Cross-layer approach for an air interface of GEO satellite communication networks

Sastri Kota¹, Giovanni Giambene²,*,† and Nico Liberato Candio²

¹Harris Corporation-GCSD, 1134 East Arques Avenue, Sunnyvale, CA 94086, U.S.A.
²Dipartimento di Ingegneria dell’Informazione, Università degli Studi di Siena, Via Roma, 56, 53100 Siena, Italy

SUMMARY

Satellite communications play a significant role in supporting next-generation IP-based networks. To deploy state-of-the-art satellite technologies supporting media-rich applications, efficient utilization of radio resources and end-to-end Quality of Service support are mandatory requirements. The study of satellite communication constraints such as attenuation, propagation delays, fading is very critical to support user service level agreements. Adaptive resource management, advanced modulation and coding techniques and congestion control algorithms must be jointly conceived. Transport protocols need to address the impact on end-to-end system performance due to lower layer protocols. In this paper, we focus on cross-layer adaptation and interaction analysis. In particular, we refer to TCP and lower layers in a geostationary-based network architecture, assuming transport protocol enhancements such as TCP NewReno, SACK, Westwood + and Hybla. At the physical layer, modulations and coding considered include BPSK and QPSK with convolutional coder/Viterbi decoder. The interest of this study is in evaluating the impact of two different transmission mode selection techniques: one scheme selects between the two modes on the basis of the layer 2 packet error rate performance (non-cross-layer method); while, the other technique selects the transmission mode on the basis of the TCP goodput performance (cross-layer method). Our simulation results demonstrate that the second scheme permits the significant improvement of the end-to-end performance. Copyright © 2007 John Wiley & Sons, Ltd.

Accepted 23 June 2007

KEY WORDS: satellite networks; cross-layer design; TCP goodput; physical layer adaptivity

1. INTRODUCTION

Today, still a large number of persons living in remote areas or in underdeveloped regions do not have a realistic perspective of achieving access to high-speed Internet for many years. This is...
a serious obstacle to make information available to all. Such digital divide problem can be solved by satellite communications that can easily reach different regions on the earth by providing the same service types everywhere.

New satellite system architectures are being envisaged to be fully Internet Protocol (IP)-based and to support digital video broadcasting, including forward and return channel protocols, e.g. DVB-S, DVB-S2, and DVB-RCS [1]. Satellite resources are expensive and satellite communications impose special constraints with respect to terrestrial systems in terms of attenuation, propagation delays, fading, etc. This is the reason why within the research community a range of issues are currently being investigated, which are expected to improve the efficiency and capacity of satellite communication systems. Future satellite networks must support various traffic types with their associated Quality of Service (QoS) requirements related to the end-to-end content delivery. QoS requirements for multimedia traffic have been covered by different standardization groups, like International Telecommunication Union (ITU), European Telecommunication Standards Institute (ETSI), and Third-Generation Partnership Project (3GPP). The main work provided by ITU is behind Recommendations Y.1541, F.700 and G.1010, where applications have been classified in eight groups according to error tolerance and delay.

The ISO/OSI reference model and the IP suite are based on a layered protocol stack. Protocols are designed such that a higher-layer protocol only makes use of the services provided by the lower layer and is not concerned with the details of how the service is being provided; the protocols at the different layers are independently designed. The layered approach is also related to different timescales at the different levels of the protocol stack: from small scales at lower layers up to large scales in the order of seconds at layer 4 and above. Due to the dynamics of the radio link and resource congestion conditions, such a layered approach shows its limits and a better coordination of the different protocol layers would be required. The main focus of this paper is to show the potential benefits in terms of goodput considering cross-layer protocol involving physical (PHY) layer and transport layer (i.e. the Transmission Control Protocol, TCP).

This paper is organized as follows. The next section provides a brief overview on cross-layer approaches and signaling methods. Section 3 deals with a short overview of the cross-layer interactions among PHY, data, network, transport, and application layers. Sections 4 and 5, respectively, describe the characteristics of our examined scenario and the performance results of a cross-layer technique involving PHY and TCP layers. Section 6 concludes this paper highlighting further research areas.

2. CROSS-LAYER DESIGN APPROACHES AND SIGNALING

The layered approach in the air interface design is based on the separate optimization of distinct parts, thus reducing the complexity and allowing the interoperability among equipments of different manufacturers through the use of standardized interfaces. However, in general, there exists tight interdependence between layers in satellite networks; hence, a strict modularity and layer independence may lead to a non-optimal performance. For instance, transport layer protocols need to take into account large propagation delays, link impairments and bandwidth asymmetry. In addition to this, error correction schemes are implemented at physical, link and (in some cases) transport layers, thus entailing some inefficiencies and redundancies.
The interest here is in protocol architectures violating the reference layered architecture, for example, by allowing direct communication between protocols at non-adjacent layers or sharing state variables between layers to obtain performance gains. Such violation of the layered architecture is what we mean by cross-layer design [2]. Some initial papers addressing the cross-layer air interface design in the satellite scenario are available in the literature; for a detailed survey of them, the interested reader should refer to [3]. Most of the current literature on cross-layer design envisages adaptivity only involving lower layer protocols; the interest of this paper is to relate adaptivity also to the transport layer where different protocol versions can be used.

The cross-layer approaches can be distinguished on the basis of the way the layered architecture is violated, according to the following typologies [2]: (i) creation of new interfaces beyond those between adjacent layers; (ii) merging of adjacent layers; (iii) joint design of protocols at different layers; (iv) vertical calibration of the whole protocol stack. The different cross-layer schemes can also be classified according to the presence or absence of cross-layer signaling. Correspondingly, three different methods can be envisaged as follows:

- **Implicit cross-layer design** (above cases (ii), (iii) and (iv)): There is no exchange of signaling among different layers during operation, but in the design phase all the layers and interactions are taken into consideration in order to perform a joint protocol optimization.
- **Explicit cross-layer design** (above case (i)): Signaling among (non-)adjacent protocol levels is used to achieve dynamic adaptations, involving together all (or many of) the protocol layers.
- **Combined approach**: An implicit approach is used to optimize the system in the normal operating conditions and the explicit method is adopted to perform adaptations according to system dynamics.

The joint protocol design of the implicit approach is a relatively simple task, requiring the optimization of the protocol stack in the design phase; no signaling is involved. While, the explicit cross-layer approach is more complex to be implemented, typically needing new signaling solutions and standardization choices. The combined approach takes advantage of both the optimization process of the implicit scheme and the adaptivity of the explicit scheme.

In a classical layered approach, the exchange of information can only be performed between adjacent layers through send and receive primitives. Non-adjacent layers can only communicate through intermediate layers. The novelty of the explicit cross-layer method is to allow the direct exchange of control information (signaling) among non-adjacent layers (see Figure 1). Two signaling directions can be envisaged, that is from higher to lower layers (**top-down approach**) and from lower to higher layers (**bottom-up approach**). Signaling can be realized with different approaches that are outlined below [4]:

- **Packet headers**: This method makes use of packet headers as in-band message carriers; there is no need of a dedicated internal message protocol. Since, a packet normally can only be processed layer by layer, this method can be visualized like a ‘signaling pipe’ and is well suited for the top-down approach.
- **ICMP messages**: ICMP (Internet Control Message Protocol) is a widely deployed signaling protocol in IP-based networks. Compared with the pipe described above, this method tries to ‘punch holes in the protocol stack’ and propagates information across layers by using ICMP messages. A new ICMP message is generated only when a parameter changes beyond a given threshold. Since cross-layer communications are carried out through...
selected ‘holes’ (not a general ‘pipe’), this method seems more flexible and efficient. However, an ICMP message is always encapsulated into an IP packet, and this indicates that the message has to pass by the network layer even if the signaling is only between link and application layers. This method is well suited for the bottom-up cross-layer approach.

- **Network service:** In this scheme, channel and link states are collected, abstracted and managed by third parties, i.e. distributed servers. Interested applications then access the servers for their required parameters from the lowest two layers. Although there is not a cross-layer signaling scheme within a terminal, we can deem this scheme as complementary to the two above techniques. However, any intensive use of such method would introduce considerable signaling overhead and delays across a radio access network.

- **Local profiles:** In this method, local profiles are used to store periodically updated information coming from protocol layers. Cross-layer information is abstracted from each necessary layer and stored in separate profiles within the hosts. Other interested layer(s) can then select the profile(s) to fetch the desired information.

However, in [4,5] and in many documents in the literature (see also [3] and references therein for a comprehensive survey) an interesting method has emerged to implement the explicit cross-layer design method. In particular, we can consider the use of a **global coordinator** of the different layers that is able to acquire internal state information from the different protocols and that can set the internal state of the protocols as a response to suitable events (e.g. buffer overflow, **Signal-to-Noise Ratio** (SNR) variations, packet losses, timer expirations, selection of a new modulation and coding (MODCOD) level, etc.). A general scheme of the global coordinator is shown in Figure 1. The global coordinator can be implemented in different ways, as follows:

- A given protocol layer takes the control to coordinate the whole protocol stack. This protocol layer could be the MAC layer (**MAC-centric approach**) or the application layer (**application-centric approach**).
A shared memory with interfaces and suitable primitives with all the layers.
The management plane.
A network server that has a dialogue with the (remote) terminal.

Such a cross-layer architecture should be adapted to the ETSI Broadband Satellite Multimedia (BSM) air interface protocol stack [6] where protocols are distinguished into two main groups (layers 1 and 2, named satellite-dependent layers, and layers 3 and 4, named satellite-independent layers). These two blocks are interconnected through a general interface (Satellite Independent-Service Access Point, SI-SAP) whose primitives have to be designed to take cross-layer signaling into account.

3. INTERACTIONS AMONG PROTOCOL LAYERS

A pictorial view of the possible interactions (explicit cross-layer design case) is shown in Figure 2 and the description of them is provided below.

3.1. PHY and MAC layer interactions

Considering a PHY layer supporting ACM, it is important to optimize the selection of MODCOD levels depending on channel conditions by employing a closed-loop control via a return (terrestrial or satellite) channel from each user. More conservative transmission modes should be used during rain fades, thus trading the available transmission capacity in favor of a powerful coding protection level.
3.2. **PHY and network layer interactions**

In the IP traffic management, user mobility should be adequately taken into account. In particular, efficient mobility management protocols have to be employed at layer 3 to prevent excessive delays incurred in rerouting the IP data flows during handoff phases. In case of non-GEO satellites, mobility management protocols could use the predictable motion of satellites in their constellation to define in advance the cell where the user is moving in order to reduce the rerouting delay to a minimum.

3.3. **MAC and network layer interactions**

In IP-based networks, QoS provision can be based on two approaches [7]: Integrated Services (IntServ) and Differentiated Services (DiffServ). The IntServ concept is to reserve resources (through the RSVP protocol) for each flow through the network, whereas the DiffServ approach achieves scalability by aggregating traffic flows into classes that are conveyed by means of IP-layer packet marking, using the Type of Service (ToS) field in the IPv4 header or the Differentiated Service (DS) field in the IPv6 header. In both IntServ and DiffServ cases, layer 3 queues should be adequately mapped to layer 2 queues. Moreover, it is important that the resource allocation scheme at layer 2 manages traffic in a way that is compatible with the QoS support operated at layer 3.

3.4. **Interactions among MAC, network layer and transport layer**

Currently, TCP is the dominant transport layer protocol; it is used for controlling the data sending rates on the Internet and for assuring a reliable end-to-end delivery of data. The standard TCP congestion control mechanism is known to perform poorly over satellite links, due to both the large Round-Trip Time (RTT) and the high packet error rates (PERs). In the classical TCP version the loss of a packet causes an expiration of the Retransmission Time Out (RTO) with TCP coming back to the slow start phase with a sudden reduction of the data injection rate. While in wired networks packet losses are due to buffer overflows and the data rate reduction is an appropriate countermeasure to losses, in the wireless and satellite scenarios, frequent packet losses are caused by the unreliable radio medium. Hence, the reduction of the data rate in response to packet losses in the satellite scenario causes a low TCP goodput and a low utilization of satellite links. Suitable TCP versions have been proposed to improve the TCP goodput in these cases (see Section 4).

Different cross-layer mechanisms can be used in a satellite IP network to improve the performance of the TCP protocol. Three interesting examples are detailed below [3].

- In DVB-S/DVB-RCS systems, a Demand Assignment Multiple Access (DAMA) should be operated at layer 2 where bandwidth requests are based on the TCP behavior in order to reduce the queuing delay and congestion phenomena that may cause RTO expirations at layer 4.
- In split scenarios, the end-to-end TCP semantics is broken and interconnecting Performance Enhancement Proxies (PEPs) are used at the earth gateways that close the TCP flows. PEPs shield high-latency and error-prone satellite network segments from the rest of the network, in a transparent way to applications. A critical issue for PEPs is the design of buffer size and related management rules. Interesting proposals envisage the adoption of Active Queue Management (AQM) at the MAC layer: when a
router determines that the bandwidth is fully utilized, packets are dropped even when the queue is not filled to reduce the buffer congestion and the data injection rate of the TCP sender.

- Finally, Explicit Congestion Notification (ECN) signaling could be provided by layer 3 to higher layers. In this way, TCP could distinguish between packet losses due to congestion and packet losses due to errors on the wireless link. Note that Next-Generation TCP/IP protocol suite of Windows® supports improved functionalities and, among others, also ECN.

### 3.5. Interactions among PHY, MAC and higher-layer protocols

A monitoring action should be jointly performed by application and MAC layers in order to control adaptively the service priority (top-down approach, explicit cross-layer design). Conversely, the adaptation of MODCOD levels employed at the PHY level with ACM should be fed back to the application layer to change dynamically the source generation bit rate in case of streaming services (bottom-up approach, explicit cross-layer design). Such an adaptation in the source coding is possible with the new H.263 and H.264 codecs.

In an IP-based satellite network, IP datagrams suffer from residual errors provided by the reassembly of layer 2 packets. Note that IP datagrams are discarded in a router if the header checksum verification fails; once discarded, layer 3 does not attempt any recovery procedure, since this is the task of higher layers. Hence, a receiver of a satellite link experiences an erasure channel at layer 3 or above. Consequently, end-to-end coding suitable for erasure channels can be employed (at transport or application layer) to recover these errors. In such a case, a joint optimization (implicit cross-layer scheme) of higher-layer coding and lower-layer coding is needed in order to avoid redundancies and inefficiencies in the protocol stack architecture.

Finally, the selection of different MODCODs at the physical layer (PHY) could be related to the goodput at layer 4, in the presence of TCP-based applications. This approach is important, since different TCP versions require distinct optimizations of the trade-off between higher information bit rates and lower coding protection that can be achieved by increasing the MODCOD level (i.e. reducing the coding protection in favor of a higher-order modulation). This optimization study is quite interesting and will be addressed in the next sections of this paper.

### 4. SYSTEM MODEL

GEO satellite links have a unique combination of characteristics that affect the throughput of TCP-based traffic; particularly, critical elements are both the large RTT and the high PER value. Recently, much effort has been devoted by the scientific community in the design of TCP enhancements, expressly defined for networks with large Bandwidth Delay Product (BDP) values. The interest of this section is to focus on the TCP performance as a result of interactions with lower-layer protocols and, in particular, MODCOD levels. In this study, we have used different TCP versions supporting an explicit treatment of packet losses due to errors produced by the radio channel. In particular, we have considered [8–11]: TCP NewReno, SACK, Westwood+ and Hybla.
Note that TCP Reno suffers from multiple packet losses in a window of data (this is particularly important in satellite networks where the high BDP value entails the use of large windows): the sender must wait for RTO to trigger the retransmission of lost packets. A possible improvement to cope with multiple losses in a window of data is represented by the TCP NewReno version [8] that modifies the fast recovery phase of TCP Reno. TCP NewReno can recover from multiple lost packets (without waiting for the RTO expiration) by using the ‘partial acknowledgments’ (ACKs) received during the fast recovery phase. As soon as a loss is identified, the fast retransmit algorithm is started by resending the lost packet. Then, the fast recovery algorithm manages the ACKs as follows: (i) if there has been a single lost packet, the ACK refers to all the packets transmitted up to the fast retransmit phase; (ii) otherwise, the ACK is partial and acknowledges some of the packets sent before the fast retransmit phase. In this study, we refer to the TCP NewReno Impatient version that resets the retransmit timer only after the first partial ACK received [8]; this choice causes an RTO to occur soon when there are many losses in a window of data (typically in the starting phases of a connection).

Another improvement to deal with multiple packet losses in a window of data is the adoption of Selective ACKnowledgement (SACK) [9]: the receiver informs the sender on the successfully received packets, so that the sender can re-transmit just the actually lost packets. SACK can be implemented with both fast-recovery and fast-retransmit algorithms of the TCP NewReno version.

Differently from TCP, Reno and NewReno reduce the congestion window \((cwnd)\) to one-half after three duplicated ACKs; TCP Westwood (and the recent Westwood+ version [10]) sets \(cwnd\) and the slow start threshold \((ssthresh)\) after a loss episode on the basis of an end-to-end bandwidth estimate, \(B_{we}\), made before the loss event was detected; in particular, \(ssthresh\) is made equal to \(B_{we} \times RTT\). Hence, TCP Westwood avoids a conservative \(cwnd\) reduction, thus allowing a faster-recovery phase. A similar modification is made when RTO expires: \(ssthresh = B_{we} \times RTT\) and \(cwnd\) is reset to its initial value. The bandwidth \(B_{we}\) of a connection is continuously estimated by considering the amount of data sent and the ACK interarrival time. Then, these sample values are averaged by a low-pass filter, since only lower frequencies of the input traffic rate may lead to congestion.

TCP Hybla has been conceived to address the problems of typical satellite connections, such as high propagation delays and high PER values. In heterogeneous networks, TCP connections characterized by large RTT (i.e. including a satellite segment) present poor performance as compared to wired connections with shorter RTT. In fact, since the congestion window growth depends on the reception of ACKs, a large RTT value affects the throughput and the channel utilization. To reduce the TCP goodput differences, TCP Hybla [11] proposes a modification to the \(cwnd\) update algorithm on the receipt of an ACK in order to accelerate the \(cwnd\) increase in both slow-start and congestion-avoidance phases for connections with large RTT values. Hybla foresees the adoption of SACK in order to recover quickly from losses due to both channel errors and the more aggressive injection of data in the network.

In order to carry out our study on cross-layer interactions between PHY and transport layers, we have considered a simple network architecture featuring a GEO bent-pipe satellite; as shown in Figure 3, a client (TCP receiver) is connected to a router-earth station that is connected via satellite to another router-earth station that is linked to a remote server (TCP sender) from which files are downloaded. A return channel via satellite is used to send both lower layer signaling (i.e. link quality estimation for MODCOD adaptation) and layer 4 ACKs.
The GEO satellite link has a raw bandwidth of 2 MHz for both uplink and downlink directions. The Ka-band (20–30 GHz) is used. Two MODCOD options are considered at PHY, as detailed below, referring to a DVB-S-like system with adaptivity. In this study we have not modeled framing at layer 2: the selection of the MODCOD at PHY determines the information bit rate that is available to convey IP datagrams (i.e. a simple Constant Rate Assignment). A further analysis that also considers layer 2 framing and the joint impact of ACM and BoD is left to a future study.

The earth station-to-earth station one-way propagation delay is about 260 ms. The terrestrial links from the earth station and the client and from the other earth station to the server are at 30 Mbit/s with a (one-way) propagation delay of 10 ms. Hence, the satellite link is the system bottleneck and the propagation delay contribution to RTT is equal to 560 ms.

In our study we assume: (i) earth stations with a Line-of-Sight (LoS) path to the GEO satellite; (ii) a memoryless channel with uncorrelated losses; (iii) attenuation fluctuation only due to slow troposphere events (i.e. long-term variations of the received signal strength due to cloud attenuation and rain fades, no shadowing) so that the channel can be considered of the Additive White Gaussian Noise (AWGN) type; (iv) residual packet losses after the decoding process are uncorrelated and occur according to a given PER at the transport level; (v) channel variations are very slow compared with the delay of the feedback signal to inform the sending earth station to modify its transmission mode; (vi) no Automatic ReQuest repeat (ARQ) technique is used at layer 2, since the adoption of an ARQ scheme in our architecture would entail a too high delay to recover packet losses that could cause an RTO expiration at the TCP level.

Figure 3. Reference network architecture.

We will also perform simulations for a second case study where the available bandwidth on the satellite link is augmented to 20 MHz. Correspondingly, also the buffer capacities are increased by 10 times. This study will be only intended to evaluate the trend due to bandwidth increase.
We study the goodput performance for the transmitted packets at the TCP level. An FTP application (persistent TCP connection) is assumed that produces TCP traffic according to an ACK-clocked model. ACM is used in this study for the satellite link in order to provide an acceptable quality in different radio channel conditions. The transmission mode (i.e. MODCOD) adaptation is performed by the sending earth station on the basis of the channel quality measure made by the receiving earth station. We have adopted two modulations: BPSK and QPSK. Moreover, we have employed a convolutional coder/Viterbi decoder; in particular, the standard NASA 1/2 rate convolutional code with constraint length 7 and derived punctured code with rate 3/4 [12]. We consider two distinct transmission modes (MODCOD) that are operated at parity of transmission bandwidth (2 MHz) and power. In particular, we have the following:

- **Mode #1**: BPSK with rate 1/2 convolutional encoder and resulting information bit rate of 1 Mbit/s.
- **Mode #2**: QPSK with rate 3/4 convolutional encoder and resulting information bit rate of 3 Mbit/s.

A packet length of 1500 bytes has been adopted (this is the maximum packet length allowed by the Ethernet standard). Therefore, the BDP product is 560 000 bits (about 47 packets) for mode #1 and 1 680 000 bits (140 packets) for mode #2. Each router has a buffer with a capacity of packets equal to BDP for mode #2 (i.e. the maximum BDP value between modes #1 and #2); these queues adopt a ‘droptail’ policy to manage the congestion (i.e. arriving packets are lost according to the buffer capacity).

Figure 4 provides the PER performance as a function of $E_b/N_0$ (at the level of coded bits for the BPSK case, i.e. the reference $E_b/N_0$ value for short) for the two transmission modes.

![PER performance for the two transmission modes as a function of $E_b/N_0$ (reference value for BPSK).](image-url-pattern)
envisaged in this study. This graph also shows a possible selection between modes #1 and #2 on the basis of a PER requirement.

5. PERFORMANCE RESULTS

In order to perform our simulations, we have used the patch implemented by D. Wei and P. Cao for the ns-2 2.29 version [13]. This patch permits to run different Linux TCP congestion control algorithms on ns-2 (e.g. TCP-Hybla, Westwood+, NewReno, and SACK), with similar simulation speed and memory usages as other ns-2-based TCP versions. In this way, we can obtain results that closely reproduce the behavior of real TCP connections. We have implemented the scenario described in Section 5 (see Figure 3) under the ns-2 environment. Simulations have been performed for static radio channel conditions, i.e. assuming links with fixed $E_b/N_0$ all over the simulation length.\(^8\) We are interested in defining an optimal $E_b/N_0$ threshold (PHY) for the selection between modes #1 and #2. Two criteria are compared:

- **Scheme a**: We select the transmission mode on the basis of a PER threshold value (i.e. non-cross-layer, classical approach for current air interfaces with ACM).
- **Scheme b**: We select the transmission mode with the aim of maximizing the goodput performance at the transport layer (implicit cross-layer approach, top-down method).

Note that, in switching from modes #1 to #2, we have a larger bandwidth versus a higher PER value. Thus, the most convenient switching point (in terms of $E_b/N_0$) depends on a suitable trade-off between these two conflicting aspects for what concerns the TCP performance. Such a trade-off depends on the adopted TCP version.

Referring to the classical scheme a, Figure 5 shows the TCP goodput as a function of the $E_b/N_0$ value considering a PER switching threshold equal to either $10^{-1}$ (upper graph) or $10^{-4}$ (lower graph). In both cases, we can see that such selection criterion is inefficient. In fact in the upper graph the switching point is too anticipated for some TCP versions, thus resulting in a drop of performance when changing from transmission mode #1 to transmission mode #2. Whereas, in the lower graph, the switching point is too delayed so that the system is constrained to still use transmission mode #1 when mode #2 would be more convenient; also in this case there is not an efficient utilization of resources. The results in Figure 5 prove that an appropriate selection criterion between modes #1 and #2 should not be based on PER performance but rather on the transport layer performance and be directly related to $E_b/N_0$ (cross-layer approach). This is the reason why we consider the selection scheme b here. The related results are shown in Figure 6; the corresponding switching points are detailed in Table I.

Before further discussing about the results in Figure 6 and Table I, it is important to capture, according to the description made in Section 4, the distinct behaviors of different TCP versions in response to packet losses:

- NewReno drops $cwnd$ and hence the packet injection rate in the network and recovers a packet per RTT.

\(^8\)This approach is suitable for determining the optimal MODCOD configuration, considering different possibilities for the reference (nominal) $E_b/N_0$ value. A further study would be required to perform a dynamic adaptation on the basis of varying channel conditions around the nominal, clear-sky ones. Some considerations on this issue will be provided at the end of this section.
Figure 5. Simulation results for the different TCP versions as a function of the $E_b/N_0$ (reference value for BPSK) in the case with selection criterion $a$; the maximum 95% confidence interval amplitude for all the curves is $\pm 7\%$.

Figure 6. Simulation results for the different TCP versions as a function of the $E_b/N_0$ (reference value for BPSK) in the case with adaptive modes 1 and 2 (selection criterion $b$); the maximum 95% confidence interval amplitude for all the curves is $\pm 7\%$. 
SACK can recover multiple packet losses in an RTT.

Westwood+ restores $cwnd$ at a value corresponding to the capacity estimated for the end-to-end connection.

Hybla allows a very fast $cwnd$ increase in a satellite scenario after a packet drop to compensate for the high propagation delays.

Hence, these different TCP versions have a different robustness to the packet errors introduced by the radio channel; these characteristics should be made available in some form to the PHY layer to allow an appropriate selection between the modes according to Table I. In particular, from the results in Figure 6 and Table I, we can note that Hybla and Westwood+ achieve the best performance, being particularly efficient in recovering from packet loss events; while, lower goodput values are obtained with SACK and TCP NewReno. Moreover, from Figure 6 we can note that as $E_b/N_0$ increases (thus, PER reduces), we arrive at a situation where the TCP goodput saturates. Note that the TCP throughput as well as this saturation effect can be explained by means of the well-known square-root formula [14] that is valid assuming a steady-state condition and PER lower than 1%:

$$\eta_t = \min\left\{ \frac{l}{RTT} \sqrt{\frac{\alpha}{PER\cdot IBR}} \right\} \text{[bit/s]}$$

where $l$ is the TCP segment length in bits (i.e. 12000 bits in our case), $\text{IBR} = \text{IBR(mode#)}$ is the information bit rate experienced at the TCP level that depends on the transmission mode, $\alpha$ for TCP NewReno is a constant equal to 1.5 since an ACK is sent for every received packet, $\text{PER} = \text{PER (mode#, } E_b/N_0)$, $\text{RTT} = d + l/\text{IBR}$ and $d$ is the network RTT, here approximated by the round-trip propagation delay of 560 ms.

The TCP goodput (i.e. the TCP throughput at the receiver) can be obtained by multiplying (1) by a factor equal to $1 - \text{PER}$ in order to take into account the packet losses due to the channel.

On the basis of (1), as $\text{PER}$ reduces the left term in the minimum function increases and would go over $\text{IBR}$ that of course represents the maximum throughput value. The limiting $\text{PER}$ value, $\text{PER}^*$, while permits exploiting the available bandwidth (saturation condition) fully can be determined by equating the two terms in the minimum in (1) and then solving with respect to $\text{PER}$. By considering $\text{BDP} = \text{IBR} \times \text{RTT}/l$ and $\alpha = 1.5$ for TCP NewReno, we have

$$\frac{l}{RTT} \sqrt{\frac{1.5}{\text{PER}^*}} = \text{IBR} \Rightarrow \text{PER}^* = \frac{1.5}{\text{BDP}^2}$$

On the basis of (2), TCP NewReno is able to exploit fully the available bandwidth for $\text{PER} \leq \text{PER}^*$, where $\text{PER}^*$ corresponds to $6.8 \times 10^{-4}$ for mode #1 (i.e. $E_b/N_0 \approx \sim 3$ dB) and to $7.65 \times 10^{-5}$ for mode #2 (i.e. $E_b/N_0 \approx \sim 8$ dB).

<table>
<thead>
<tr>
<th>TCP version</th>
<th>$E_b/N_0$ switching threshold (dB)</th>
<th>PER(mode#2)</th>
<th>BER(mode#2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hybla</td>
<td>6</td>
<td>$8 \times 10^{-2}$</td>
<td>$4.3 \times 10^{-5}$</td>
</tr>
<tr>
<td>Westwood+</td>
<td>6.2</td>
<td>$4.2 \times 10^{-2}$</td>
<td>$2 \times 10^{-5}$</td>
</tr>
<tr>
<td>NewReno</td>
<td>7.5</td>
<td>$5.9 \times 10^{-4}$</td>
<td>$1.7 \times 10^{-7}$</td>
</tr>
<tr>
<td>SACK</td>
<td>7.5</td>
<td>$5.9 \times 10^{-4}$</td>
<td>$1.7 \times 10^{-7}$</td>
</tr>
</tbody>
</table>

Table I. Description of the $E_b/N_0$ switching thresholds for different TCP versions (selection scheme $b$).
Finally, the graph in Figure 7 provides the TPC goodput behavior (with selection criterion $b$, implicit cross-layer approach) for a different available satellite bandwidth (i.e. 20 MHz) that corresponds to an information bit rate of 10 Mbit/s in mode #1 and 30 Mbit/s in mode #2. We have performed these simulations to investigate the impact on the switching point between the two transmission modes due to the increase of a factor of 10 in the available bandwidth. This study permits the capturing of the trend of this cross-layer technique when more capacity would be available. From this graph, we can see that the switching thresholds have changed with respect to the corresponding ones in Figure 6, especially for Westwood+, NewReno and SACK. For instance, referring to NewReno, the $E_b/N_0$ switching threshold was 7.5 dB with 1–3 Mbit/s transmissions and now it is 8.1 dB with 10–30 Mbit/s transmissions. Note that the presence of UDP traffic in conjunction with the TCP one could only reduce the capacity available for TCP. This practically would simply entail the shift of the curves in Figures 6 and 7 towards lower TCP goodput values.

The graphs in Figures 6 and 7 can permit the determination of the most convenient transmission mode for different $E_b/N_0$ values and diverse TCP versions. The actual mean goodput for a given TCP version depends on the dynamic behavior of $E_b/N_0$ due to meteorological events. We can account for this by using either channel traces or the related statistics, i.e. the distribution of the $E_b/N_0$ value. Such distribution depends on meteorological phenomena, the link budget, the antenna pattern, and the position of the earth stations. For

\[\text{Note that in such conditions, the system should update the transmission mode (i.e. modes #1 or #2) on a regular time basis corresponding to the superframe duration (DVB-S/-RCS case).}\]
instance, Figure 8 describes the $E_b/N_0$ distribution due to rain fades (also with scintillation effects at lower power levels) for Ka-band transmissions, obtained from simulations on the basis of the Matricciani’s model [15]. In particular, this graph assumes a reference $E_b/N_0$ value due to link budget design (i.e. 0 dB degradation value for $E_b/N_0$ in abscissa) and considers the $E_b/N_0$ degradation due to rain. Distributions like this one could be used ‘to weight’ the results in Figures 6 and 7 on the basis of a ‘reference’ $E_b/N_0$ value due to system design. The reference $E_b/N_0$ value is a key parameter of the satellite link that depends on the location of the receiver, the available transmission power, antenna types and can be referred to clear sky conditions. Note that the approach that uses the $E_b/N_0$ distribution due to meteorological phenomena as a weight for the curves in graphs like those in Figures 6 and 7 does not take into account the correlation we have in the $E_b/N_0$ behavior. Such a correlation is expected to have an impact on the transport layer performance (typically, uncorrelated losses are a worst-case assumption). Anyway, if we can assume that the $E_b/N_0$ variation is much slower than the bit duration and TCP packet length (as it occurs for meteorological phenomena), we can assume that TCP reaches the equilibrium in the different channel conditions (the transmission is done in quasi-static conditions for the channel) and the approach that uses the weights derived from the $E_b/N_0$ distribution is acceptable. Hence, according to the distribution shown in Figure 8 we can see that the most significant weight is given to the ‘reference’ $E_b/N_0$ value. In conclusion, the weighting of the TCP goodput curves in Figures 6 and 7 due to the $E_b/N_0$ distribution in Figure 8 practically does not modify the optimal switching point.

The above considerations do not explicitly address the case of a mixed population of users experiencing different channel conditions. In such a case, we may assume that the system divides the resource among groups of users with similar channel conditions, so that for each group we

Figure 8. Distribution of the $E_b/N_0$ degradation due to meteorological phenomena for Ka-band transmissions (being the 0 dB level the reference one, i.e. clear sky conditions).
can re-apply the considerations made above. A further investigation of these aspects is beyond the scope of this paper.

In conclusion, our simulation results demonstrate that the TCP-driven selection of the PHY transmission modes can permit the improvement of the higher-layer goodput (as well as the user satisfaction) in the satellite network scenario.

6. CONCLUSIONS

Cross-layer design is a novel research field that is particularly important to improve the performance of wireless networks and, in particular, IP-based satellite communication systems. This paper has proven that a wide-scale adaptivity as that allowed by cross-layer design can permit the provision of better QoS to users while guaranteeing an efficient use of the scarcely available satellite resources. A case study has been proposed in this paper in order to investigate the interactions between mode adaptivity at layer 1 and TCP goodput behavior at layer 4 in the presence of a satellite error-prone channel. We have obtained an important result showing that the PHY mode selection scheme should be driven by the TPC goodput performance. This interesting outcome is an important initial step to stimulate research in cross-layer techniques even on a larger scale still with a cautionary approach to maintain the basic aspects of the modularity approach of the OSI protocol reference model.

The cross-layer air interface design is an interdisciplinary field that warrants research of new methods and techniques. Future research includes impact analysis of multimedia applications performance due to lower layer protocol selection. In addition, simulation analysis for broadcast standards such as DVB-S2 and DVB-RCS will be evaluated in GEO satellite systems supporting multimedia services.

APPENDIX: LIST OF ACRONYMS

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>3GPP</td>
<td>Third-Generation Partnership Project</td>
</tr>
<tr>
<td>ACK</td>
<td>Acknowledgement</td>
</tr>
<tr>
<td>ACM</td>
<td>Adaptive Coding and Modulation</td>
</tr>
<tr>
<td>AF</td>
<td>Assured Forwarding</td>
</tr>
<tr>
<td>APSK</td>
<td>Amplitude and Phase Shift Keying</td>
</tr>
<tr>
<td>AQM</td>
<td>Active Queue Management</td>
</tr>
<tr>
<td>ARO</td>
<td>Automatic Repeat ReQuest</td>
</tr>
<tr>
<td>AWGN</td>
<td>Additive White Gaussian Noise</td>
</tr>
<tr>
<td>BDP</td>
<td>Bandwidth Delay Product</td>
</tr>
<tr>
<td>BoD</td>
<td>Bandwidth-on-Demand</td>
</tr>
<tr>
<td>CRC</td>
<td>Cyclic Redundancy Check</td>
</tr>
<tr>
<td>DAMA</td>
<td>Demand-Assignment Multiple Access</td>
</tr>
<tr>
<td>DS</td>
<td>Differentiated Service (field)</td>
</tr>
<tr>
<td>DVB-RCS</td>
<td>Digital Video Broadcasting-Return Channel via Satellite</td>
</tr>
<tr>
<td>DVB-S</td>
<td>Digital Video Broadcasting-Satellite</td>
</tr>
<tr>
<td>DVB-S2</td>
<td>Digital Video Broadcasting-Satellite 2</td>
</tr>
<tr>
<td>ECN</td>
<td>Explicit Congestion Notification</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

The authors wish to thank Dr Nedo Celandroni (CNR-ISTI, Italy) for providing the material concerning the performance of NASA convolutional codes. A special acknowledgment has to be given to the SatNEx II project (IST-027393, http://www.satnex.org/) that has made possible part of the research work presented in this paper. Finally, the authors also wish to thank Dr Matteo Berioli (DLR, Germany) for his support on Ka-band channel data.

REFERENCES

1. ETSI. Digital video broadcasting (DVB); interaction channel for satellite distribution systems. EN 301 790.


6. ETSI. Satellite earth stations and systems (SES); broadband satellite multimedia (BSM) services and architectures; functional architecture for IP interworking with BSM networks. *TS 102 292, V1.1.1*, 2004–2002.


**AUTHORS’ BIOGRAPHIES**

**Sastri Kota** is a Senior Scientist in Harris Corporation and Adjunct Professor in University of Oulu, Finland. He has held technical and management positions at Loral, Lockheed Martin, SRI International, the MITRE Corp, and Xerox Corp, contributing to military and commercial satellite communication systems and broadband network design and analysis. Dr Kota received his PhD from the University of Oulu, Finland; Electrical Engineer’s Degree from Northeastern University, Boston, MA; MSEE from IIT; BSEE from BITS, India. He is the principal author of a book ‘Broadband Satellite Communications for Internet Access’, co-edited book ‘Emerging Location Aware Broadband Wireless Ad Hoc Networks’, and wrote book chapters on Satellite TCP/IP and Broadband Networks. He has published over 120 papers and served in various capacities in IEEE, AIAA, and ACM conferences. He is member of the editorial boards on the International Journal of Satellite Communications and Networking and the Space Communications Journal. He is a Senior Member of IEEE, Associate Fellow of AIAA, and Member of ACM.

**Giovanni Giambene** received the Dr. Ing. degree in Electronics in 1993 and the PhD 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 ‘MultimedialitA’, 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).

Mr Nico Liberato Candio was born in Putignano, Italy, in 1972. He received the Bachelor Degree in Telecommunications Engineering in 2006 from the University of Siena, Italy, with a thesis entitled ‘TCP-Hybla: performance measures in broad band networks’. From April 2006 he is carrying on research activities with the University of Siena—CNIT research unit. His research interests include: satellite communications, TCP benchmarking and efficiency, mobile and wireless networks. At present, Mr N. L. Candio works as software engineer with Motorola Technology Center, Turin, Italy.