Cross-layer Management of Radio Resources in an Interactive DVB-RCS-based Satellite Network

Paolo Chini, Giovanni Giambene[1] and Danilo Bartolini, Michele Luglio, Cesare Roseti[2]

[1] CNIT - Dipartimento di Ingegneria dell'Informazione - Università degli Studi di Siena, Via Roma, 56 - 53100 Siena, ITALY,
`giambene@unisi.it`,
[2] Università degli Studi di Roma "Tor Vergata", Via del Politecnico, 1 - 00133 Roma, ITALY
{`luglio` , `cesare.roseti`@uniroma2.it}

Index Terms : *Satellite Networks, DVB-RCS, TCP.*

Abstract. Recently the request for multimedia services has been rapidly increasing and satellite networks appear to be attractive for a fast service deployment, for providing broadband access and to extend the typical service area of terrestrial systems. Due to both the intrinsic propagation delay and the presence of the lossy radio channel, the *Transport Control Protocol* (TCP) is particularly inefficient in the satellite scenario. We envisage a group of terminals that have to transmit (uplink) to a *Network Control Center* (NCC) via a geostationary bent-pipe satellite; the NCC is interconnected to the Internet through a router. Terminals employ the DVB-RCS standard to communicate with the NCC. The NCC centrally coordinates the resource allocation for uplink transmissions. The focus of this paper is on the proposal of a novel mechanism where resources are allocated by the *Medium Access Control* (MAC) layer at the NCC depending on the behavior of the TCP congestion window of the remote terminals. In addition to this, the MAC layer can also intervene on the TPC behavior to prevent system congestion. The interest is here in optimizing the air interface by envisaging a cross-layer interaction between the protocols of layer 4 and layer 2. The obtained results show that our scheme permits to prevent the occurrence of TCP time outs (due to the satellite network congestion), thus improving the utilization of radio resources as well as the throughput perceived by users. Moreover, referring to the ftp application (elephant connections), we have proved that our technique allows reducing the mean file transfer time with respect to a classical allocation scheme. This paper has been carried out within A&TCP research group of the "SatNEx" NoE project (URL: www.satnex.org; contract No. 507052) within the 6-th framework of the European Commission.

1 Introduction

Satellite communications have an important role for telephone communications, broadcasting, computer communications, etc. They can be the sole communication medium in unaccessible regions on the earth or in the Oceans. Moreover, satellite networks are the best candidate to allow a fast provision of broadband communications in different areas.

Towards the full protocol integration between terrestrial and satellite communication systems, it is important to study techniques to support TCP traffic with adequate efficiency and throughput also in satellite systems [1]. Typical characteristics of TCP are the gentle probe of system resources by progressively increasing the injection rate of data in the network by using sliding windows with dynamically adapted width; in such mechanism, a fundamental role is played by the *congestion window* (*cwnd*) [2]. This approach is needed since the network is a 'black box' and the TCP sender has no mean to know the network congestion status before sending data. Congestion is revealed by the loss of TCP segments and consequent expiration of TCP time outs. Consequently, the *cwnd* value is reset to its initial value (typically 1 or 2 TCP segments) with a sudden and significant throughput reduction at the TCP level. Hence, the TCP congestion control scheme leads to well-know inefficiencies in the satellite scenario due to both the high *Round-Trip Times* (RTT) and the frequent errors (with related losses of data) on the radio channel. In particular, TCP does not distinguish between segment losses due to network congestion and those caused by the radio link. Whereas, in the presence of a radio channel, many errors will be produced with consequent frequent *Retransmission Time Out* (RTO) expirations that drastically reduce the throughput. However, these losses are not due to network congestion and the *cwnd* reduction is not appropriate.

Satellites are typically bandwidth and power limited. Referring to TCP-based applications, we have that the traffic injected in the network is time-varying just due to the *cwnd* mechanism: *cwnd* can increase on an RTT basis (slow start or congestion avoidance case) or suddenly decrease in the presence of segment losses. On the basis of the above, a fixed bandwidth allocation to a given TCP flow is particulary inefficient. Also a classical dynamic channel allocation (considering the current state of radio resource allocation and the behavior of the physical layer) can be quite inefficient, being unable to track the TCP behavior. To overcome these difficulties and to achieve a better utilization of resources, we propose here a novel type of dynamic resource allocation centrally controlled by a *Network Control Center* (NCC) that operates as follows [3],[4]:

- As far as resources (layer 2) are available, the NCC tries to allocate transmission uplink resources to a terminal so as to follow the behavior of its *cwnd* (layer 4).
- When all resources are allocated, the NCC (layer 2) stops a further increase in *cwnd* (layer 4); otherwise, resources need to be rearranged among the active connections depending on their priority levels.

According to the above, at the remote terminals we have envisaged a cross-layer dialogue so at to make resource requests to the NCC that are TCP-aware (interaction from layer 4 to layer 2). The NCC allocates resources to terminals on the basis of these requests; in case of congestion the NCC is alsp able to temporarily stop a further increase in the *cwnd* value of remote terminals. Hence, there is a twofold cross-layer interaction in our scheme [5],[6]. We have obtained that such method permits to reduce the occurrence of TPC time outs due to congestion (i.e., delays experienced by TCP traffic due to the unavailability of layer 2 resources assigned at the MAC layer), since congestion in the satellite network is signaled in advance. Moreover, the available resources are better utilized, thus reducing the mean delay to transfer files with the ftp application (elephant connections).

2 System Architecture

In this paper we focus on an *Interactive Satellite Network* (ISN), based on the DVB-RCS standard [3],[4]. In particular, we consider a *GEOstationary* (GEO) bent-pipe satellite, *Return Channel Satellite Terminals* (RCSTs) and a *Network Control Center* (NCC) that is connected to a router to access the Internet (see Fig. 1). RCSTs are fixed and use the *Return Channel via Satellite* (RCS) that allows transmitting data or signaling. The NCC is the heart of the network: it provides control and monitoring functions and it manages network resources allocation according to a *Dynamic Bandwidth Allocation* (DBA) approach.

Typically, the RCST is connected to a local network where different terminals are present. For the sake of simplicity, in the following analysis we refer to a single terminal (user) connected per RCST; the extension of such study to the case of multiple terminals per RCST generating concurrent TCP flows is feasible[3], but beyond the scope of this paper where we are interested to prove the importance of the cross-layer interaction between layer 2 and layer 4 to achieve both a higher TPC throughout (layer 4) and an efficient utilization of resources (layer 2). In such a scenario, it is also possible to employ a *Performance Enhancing Proxy* (PEP) and to have a split of the TCP connection to be realized at the NCC [7]. However, we focus here on an end-to-end solution leaving the comparison with the PEP approach to a further work. While it is widely acknowledged that the deployment of split-based solutions may lead to a considerable performance improvement, we lost the typical end-to-end semantics of TCP causing congestion at the intermediate buffers and the risk of undelivered packets [8]. Finally, the IPsec protocol for secure transmissions requires that TCP operates end-to-end.

[3] In such a case, we may expect that the RCST aggregates the traffic needs coming from the current *cwnd* values of its different TCP flows; when congestion occurs, the RCST receives from the NCC a notification to reduce the increase in the injection of traffic, so that the RCST can stop the increase of the *cwnd* values according to a local criterion.

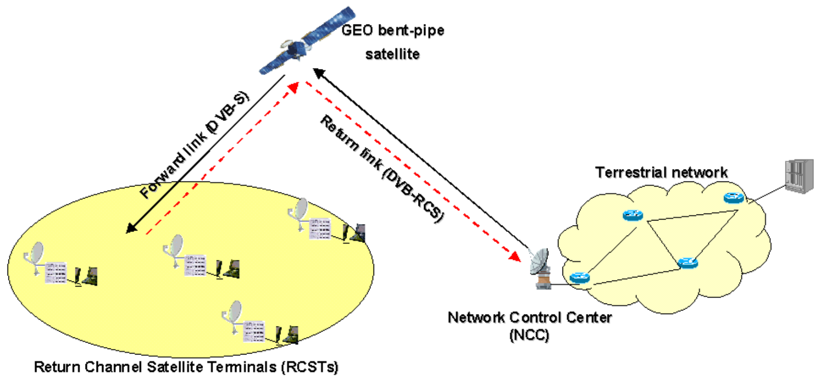

Fig. 1. System architecture.

3 DVB-S and DVB-RCS Air Interface Resources

DVB-S is used for the forward link (from NCC to RCSTs) and DVB-RCS is employed for the return link (from RCSTs to NCC); in both cases, a QPSK modulation is used. DVB-S can achieve a (layer 2) maximum data rate of 38 Mbit/s, whereas DVB-RCS has a maximum data rate of 2 Mbit/s.

DVB-S has been conceived for primary and secondary distribution (*Fixed Satellite Service*, FSS) and *Broadcast Satellite Service* (BSS), both operated in the Ku (11/12 GHz) band [9]. Such system is used to provide both a direct reception from the satellite (*Direct-To-Home*, DTH) for a single user with an integrated receiver-decoder and a collective access.

Below the transport layer and the IP layer we have the *Multi Protocol Encapsulation* (MPE) that provides segmentation & reassembly functions for the production of MPEG2-TS (*Transport Stream*) packets of fixed length (188 bytes) that are transmitted in a time division multiplexing way. To the block of data coming from the application layer, a TCP header of 20 bytes, an IP header of 20 bytes and an MPE header+CRC trailer of 12+4 bytes are added; such bytes are fragmented in the payloads of MPEG2-TS packets. Then, transmission is performed according to several steps, such as: channel coding (external Reed-Solomon coding, convolutional interleaver, internal convolutional coding, puncturing), baseband shaping of impulse, QPSK modulation.

DVB-RCS is used to support the following services:

- Classical Internet-based services, such as email, file transfer, Web-browsing, etc;
- Multicast traffic;
- "Voice over IP" (VoIP).

Note that according to the above list, VoIP is based on the UDP protocol, whereas multicast applications typically adopt *ad hoc* layer 4 protocols. Hence, only the first category of services is TCP-based and will be the target of this study.

The DVB-RCS air interface is of the *Multi Frequency - Time Division Multiple Access* (MF-TDMA) type: resources are time slots on different available carrier frequencies with different possible available bandwidths. DVB-RCS resources are divided in super-frames that are characterized by suitable portions of time and frequency bands. The DVB-RCS standard considers that the maximum super-frame length is 93.2 s. Each super-frame is divided in frames that are composed of time slots; the frames can have different duration, bandwidth and composition of time slots. Slots can be assigned to an RCST belonging to the same frequency or to different frequencies with the constraint that only one frequency at once can be transmitted (see Fig. 2).

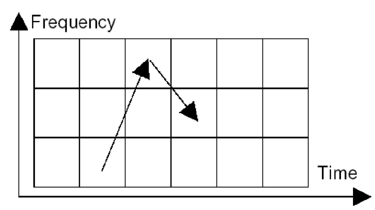

Fig. 2. Use of MF-TDMA slot resources on different carrier frequencies.

The NCC assigns to each RCST a group of bursts (i.e., the related slots), each of them characterized by a frequency, a bandwidth a start time and a duration. The NCC communicates resource allocations to RCSTs through the *Terminal Burst Time Plan* (TBTP), an informative message belonging to the set of *System Information* (SI) tables[4]. Note that TBTP can be updated (resent) at every super-frame.

The DVB-RCS standard envisages 5 types of resource allocation:

- *Continuous Rate Assignment* (CRA),
- *Rate Based Dynamic Capacity* (RBDC),
- *Volume Based Dynamic Capacity*(VBDC),
- *Absolute Volume Based Dynamic Capacity* (AVBDC),
- *Free Capacity Assignment* (FCA).

Whereas in the former 4 categories a quantity of slots (if available) is assigned to the RCSTs according to their requests (a fixed quantity in CRA or a variable one in the other 4 cases), the FCA assignment strategy allocates the resources not used in a given super-frame (after the satisfaction of the other types of assignment requests), without explicit requests made by the RCSTs (a sort of *bonus* to reduce possible delays).

[4] The TBTP is a message sent by the NCC to a group of RCSTs or to a single RCST to assign a single time slot or a contiguous block of time slots.

Since in this paper we need to have a DBA scheme, the two major candidates are VBDC and AVBDC due to the fact that in both cases time-varying resource requests are made by the RCSTs (on the basis of the TCP *cwnd* behavior). In the VBDC assignment, an RCST dynamically requests the total number of slots needed to idle its queue; requests are cumulative. Such category is typically used for Web-browsing traffic. Whereas, in the AVBDC case, an RCST dynamically requests the number of slots, but requests are absolute (not cumulative). Even if VBDC is the default mode in the DVB-RCS standard, for our scenario it seems that an AVBDC-like scheme should be used, since, according to our envisaged cross-layer approach, requests are not exactly related to the status of the buffer and are not cumulative, but rather they refer to the prospected next value of the congestion window *cwnd* for each TCP flow (mapped here to a single RCST).

4 Cross-layer Air Interface Design

Several techniques (such as DBA, adaptive modulation and coding, etc.) need to be adopted in satellite communication systems to improve their efficiency. These techniques permits to follow the dynamics of the system. The conventional OSI protocol stack is based on independent layers, thus precluding the adaptation of each layer according to changing system conditions. The cross-layer approach proposed in this paper is a new paradigm that addresses adaptation considering both system dynamics and the highly varying traffic demand of applications. The classical way to design an interface is to specify separately the physical, link, network and transport layers. Nevertheless, the overall performance can be improved by means of a joint design of several protocol layers. This is the aim of the cross-layer approach, where a protocol stack optimization is achieved by introducing novel interactions even between protocols belonging to non-adjacent layers. Due to the specificity of the optimization process, the cross-layer design needs to be suitably tailored for our DVB-RCS scenario.

Focusing on the resource allocation made at layer 2, we can consider the following 'cross-layer interactions':

- *Physical layer*: radio channel conditions should be continuously estimated. In particular, signal strength and BER estimations should be made available to implement *Adaptive Modulation and Coding* (AMC) and the selection of appropriate formats and priority levels at layer 2.
- *Network layer*: mechanisms for IP-layer QoS provision should be adequately mapped to layer 2 radio resource allocation protocols. Adequate attention should be paid to both IntServ and DiffServ approaches. Different multimedia traffic should be provided either with reserved capacity or capacity on demand.
- *Transport layer*: resource management schemes should be improved to account for the characteristics and behaviors of transport layer protocols, such as TCP and UDP.

– *Application layer*: different traffic types should have specific service level agreements and a monitoring action should be performed jointly with layer 2 in order to modify adaptively the service priority.

In the light of the above, this paper proposes the definition of a layer 2 MAC protocol that on the RCST side exploits information coming from TCP [5],[6] and interacts with the layer 2 on the NCC side for resource allocation decisions. In particular, resources for uplink transmissions are assigned to the RCSTs by the NCC taking into account the requests made by RCSTs that are related to their congestion window values.

4.1 Dynamic resource allocation with interactions between layer 2 and layer 4

Our resource management scheme centrally coordinated by the NCC is described below referring to the network architecture described in Fig. 1, with protocol stacks detailed in Fig. 3. For what follows we refer to the TCP NewReno version [10],[11] where some packet losses can be recovered without a time out expiration by means of the duplicated ACK mechanism.

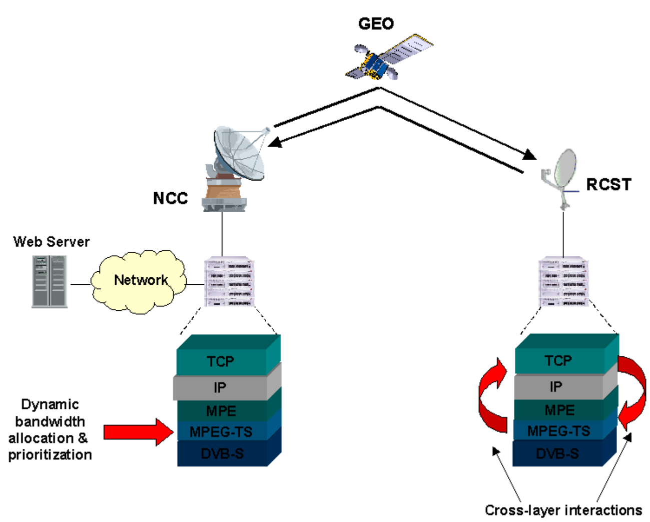

Fig. 3. Protocol stack and cross-layer interactions at both the RCST and NCC.

In the return channel, RCSTs send their resource requests to the NCC on the basis of the expected behavior of the *cwnd* value in the next RTT; this is a special message from layer 4 to layer 2 of the RCST and, then, to layer 2 of the NCC. Then, the NCC looks at the available uplink resources of the the MF-TDMA air interface and sends a broadcast response every super-frame to update the resource allocation to the RCSTs (forward channel) by means of the TBTP message so that every RCST knows the time slots where to transmit. Hence, resource allocation to RCSTs is dynamic on the basis of the behaviors of their congestion window. When a further increase in the need of allocated resources cannot be fulfilled by the NCC (all the MF-TDMA time slots are assigned), the layer 2 of the NCC sends a special message to the layer 4 of RCSTs to stop temporarily further increases in the *cwnd*.

The MAC header in the resource request message sent by the RCST to the NCC contains the two following data that are essential for our resource allocation scheme: (*i*) the TCP phase flag; (*ii*) the estimate of the congestion window value ($cwnd_{next}$) for the next RTT. The NCC filters any incoming segment in order to compare the $cwnd_{next}$ value with the amount of resources already assigned (a_{res}) to the corresponding source (in terms of packets). Two cases are possible:

- If $cwnd_{next} < a_{res}$, the NCC considers that a packet loss has occurred so that the *cwnd* has been reduced; then, for the next RTT $cwnd_{next}$ resources will be allocated to the RCST and $a_{res} - cwnd_{next}$ resources will be made available for other RCSTs.
- If $cwnd_{next} > a_{res}$, the NCC considers that the RCST (i.e., TCP sender) needs a capacity increase in the next RTT. Such resource request is managed as outlined below.

The request is inserted in either a high-priority queue or a low-priority one respectively depending on the slow start phase or the congestion avoidance one, as specified by the TCP phase flag in the MAC header[5]. Requests in the queue are served as follows:

- The NCC allocates further resources (in the next RTT through the broadcast TBTP message) first to requests in the high priority queue and then requests in the low-priority queue are considered if there is still available capacity.
- A minimum resource allocation has to be provided to RCSTs starting to transmit (slow start phase). The minimum capacity allocation is of one slot per MF-TDMA super-frame. When all resources are already assigned, a new RCST entering the service will receive a minimum resource allocation of one slot, by de-allocating it from terminals in congestion avoidance.

[5] We decided to prioritize the allocation of new resources to connections in slow start with respect to those in congestion avoidance. Moreover, in this first implementation of the simulator we have decided that allocated resources to a TCP flow are not de-allocated unless *cwnd* reduces due for instance to a packet loss on the radio channel. Once all the resources are allocated, the RCSTs continue to have assigned the same resources in the next RTTs intervals. A further improvement will be to consider a resource allocation that is able to share dynamically resources among the active TCP flows even in the presence of resource congestion, so that resources can be de-allocated to a given RCST and allocated to another RCST in order to reach a better fairness in the service. The consideration of such aspect is left to a future study.

– When new resources are available (due to the end of a connection, due to a *cwnd* reduction in a connection, etc.) these are primarily assigned to RCST in the slow start phase (high priority queue).
– Within each queue the available resources are assigned to all the pending requests according to the *Maximum Legal Increment* (MLI) algorithm [12] that guarantees a fair sharing of resources.

If the resource request of a given RCST cannot be fully satisfied in the current super-frame, the NCC creates a "waiting list" to assign resources in the next super-frames. If also the amount of the needed resources exceeds the available resources in an RTT, the NCC defines a $cwnd^*$ value lower than the current $cwnd_{next}$ value defined by the RCST; the $cwnd^*$ value, conveyed in a field for TCP options in the segment overhead, is sent back to the RCST up to layer 4 to modify the current *cwnd* value used by the RCST as: $cwnd \leftarrow cwnd^*$. Such procedure is repeated until the congestion persists; hence, during congestion periods, an RCST has a flat *cwnd* behavior forced by the layer 2 of the NCC. Such approach is quite important: since the NCC has a complete control over the resources of the ISN, it can stop (in the presence of congestion) further increases of *cwnd* that would only cause packet losses and throughput reduction at the TCP level.

When the RCSTs finish to transmit or the NCC decides to stop them, the RCSTs leave the network by means of a log-off procedure so that their capacity allocations are released by the NCC.

Note that in typical (non-cross-layer) implementations, the buffer at the MAC level of RCSTs has a value equal to the *Bandwidth-Delay Product* (BDP) of the ISN. Hence, MAC requests a certain number of requests according to the contents of the buffer with DBA. Whereas, in our case with DBA and cross-layer approach, the MAC layer requests resources according to the *cwnd* value estimated for the next RTT; in such case the MAC buffer contains only the packets waiting for the transmission (the packets to which the resources are assigned and that wait for the correspondent slots).

5 Simulation Results

We have implemented a simulator in the ns-2 environment to test the potentialities of our proposed cross-layer approach [13]. We have modified the C++ code and created two new specific classes: "Cross-layer" and "DBA-algorithm". The former (inside the RCST) performs the functions for the exchange of information between TCP and MAC and the functions for the estimation of the requested resources for the next RTT (i.e., $cwnd_{next}$). The latter (inside the NCC) implements the algorithm to allocate/deallocate resources to RCSTs with the evaluation of $cwnd^*$. The simulator is realized through one node for the NCC, one node for the GEO-satellite and one node for each active RCST. For these elements we have defined the appropriate values of longitude and latitude. For each RCST we have a TCP agent attached to the relative node and the FTP application attached to the TCP agent in order to generate the traffic flow. The nodes relative to the RCSTs perform the cross-layer interaction through a C++ cross-layer class. On the NCC side we have a "sink", attached to the node, that sends the ACKs. The NCC node performs the resource allocation through the C++ DBA a lgorithm class.

Our simulation scenario is numerically characterized as follows:

RTT(GEO satellite) = 540 ms; number of frequency carrier = 1; super-frame length = 54 ms (10 super-frames = 1 RTT); TCP segment size = 1444 bytes plus headers of TCP (16 bytes), IP (20 bytes), and MPE (20 bytes), so that the resulting packet length is of 1500 bytes; layer 2 (MPE) buffer length = 50 packets; bit-rate on a carrier = 2 Mbit/s; transport protocol = TCP NewReno; traffic model = ACK clocked for ftp; channel model = AWGN, clear sky conditions, Ku band with *Packet Error Rate* (PER) $\in \{0, 5 \times 10^{-3}\}$; ftp transfer with file size = 5 Mbytes or 7.5 Mbytes (elephant connections). We have adopted a resource granularity of 1 slot (minimum resource allocation) corresponding to 4 packets (1500 bytes) that represent a train of MPEG2-TS packets, transmitted in a burst (slot).

Fig. 4 shows the operation of the resource allocation algorithm that permits to assign resources to a single RCST according to the behavior of its *cwnd*. This graphs has been obtained for a single RCST having to transfer a file of 7.5 Mbytes (i.e., 5000 packets) with a PER = 3×10^{-4}. The initial *ssthresh* value is of 50 packets. The cross-layer technique permits to request to the DBA controller an amount of resources corresponding to the estimated *cwnd* value for the next RTT (such request is fully fulfilled only if there are available transmission resources). *cwnd* reductions are due to packet losses due to the radio channel. In such cases, the *cwnd* value is halved, according to the NewReno scheme and also the DBA scheme is able to change adaptively the resource allocated on an RTT basis. It is important to note that no TCP time outs occur with our algorithm.

The behavior of the mean file transfer time as a function of PER is shown in Fig. 5 for 30 RCSTs connections (starting at exponentially distributed intervals with mean value of 10 s) having to transfer files of 5 Mbytes. This graph shows that the mean file transfer time in the fixed allocation case is much higher than that in the proposed DBA scheme. The behavior of the mean file transfer time is quite flat in the shown range of PER values (for higher PER values, critical behaviors of the mean transfer time would be obtained).

Fig. 6 refers to a case with 3 RCSTs starting TCP-based flows at different instants with PER = 1.5×10^{-4} and files to be transferred of 7.5 Mbytes. When the first RCST that has started to transmit has a loss due to the channel, its *cwnd* is halved and this permits to share the remaining resources among the other RCSTs that have started to transmit (both RCSTs are in the slow start phase). When the second RCST enters the congestion avoidance phase, resources are assigned with priority to the third RCST that is still in the slow start phase and has acquired the higher priority.

Finally, Fig. 7 presents the mean file transfer time as a function of the mean rate for the arrival of RCST TCP-based connections and file transfers of 7.5 Mbytes/s. As expected, we can note a significant reduction in the mean file transfer time of the proposed dynamic allocation scheme due to the most efficient use of radio resources.

6 Conclusions

DVB/RCS is an open standard for interactive broadband satellite services. We considered a group of terminals that, using TCP as transport protocol, send data to the NCC that dynamically manages the assignment of resources. In order to improve the performance at the transport level and the resource utilization, we have proposed a dynamic bandwidth assignment methodology that takes into account TCP evolution. Such approach has been based on an innovative cross-layer interaction between TCP and MAC layers: our scheme aims at synchronizing the requests of resources with the TCP transmission window trend. The obtained resources clearly prove that the cross-layer approach permits to improve the performance at the transport level while achieving a high utilization of radio resources and reducing the occurrence of TCP time outs.

References

1. M. Allman, Glover, D., Sanchez, L.: Enhancing TCP over Satellite Channels using Standard Mechanism, IETF RFC 2488, January 1999.
2. Stevens, W.: TCP Slow Start, Congestion Avoidance, Fast retransmit and Fast recovery Algorithms. Internet RFC 2001, 1997.
   Allman, M., Paxons, V., Stevens, W.: TCP Congestion Control, IETF RFC 2581, April 1999.
3. ETSI EN 301 790. Digital Video Broadcasting (DVB); Interaction Channel for Satellite Distribution Systems. V1.3.1, 2003.
4. ETSI TR 101 790. Digital Video Broadcasting (DVB); Interaction Channel for Satellite Distribution Systems; Guidelines for the use of EN 301 790. V1.2.1, 2003.
5. Shakkottai S., Rappaport, T. S., Karlsson, P. C.: Cross-Layer Design for Wireless Networks. IEEE Commun. Mag., October 2003, pp. 74-80.
6. Guainella, E., Pietrabissa, A.: TCP-Friendly Bandwidth-on-Demand Scheme for Satellite Networks", ASMS conference 2003.
7. Border, J., Kojo, M., Griner, J., Montenegro, G., Shelby, Z.: Performance Enhancing Proxies Intended to Mitigate Link-related Degradations, IETF RFC 3135, June 2001.
8. Floyd S., Fall, K.: Promoting the use of end-to-end congestion control in the Internet, IEEE/ACM Trans. on Netw., vol. 7, pp. 458472, Aug. 1999.
9. ETSI EN 300 421. Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for 11/12 GHz satellite services. V1.1.2, 1997.
10. Allman, M., Paxons, V., Stevens, W.: TCP Congestion Control, IETF RFC 2581, April 1999.
11. Floyd, S., Henderson, T.: The NewReno Modification to the TCP's Fast Recovery Algorithm, IETF RFC 2582, April 1999.
12. Acar, G., Rosenberg, C.: Algorithms to compute for Bandwidth on Demand Requests in a Satellite Access Unit, Proc. of the 5th Ka-band Utilization Conference, 1999, pp. 353-360.
13. NS-2 Network Simulator (Ver. 2) LBL, URL: http://www.mash.cs.berkley.edu/ns/

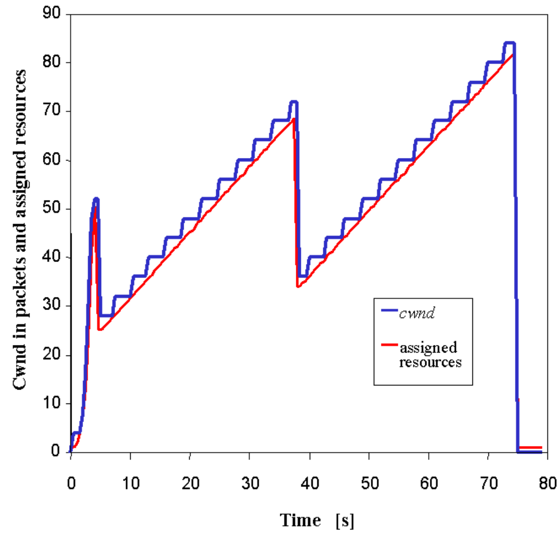

Fig. 4. Behavior of the TCP congestion window as compared to the assigned resources per RTT.

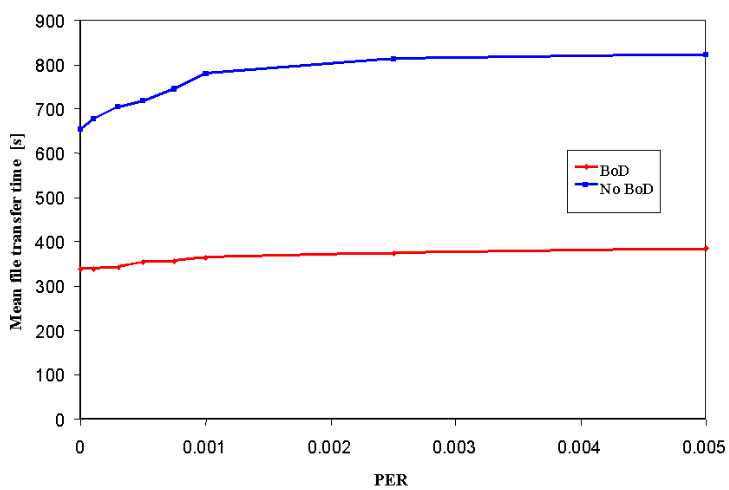

Fig. 5. Mean file transfer delay as a function of PER for both our scheme (BoD) and the classical fixed channel allocation (non-BoD).

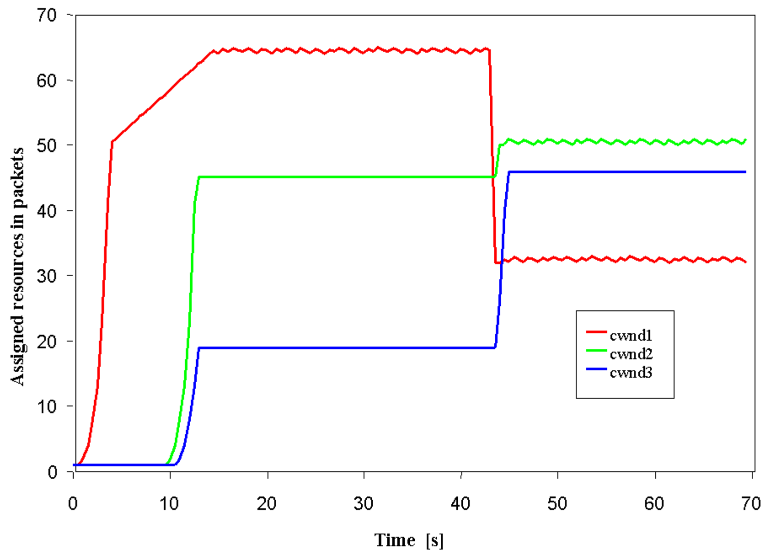

Fig. 6. Behavior of assigned resources in the time in the presence of the start of three concurrent TCP connections.

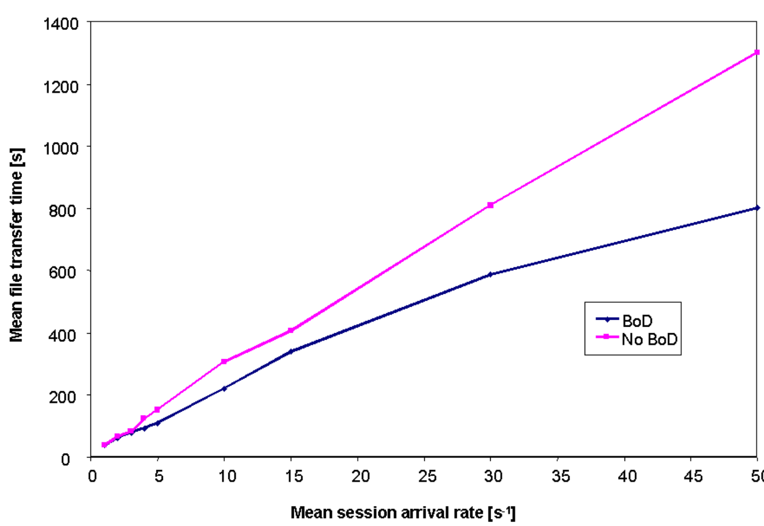

Fig. 7. Mean file transfer delay for both BoD and non-BOD schemes as a function of the arrival rate of TCP connections.