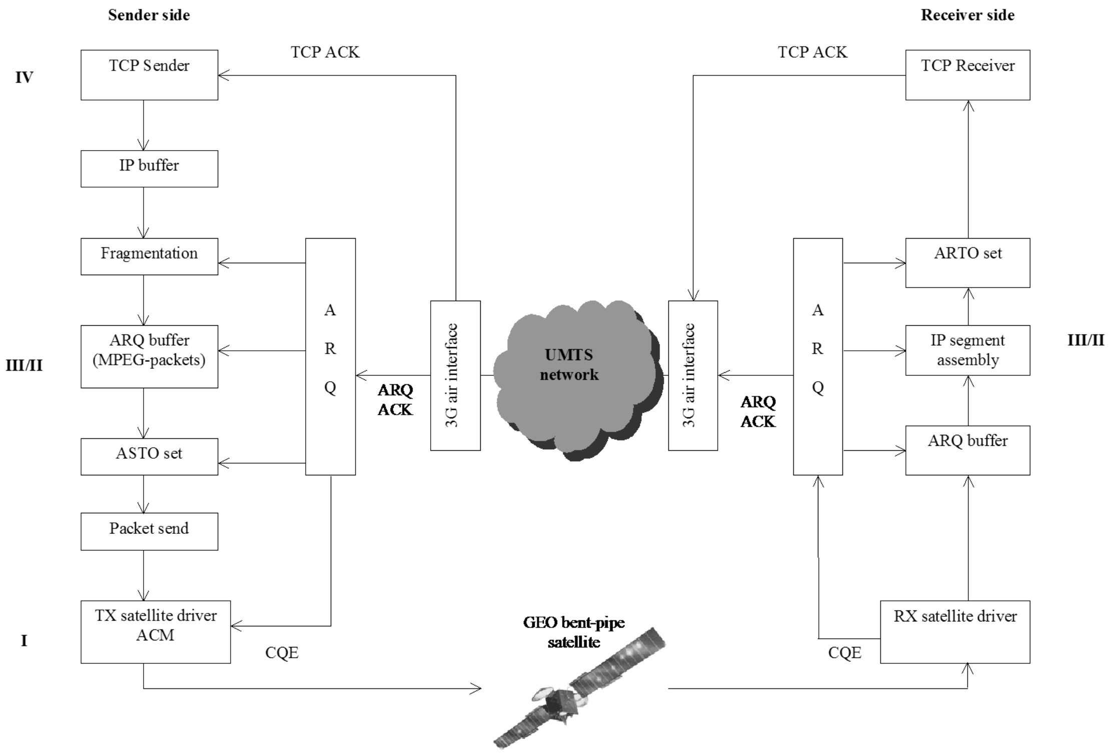

Fig. 2. The protocol stack.

ARTO (*ARQ Receiver Time Out*) timer (which is unique in the receiver) for the next segment completion. After ARTO expiration, the receiver behaves as if the current segment had been completed; consequently, it delivers the successive recomposed segments, if any, to the higher layer and frees the related space in RLB. In this case, the TCP sender receives duplicated TCP ACKs due to the missed segment(s), which cause fast recovery instead of slow start due to RTO expiration. To exploit the whole channel information bandwidth, b_r, we must have:

$$\text{RLB} = \text{SLB} \geq \text{ASTO} \times b_r, \tag{3}$$

where we assume b_r = $b_{r,good}$, i.e., the information bit-rate in the GOOD state. Furthermore, in order to prevent TCP RTO's and to avoid or to limit duplicated frames due to RTT's jitter, it must be:

$$\text{RTO} - \text{RTT} > \text{ARTO} > \text{ASTO} > \text{RTT}. \tag{4}$$

RFC 2988 states rules to compute RTO, which depends on the current measurement of RTT and by its jitter; anyway, if the computed value of RTO results lower than 1 s, it is set equal to 1 s. In our case, it is reasonable assuming that the jitter of the RTT is not most likely so high to cause an RTO computation greater than 1 s. Therefore, unless otherwise specified, in our simulations we set ASTO = 0.32 s, while the value chosen for ARTO = 0.6 s is such to tolerate a moderate jitter on RTT without causing segment losses false alarm. Several simulation runs have been performed with different ARTO and ASTO values leading us to the choice of these settings; however, the choices of both these timers are not easily testable via simulation because the jitter of RTT is difficult to simulate. With the values of ASTO and ARTO chosen, an ARQ persistency grater than one would cause an ARTO expiration before the packet recovery, in case of failure of the first packet retransmission. For instance, two retransmissions (persistency 2) would require an ARTO greater than 2$\times$ASTO; thus, we should increase the previous ARTO value with the consequence of reducing the jitter tolerance on RTT, according to (4).

IV. Simulation Results

By using the ns-2 simulator, we implemented the scenario described in Fig. 1, assuming the FER values shown in Table 1. We have not adopted the *delayed ACK* algorithm (i.e., an ACK is sent for each received TCP segment). Delayed ACKs would entail a considerably worse performance due to both a slower increase in *cwnd* and more frequent RTO expirations.

For each test, we ran 20 independent simulations, each lasting 3 minutes, with exponentially distributed GOOD/BAD duration times. All figures indicate the 95%-level confidence intervals around the mean values.

The TCP goodput (i.e., the end-to-end transfer rate) has been considered in a single TCP connection case. In all simulations, ARTO is always active, while ASTO is only present when explicitly specified. The TCP sender output rate (throughput), Γ, is typically higher than the goodput, due to retransmissions performed by the sender; it can be approximated as:

$$\Gamma = min\{cwnd/\text{RTT}, E[b_r]\} \tag{5}$$

where *cwnd* is upper-bounded by the *advertised receiver window* (*rwnd*). The TCP traffic is 'ACK-clocked', meaning

TABLE II
AVERAGE GOODPUT IN RCS AND RCT CASES WITH CONFIDENCE INTERVALS LOWER THAN 4.5%.

	ASTO [ms]	CQE time [ms]	SACK goodput [kB/s]	NewReno Imp. goodput [kB/s]
RCS	540	30	126.3	115.40
RCT	320	30	101.55	81.21

that for each received ACK there is always a new segment available for transmission.

Our simulation results point out the influence of the various layer 1-2 techniques and parameters on the system performance, in terms of the TCP connections' average goodput. We simulated the system by using either a *Return Channel via Terrestrial* segment (RCT) or the *Return Channel via Satellite* (RCS), for sending ACKs and channel quality estimation updates. Results relative to two different TCP versions are reported in Table 2. We can observe that the goodput gain of RCT over RCS, both used without ARQ, is noticeable for both TCP versions considered. However, the improved goodput performance is not the only advantage in using RCT; in fact, satellite transmissions are much more costly and impose more constraints to the MT than a UMTS connection. Therefore, the following results refer to the RCT case according to the architecture depicted in Fig. 1.

Figure 3 shows the most significant system timings when ACM and ARQ are employed and a GOOD/BAD condition of 6s/2s is experienced. Here, CL denotes the *Channel Latency*, that is the round trip propagation delay of the signal, while CQET denotes the *CQE time*. In the figure, at t = 6 s the channel switches from GOOD to BAD. Thus, all packets received from 6 s up to 6+CL+CQET s are lost (SER = 1), if ASTO is set to b >CL+CQET; instead, packets are lost from 6 s up to 6+ASTO s, if ASTO is set to a <CL+CQET. This occurs because we chose that the ARQ sender retransmits timed-out packets with the highest protection, i.e., LBW mode. In the last case, the number of lost packets does not depend on the CQET. At time t = 8 s, the channel switches from BAD to GOOD; thus, the channel is under-exploited for a time interval of length CL+CQET.

We made simulation runs with different ASTO and CQET values. Results, not reported here for the sake of brevity, showed an increase in the goodput in correspondence to values of CQET higher than ASTO–CL. In fact, if CQET is lower than ASTO$-$CL, a number of good packets are accumulated in the receiver ARQ buffer, after the train of lost packets caused by the GOOD to BAD switching of the channel. This is due to the fact that the notification of the channel state change reaches the ARQ sender before the ASTO expiration. On this event, the ARQ sender begins retransmitting lost packets, which are received in sequence by the ARQ receiver, assembled in segments, and delivered in sequence to the TCP receiver. When all lost packets are correctly received, previously accumulated packets cause the delivery of a segment train from the ARQ to the TCP receiver. This train of segments generally causes a train of ACK's, which arrive back-to-back at the TCP sender if no ACK spacing feature is implemented, as in our case.

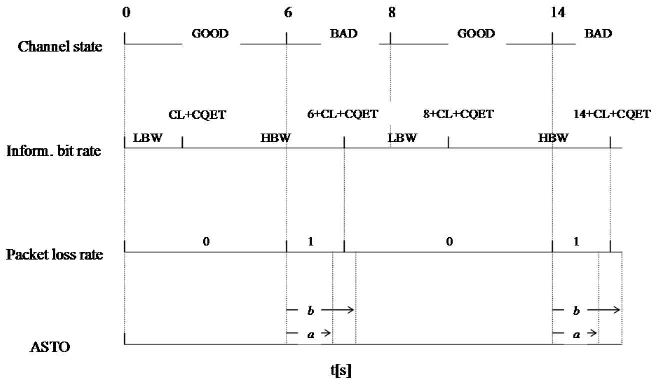

Fig. 3. System timings (case b with ASTO $>$ CL+CQET and case a with ASTO $<$ CL+CQET).

Finally, the train of ACK's gives rise to a train of transmitted segments by the TCP sender, which may cause a multi-segment overflow in the IP buffer. The consequent recovery phase may be long (NewReno-SS or SACK) or may provoke an RTO expiration (NewReno-Imp). The generation of the train of segments by the TCP sender does not occur if CQET is longer than the difference ASTO$-$CL. In this case, there is no packet accumulation at the ARQ receiver and all packets are lost until the ASTO expiration. As in nearly all cases in which timeouts are employed, in our case it is convenient to size the timeout itself as short as possible. This can be done for channels with a low delay variation (jitter); in fact, the risk in sizing the timeout too short is that it may expire even for an occasional longer delay experienced by the expected ACK and not for an actual loss. Of course, this fact should occur with a very low probability, in order not to decrease the system efficiency.

We previously explained that CQET has no relevant influence on system performance, when a well-calibrated ASTO value is chosen and retransmissions of lost packets are performed at the low rate. However, this is not true if the ASTO timer is not employed.

In Fig. 4, the performance of both NewReno variants and SACK are compared each other, with and without an ASTO of 320 ms, for different values of CQET. The advantage in using ASTO is evident, especially for high values of CQET.

Figure 5 shows all possible cases (36 in total) we simulated by combining different options at layers 1, 2 and 4 in the aim of our cross-layer study. Simulation results are reported in Figure 6 in terms of TCP goodput. These results are organized in 9 blocks (from the leftmost –i.e., first– block to the rightmost –i.e., 9-th– block); in each of them we have the results for the 4 TPC variants considered. Losses assumed in GOOD (BAD) state are those specified in Table 1, when HBW (LBW) is adopted. The most evident result is that, if ARQ is not used, it is not worthwhile to use ACM, at least with the system parameters we adopted. In fact, all TCP versions with ACM and without ARQ under-perform all cases in which LBW is permanently adopted. The performance of the last (i.e., 9-th) block of results is the same of those in the 6-th and in the 3-rd blocks: when LBW is permanently used (NO ACM), the system performance is insensitive to the adoption of ARQ since all these cases are practically without losses. Moreover, the results in these blocks outperform those in the 8-th block in which HBW is permanently used (NO ARQ, NO ACM, HBW mode). On its turn, this last case slightly outperforms the cases in which ACM is used without ARQ (i.e., 7-th block of results). The reason for this is that the gain in exploiting HBW, as soon as the channel becomes good, overcompensates all losses during whole BAD states when HBW is permanently adopted. The same modality that allows to exploit the HBW as soon as the channel becomes good explains why the cases in the second block (a part from the SACK version) slightly outperforms cases in the first block. Note that the cases in the second block uses ARQ with ASTO. Figure 6 also shows that TCP NewReno with the SACK option yields the best performance in high efficiency cases, whereas Westwood works better then others in low efficiency cases. This is due to the fact that the SACK recovery phase is faster than the one of the other versions, when multiple segments are lost in a window of data and Westwood better manages high loss situations. The Westwood version that produced results shown in Fig. 6 did not adopt the SACK option; surprisingly, runs performed with this option did not produce sensibly different results. The gain in performance of NewReno with SACK is particularly relevant if ACM is employed together with ARQ without ASTO.

We now show the system performance degradation when the FEC adopted in the BAD state is not sufficient to obtain the FER reported in Table 1; thus, higher FER values are assumed. Figure 7 shows the goodput of the four TCP versions considered versus the FER in BAD state; in each graph two cases are represented: ARQ+ACM and ARQ with HBW (i.e., packets are always transmitted in HBW and lost ones are retransmitted in LBW, without ACM). ASTO is set to 320 ms in both cases. As previously observed, these two cases do not exhibit much difference in performance. Furthermore, SACK performs slightly better than the others for a low FER, while Westwood seems to manage a high FER level better than the others.

Figure 8 shows the goodput of all TCP versions considered versus the average GOOD state duration, when the average duration of the BAD state is 2 s and the FER values are as specified in Table 1. Once again, we can observe that the best performance is achieved by SACK.

In Fig. 9, we reported the goodput degradation of all TCP versions considered versus the ACK loss rate. The values reported in the abscissa refer to ARQ ACKs; however, in the simulations we took into account a TCP ACK loss rate as well. Such a loss (ALR_{TCP}) has been taken equal to $1-(1-ALR_{ARQ})^5$, where ALR_{ARQ} is the ARQ ACK loss rate. The reason for this is that we considered independent bit errors on the return channel and a TCP ACK is 5 times longer than an ARQ ACK. Even if this packet error model is simplified, we believe that results in Fig. 9 clearly prove that moderate ARQ ACK loss rates have a negligible impact on the TCP goodput. This confirms the validity of previous results.

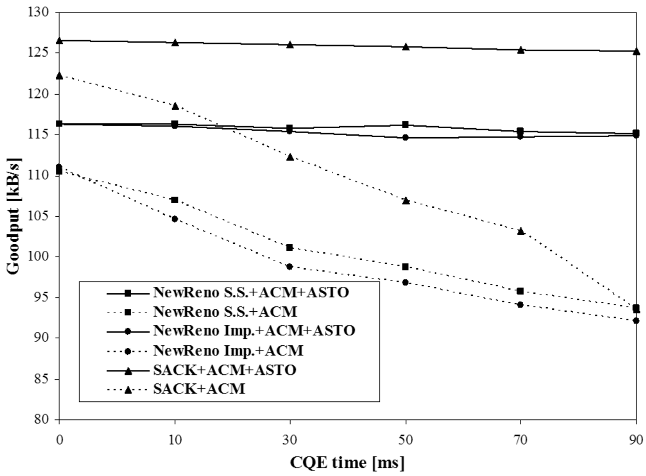

Fig. 4. Performance of different TCP types (with ARQ and ARTO) versus CQE time, with and without an ASTO of 320 ms. The maximum confidence interval for all cases resulted of ±4.6%.

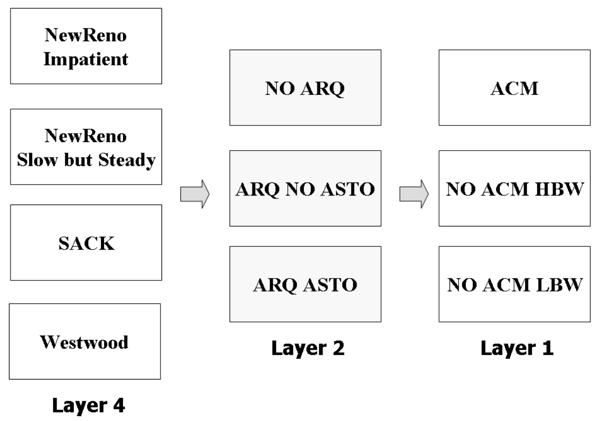

Fig. 5. Different selections available at the different layers to be optimized in the aim of the cross-layer design.

V. Conclusions

We focused on a hybrid network architecture, which involves a GEO bent-pipe satellite link in the forward path and a 3G terrestrial cellular system in the reverse path, and studied the goodput performance of different TCP variants. A GOOD-BAD channel model has been assumed for MT's. This system presents advantages with respect to a network that uses both forward and return links via satellite, both in terms of RTT and in allowing less complex MT's. An ACM scheme, based on a modification of the DVB-S standard, has been proposed in order to adapt the transmission characteristics to the satellite channel quality.

Four different TCP versions have been compared each other with and without the employment of ACM and ARQ in the aim of the cross-layer study. The individual influence of the various techniques adopted on the TCP performance was evaluated. The main result from our simulations is that ACM does not yield any goodput improvement if ARQ is not used as well. Furthermore, the relevance of the timeout mechanism in the

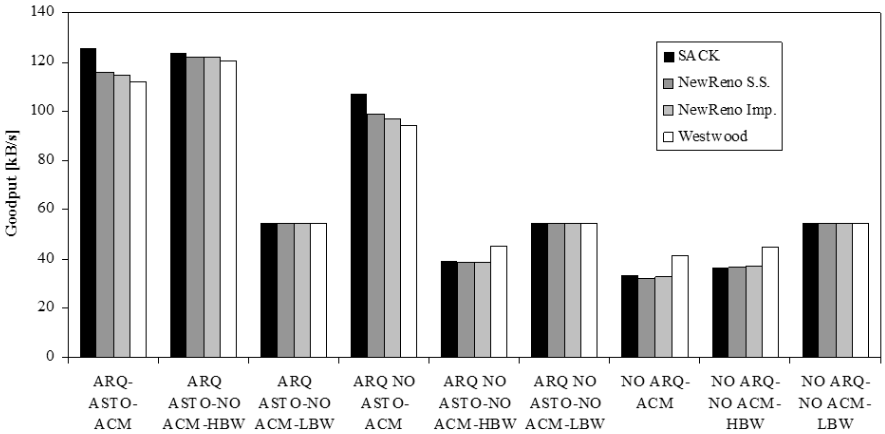

Fig. 6. Average goodput of four TCP types by using different techniques. When present, ASTO = 320 ms and CQET = 50 ms. The maximum confidence interval of all values was ±4%.

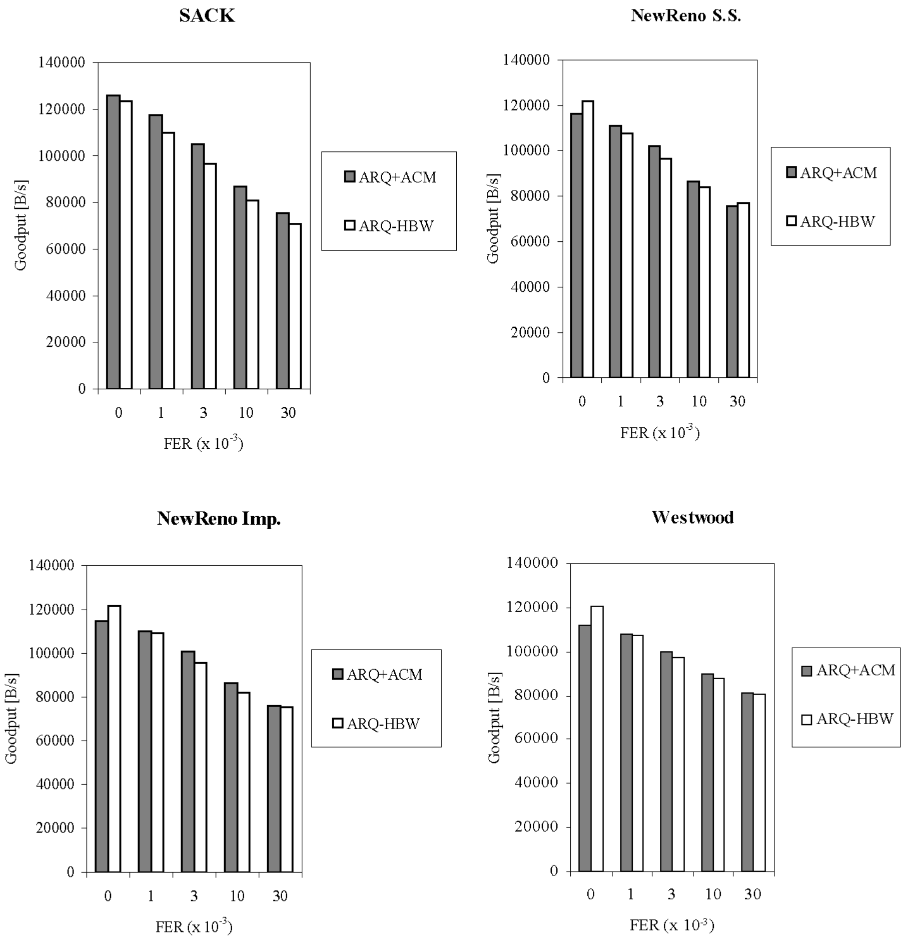

Fig. 7. Goodput of different TCP types versus FER in BAD state, using ARQ+ACM and using ARQ and HBW. ASTO is 320 ms in both cases. The maximum confidence interval of all values was ±4.3%.

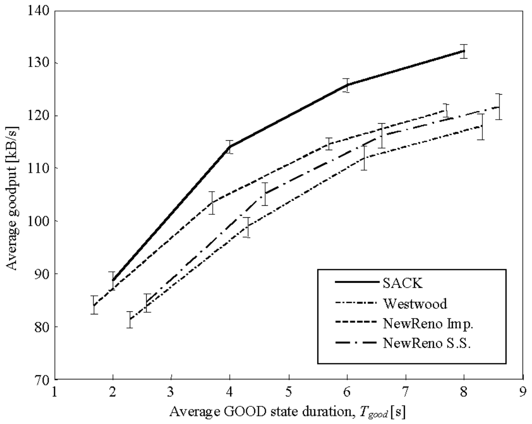

Fig. 8. Goodput of different TCP versions versus the average GOOD state duration, with an average BAD state duration of 2 s. ACM+ARQ with ASTO = 320 ms are employed, and CQET = 50 ms for all versions.

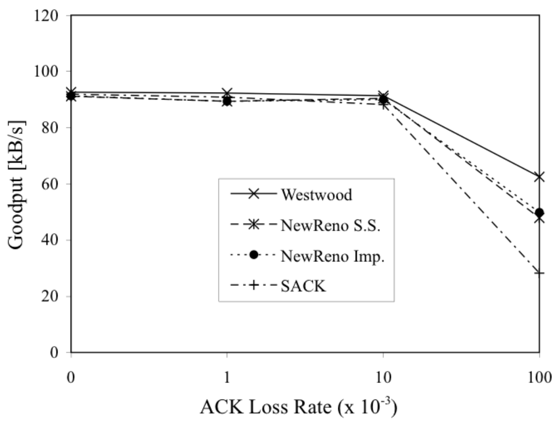

Fig. 9. Goodput of various TCP versions versus ACK loss rate. The maximum confidence interval for all cases was ±4.2%.

ARQ protocol has been pointed out. In fact, choosing the unitary persistence of ARQ and retransmitting corrupted frames with the maximum protection, the system's performance is nearly insensitive to the use of ACM, provided that suitable values of timeouts be chosen. These two major results have been confirmed by the behavior of all four versions of TCP considered. We can also assert that TCP SACK exhibited a slightly better performance than others in most efficient combinations, whereas TCP Westwood seems to handle heavy loss situations better than others. However, Westwood seems to suffer more than others from changes in the channel rate due to the adaptation of transmission parameters to the channel state; this usually occurs when ACM is employed.

In a future study, we envisage to perform the same study with DVB-S2 standard for mobile users together with a deeper investigation with a more realistic 3G channel, based on correlated losses.

ACKNOWLEDGMENTS

This work has been carried out within the European-funded network of excellence SatNEx II (contract No. IST-027393).

The authors would like to thank Dr. Raffello Secchi of ISTI-CNR for his support.

REFERENCES

[1] H. Holma and A. Toskala. *WCDMA for UMTS - Radio Access for Third Generation Mobile Communications*. John Wiley & Sons, 3rd ed.

[2] M. Allman, D. Glover, L. Sanchez, "Enhancing TCP over Satellite Channels using Standard Mechanism", IETF RFC 2488, January 1999.

[3] M. Allman, V. Paxson, W. S. Stevens, "TCP congestion control", IETF RFC 2581, April 1999.

[4] S. Floyd, T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm", IETF RFC 2582, 1999.

[5] M. Mathis, J. Mahdavi, S. Floyd, A. Romanow, "TCP Selective Acknowledgment Options", IETF RFC 2018, 1996.

[6] C. Partridge, T. J. Shepard, "TCP/IP Performance over Satellite Links", *IEEE Network*, September-October 1997, pp. 44-49.

[7] M. Allman (Editor), "Ongoing TCP Research Related to Satellites", IETF RFC 2760, February 2000.

[8] L. A. Grieco, S. Mascolo, "Performance Evaluation and Comparison of Westwood+, New Reno, and Vegas TCP Congestion Control", *ACM Computer Communication Review*, Vol. 34, No. 2, April 2004.

[9] I. F. Akyildiz, G. Morabito, S. Palazzo, "TCP-Peach: a new congestion control scheme for satellite IP networks", *IEEE/ACM Trans. Networking*, Vol. 9, No. 3, June 2001.

[10] ETSI, "Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems". Final draft ETSI EN 301 790 V1.3.1 (2002-11).

[11] ETSI, "Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2)", Draft EN 302 307.

[12] J. Zhu, S. Roy, J. H. Kim, "Performance Modelling of TCP Enhancements in Terrestrial–Satellite Hybrid Networks", *IEEE/ACM Transactions on Networking*, Vol. 14, No. 4, pp. 753-766, August 2006.

[13] N. Celandroni, F. Potortì, "Maximising Single Connection TCP Goodput by Trading Bandwidth for BER", *Internat. J. Commun. Syst.*, Vol. 16, No. 1, pp. 63-79, Febraury 2003.

[14] N. Celandroni, "Comparison of FEC types with Regard to the Efficiency of TCP Connections over AWGN Satellite Channels", *IEEE Trans. on Wireless Commun.*, Vol. 5, No. 7, pp. 1735-1745.

[15] E. Lutz, D. Cygan, M. Dippold, F. Dolainsky, W. Papke, "The Land Mobile Satellite Communication and Recording, Statistics and Channel Model", *IEEE Trans. Veh. Tech.*, Vol. 40, pp. 375-386, May 1991.

[16] S. Lin and D. J. Costello. *Error Control Coding: Fundamental and Applications*. Prentice-Hall, 1983.

[17] M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, *Spectrum Communications Handbook*, revised edition: McGraw-Hill, 1994.

[18] G. Fairhurst, L. Wood, "Advice to Link Designers on Automatic Repeat reQuest (ARQ)", IETF RFC 3366, 2002.

[19] Qualcomm incorporated, "Q1401: K=7 rate 1/2 single-chip Viterbi decoder technical data sheet", Sep. 1987.

[20] N. Celandroni, E. Ferro, F. Potortì, "Quality Estimation of PSK-modulated Signals", *IEEE Comm. Mag.*, Vol. 35, No. 7, pp. 50-55, July 1997.

[21] Bin Li, R. A. DiFazio, A. Zeira, P. J. Pietraski, "New Results on SNR Estimation of MPSK Modulated Signals", *Proc. of the 14th IEEE Int. Symposium on Personal, lndoor and Mobile Radio Communication*, pp. 2373-2377, 2003.

[22] M. Zorzi, R. R. Rao, L. B. Milstein, "ARQ Error Control for Fading Mobile Radio Channels", *IEEE Trans. Veh. Tech.*, Vol. 46, No. 2, pp. 445-455, May 1997.

Nedo Celandroni received the Dr. Ing. degree in Electronic Engineering from the University of Pisa, Italy, in 1973. Since 1976 he has been a researcher with the CNUCE Institute (now ISTI) of the Italian National Research Council (CNR). He worked for the realization of the Flight Dynamic System of the SIRIO satellite project. Since 1979 he has been involved in the field of digital satellite communications. He participated in several projects in this field: STELLA I/II (Satellite Transmission Experiment Linking Laboratories), FODA (Fifo Ordered Demand Assignment), FODA/IBEA (Information Bit Energy Adaptive), Progetto Finalizzato Telecomunicazioni, Experiments on the satellites Olympus and Italsat. His interest includes rain fade countermeasure systems, data quality estimation, VSAT systems, GEO, MEO and LEO satellites for mobile telephony and multimedia systems, wireless networks, TCP congestion management, and TCP over wireless channels. He obtained two patents, in 1989 and 1996, for the design of the FODA and FODA/IBEA systems, respectively. At present he is involved in the *SatNEx* European Network of Excellence, and in some National Projects. Dr. Celandroni is a reviewer for many international journals and conferences. He co-authored more than a hundred papers published on international journals and conference proceedings.

Erina Ferro received her Laurea degree with distinction in Computer Science from the University of Pisa, Italy, in 1975. Since 1976 Dr. Ferro was with CNUCE, an Institute of the Italian National Research Council (C.N.R.) in Pisa. In the first half of 2002 CNUCE merged with another Institute of C.N.R., thus originating the ISTI/CNR Institute (Information Science and Technology Institute "Alessandro Faedo"), where she is currently employed as a senior researcher. Her main scientific interest field was and is digital transmission via satellite. Working on this topic she participated in several international and national projects, where some systems designed by her team were implemented and used on the Eutelsat, Olympus, and Italsat satellites. Working in the TDMA satellite access schemes field she obtained two patents, in 1989 and 1996, respectively, together with other colleagues of her group. Currently, her main interest areas are satellite access schemes for multimedia traffic transmissions with guaranteed quality of service, fade countermeasure techniques, call admission control policies for resource allocation, the satellite systems' performance analysis, wireless LANs, such as IEEE802.11, and their interconnection with the satellite network. She is Head of the Wireless Laboratory at ISTI, and is responsible for ISTI of many National and International projects. The Wireless Laboratory of ISTI is a member of the *SatNEx* Network of Excellence, funded by the European Commission. Dr. Ferro is a reviewer for many international journals and conferences. She co-authored more than a hundred papers published on international journals and conference proceedings.

Giovanni Giambene received the Dr. Ing. degree in Electronics in 1993 and the Ph.D. degree in Telecommunications and Informatics in 1997, both from the University of Florence, Italy. From 1994 to 1997, he was with the Electronic Engineering Department of the University of Florence, Italy. He was Technical External Secretary of the European Community COST 227 Action ("Integrated Space/Terrestrial Mobile Networks"). He also contributed to the SAINT Project ("Satellite Integration in the Future Mobile Network", RACE 2117). From 1997 to 1998, he was with OTE of the Marconi Group, Florence, Italy, where he was involved in a GSM development program. In the same period he also contributed to the COST 252 Action ("Evolution of Satellite Personal Communications from Second to Future Generation Systems") research activities by studying PRMA protocols for voice and data transmissions in low earth orbit mobile satellite systems. In 1999 he joined the Information Engineering Department of the University of Siena, Italy, first as research associate and then as assistant professor. He teaches the advanced course of Telecommunication Networks at the University of Siena. From 1999 to 2003 he participated to the project "Multimedialit", financed by the Italian National Research Council (CNR). From 2000 to 2003, he contributed to the "Personalised Access to Local Information and services for tOurists" (PALIO) IST Project within the EU FP5 programme. At present, he is involved in the SatNEx network of excellence of the FP6 programme in the satellite field, as work package leader of two groups on radio access techniques and cross-layer air interface design (www.satnex.org). He is also vice-Chair of the COST 290 Action (www.cost290.org), entitled "Traffic and QoS Management in Wireless Multimedia Networks" (Wi-QoST). Is research interests deal with radio resource management, queuing theory and satellite communications.

Mario Marandola received his 'laurea' degree in Telecommunications Engineering from the University of Siena, Italy, in 2004. He has partecipated to the European Satellite Communications Network Excellence, SatNEx, with a joint activity between ISTI-CNR, Pisa, and CNIT-University of Siena. He has worked in the field of satellite communications on cross-layer approaches for the TCP protocol in the presence of wireless channels.