Design Alternatives for a Reliable Multicast Service in a Hybrid HAP-Satellite Network Architecture

Matteo Berioli(1), Giovanni Giambene(2), Ivano Alocci(2), Erina Ferro(3), Paolo Barsocchi(3), Fernando Perez Fontan(4), Mihael Mohorcic(5), Roman Novak(5), Stylianos Karapantazis(6)

(1) German Aerospace Center (DLR) - Institute of Communications and Navigation
D-82234 Oberpfaffenhofen, Germany, email: matteo.berioli@dlr.de

(2) CNIT - University of Siena - Dept. of Information Engineering
Via Roma, 56, 53100 Siena, Italy, email: giambene@unisi.it

(3) Information Science and Technology Institute "Alessandro Faedo" (ISTI)
CNR Pisa Research Area, Via G. Moruzzi, 1, San Cataldo, 56124 Pisa, Italy,
email: erina.ferro@isti.cnr.it, paolo.barsocchi@isti.cnr.it

(4) University of Vigo - Dept. of Signal Theory and Communications
ETSI de Telecomunicacion, Campus Universitario-Lagoas, 36200 Vigo, Spain, email: fpfontan@tsc.uvigo.es

(5) Jozef Stefan Institute - Dept. of Communication Systems
Jamova 39, Ljubljana, SI-1000, Slovenia, email: {miha.mohorcic, roman.novak}@ijs.si

(6) Aristotle University of Thessaloniki - Dept. of Electrical and Computer Engineering
GR-54 124 Thessaloniki, Greece, email: skarap@auth.gr

Abstract

One interesting application for future broadband wireless telecommunication systems is represented by multicast and broadcast services that have unique characteristics with respect to point-to-point transmissions. The focus of this paper is on a multicast protocol for the reliable delivery of huge amount of data to a wide multitude of users on the earth. In particular, we propose a novel architecture characterized by the introduction of an intermediate communication segment between a geostationary satellite and user terminals. Such a segment is based on *High Altitude Platforms* (HAPs), which can have different roles for what concerns transport level functionalities. In particular, we will consider HAPs with or without multicast protocol support. Moreover, in order to assure reliability, we will use transport level coding to protect the transmitted files, and local retransmissions performed by HAPs, thus entailing a shorter delay. Three different scenarios are compared, where HAPs have an increasing role in managing multicast traffic. A simulator has been developed in the ns-2 environment, and the obtained results have proven that the combined effect of data coding applied at the transport layer, together with local retransmissions performed by HAPs, permits to achieve both high efficiency in utilizing air interface resources and low delays in transferring data, in a reliable way.

Special issue topics: capabilities, limitations and potential of HAP systems, networking and integration issues, technical options for maximizing data rates.

Work funded by the European Commission under the framework of the "SatNEx" NoE project www.satnex.org (contract No. 507052) – Multi-HAP research group.

1. Introduction

Satellite communication systems represent an important solution to provide broadcast and multicast high-bit-rate transmissions to a wide population of users on the earth. Contents from a network repository flow through a *GEOstationary* (GEO) satellite and are sent in multicast mode to a multitude of users sub-divided in multicast groups [1]. The main limitation of such an architecture is that the return link towards the satellite may be complex and costly to be realized for small-scale, common user terminals, thus preventing any feedback on the quality of the received information. In spite of this, there is an increasing interest in realizing multicast systems that can guarantee a reliable transfer of data. This is an important requirement in many situations, such as archive transfer (e.g., exchange of data from higher-level to regional-level servers in the network), emergency scenarios, and distributed cache systems. On the other side, *High Altitude Platforms* (HAPs) constitute a real asset to wireless infrastructure operators to provide early rollout of telecommunication services in urban areas and to extend the coverage to rural, remote and impervious areas [2]. Moreover, HAPs are expected to be instrumental in future telecommunications networks, playing a pivotal role in the delivery of *Multimedia Broadcast and Multicast Services* (MBMS) [3]. In this context, the integration of satellite and HAP networks holds considerable appeal since this mixed infrastructure can lead to a powerful integrated network that can make up for the weaknesses of each other. HAPs can be realized by using either small, unmanned aircraft, or dirigibles of suitable sizes. The fundamental characteristics of HAPs are the ability to reach an altitude between 17 and 21 km, and the capability to stay in the zone of operation for long periods.

The idea developed in this paper is to introduce a HAP intermediate segment in a satellite-based network architecture for multicast content delivery. In particular, we envisage a layered architecture with an earth hub station, which multicasts contents through a GEO bent-pipe satellite towards HAPs, which in turn multicast the received data to fixed users covered by their spot-beams. This hybrid HAP-satellite multicast architecture can be used as a standalone content delivery system or it can be integrated into the global IP network. The latter configuration can extend the multicast transmission to end-nodes that are dispersed over even larger geographical areas.

An immediate advantage introduced by the use of HAPs in our GEO-based architecture is the possibility to divide the *GEO-to-user terminal* link into two parts, that is from the satellite to the HAP (Sat-HAP link), and from the HAP to the user terminals (HAP-user link). This solution entails both a shorter link that permits to easily implement the return channel from single-user terminals, and the possibility of local retransmissions (from HAP to ground terminals), thus avoiding the need to propagate the request up to the earth hub station.

Reliable protocols are not new in wide-area networks. However, these protocols apply to terrestrial networks and do not scale well in HAP-satellite architectures. We envisage the possibility to employ a data coding at the transport level for the transmitted files. This is a sort of *reliability scheme* to protect data, integrated with a new developed multicast protocol (named *Reliable Multicast Protocol for HAP*, RMPH), so that some partial losses can be recovered by means of decoding without involving retransmissions. Moreover, HAPs can also decode and correct transport level multicast data received via satellite.

The aim of this paper is to compare, in terms of utilization of the air interface resources, different cases that are distinguished on the basis of the role of the HAP segment with respect to transport protocols. In particular, we used HAPs with or without multicast protocol support. Moreover, in order to assure reliability, we consider: (*i*) Transport level coding to protect the transmitted files; (*ii*) Retransmission managed locally by the HAP; (*iii*) Transport level decoding functions on HAPs.

The remaining Sections of this paper are organized as follows: system architecture, radio spectrum choices, interference issues, radio channel characteristics, and communications standard are discussed in Section 2. Section 3 deals with the multicast protocol implemented in our scenario, the adopted reliability techniques, and the different roles of HAPs in managing multicast traffic; a particular attention is given here to the use of erasure forward error correction codes at the transport level. Section 4 contains simulation results that permit to compare the different options in terms of different performance indexes. Conclusions in Section 5 summarize the work done and the results obtained, while the Appendix will help the reader to find more details on the new developed multicast protocol.

2. System Characteristics

2.1 *System Architecture*

Our system architecture entails a hub earth station, connected to a content delivery network, and a global-coverage GEO bent-pipe satellite, which communicates with a group of HAPs for local-coverage on the earth, as shown in Figure 1. Each HAP is equipped with a regenerating communication payload, which increases platform complexity, as well as weight and power consumption. IP multicast is here envisaged at layer 3. Moreover, different functionalities can be implemented at the transport level[1] on HAPs. Referring to the scenario in Figure 1, a multicast transmission is used from the hub to the HAP through the GEO bent-pipe satellite, and a multicast transmission is used from HAPs to user terminals.

In a scenario where the hybrid HAP-satellite multicast architecture is augmented with IP multicast in global IP network, a network-level gateway, that enables different reliable multicast protocols to communicate seamlessly with each other, should be deployed in the earth hub station. This would permit the exploitation of differences in communication architectures with terrestrial networks.

2.2 *HAP General Issues*

Besides several advantages, HAPs also bring some restrictions, which potentially influence the proposed reliable multicast scenario. These restrictions mainly derive from regulatory aspects, i.e., allocated frequency bands and interference issues that require complex antenna system with precise pointing, acquisition and tracking mechanism, and a good knowledge of radio channel conditions. While detailed system development is beyond the scope of this paper, we still point out in what follows some issues related to frequency bands, interference, and radio channel characteristics.

[1] We consider three different cases in sub-Section 3.3, starting from a simple case where HAPs have no transport layer functionality, to a case where HAPs can buffer files to be retransmitted on the earth and, finally, to the case where HAPs have full transport layer functionality so that they can also decode files, thus sending only complete information to users

Frequency bands

To aid the potential deployment of HAP systems, ITU has allocated 600 MHz at 47/48 GHz (in the *mm*-wave frequency band) worldwide for HAP communications. Additionally, ITU-R examined the sharing of the 28/31 GHz spectrum (in the Ka frequency band) for Japan, and in WRC-2000 approval was obtained due to the fact that the 48/47 GHz band is susceptible to rain attenuation, thus creating serious problems in Asia and tropical regions. Further details on the use of these frequency bands can be found in [4]. In WRC-2003 the use of the 28/31 GHz band has been extended to Region 2. Currently, much pressure is exercised on ITU in order to gain approval for the use of this band in Europe. Nevertheless, the frequency bands allocated for HAP services are shared with some other services. The 47.2-50.2 GHz band is allocated to conventional *Fixed Service* (FS) and *Fixed Satellite Service* (FSS) (Earth-to-Space), while the 28/31 GHz band is shared with fixed and mobile services and FSS (Earth-to-Space). In addition to the aforementioned services, the 31 GHz band is also shared with space science service, so that frequency-sharing studies are of great importance in such a case.

Although being located much closer to ground than satellites, HAPs are essentially restricted to line-of-sight operation between transmit and receive antennas in both allocated frequency bands, as the attenuation of electromagnetic waves, due to rain and vapor, as well as various terrain and human-generated structures, is severe. For the proposed hybrid HAP-satellite network architecture, the operation of HAPs at 28/31 GHz is considered advantageous, potentially guaranteeing less data transmitted with errors than at 47/48 GHz. Line-of-sight conditions are possible in our scenario, since we refer to fixed users on the earth that are supposed to locate properly their antenna systems.

Interference issues

In HAP systems, there exist two kinds of interference [5]: among users of the HAP network itself, and from/to terrestrial and satellite networks that share the same or adjacent frequency bands. Concerning the first case, interference is caused by antennas that serve cells on the same channel and arises from overlapping main lobes or side-lobes. Apparently, the antenna specifications have a profound impact on the *Carrier-to-Interference Ratio* (CIR) [6], and thus, on QoS and the supported number of users. Several factors such as the platform movement, the use of full range of elevation angles (5°-90°), the multibeam antennas and the high density of deployed ground stations complicate the frequency sharing with satellite and terrestrial systems [7],[8]. It has been shown that interference levels are dependent upon antenna radiation patterns, separation distances between HAP and satellite or terrestrial wireless systems, the platform displacement, and the terrain. The on-board antenna design is one of the most crucial aspects of the HAP system. From the user terminal side, the most critical issues are related to *Pointing, Acquisition and Tracking* (PAT), which have to be considered relevant in the 28/31 GHz and 47/48 GHz band. For fixed user scenarios, directive antennas are used that can be aligned with the HAP direction only once during installation.

Apparently, the dominant interference path in our hybrid architecture is that from HAPs to user terminals, which mostly depends on the antenna pattern of the HAP on board-antenna; this interference is taken into account in our simulations through an increased value of *Packet Error Rate* (PER) with respect to the PERs of other links.

Radio channel characteristics

The radio channel characteristics are another issue of utmost importance for HAPs communications, dictated by the choice of the frequency bands. As in the case of satellite links, HAP links undergo free-space propagation by virtue of the unique platform position, and their path loss is much lower owing to the limited distance that the signal traverses. Notwithstanding, rain has a heavy effect on QoS and system availability for frequencies above 20 GHz. The rain effect on signal attenuation and the received power at a fixed station at 30 GHz was studied in [9] for different ground distances. Propagation effects at Ka-band are dominated by gaseous absorption, cloud attenuation, scintillation and rain fades [10]. The margin that is required to overcome the effects of rain attenuation is considerable and could not be supported by platforms with limited power, as in the case of solar-powered unmanned aircraft. Hence, the selection of modulation and coding schemes is crucial, and it may be better to introduce adaptive modulation and coding instead of building upon a fixed power margin, since the worst-case conditions occur only for 0.001% of the time. However, it is beyond the scope of this paper to elaborate upon these issues. In our study, the impact of channel characteristics on each link is reflected in the relevant PER.

2.3 *Communication Standards*

At present, there is no communication standard developed for the use on HAPs for Ka and *mm*-wave frequency bands. While there seem to be several candidate standards that could serve the purpose to deliver real broadband wireless access, in particular IEEE 802 family standards (IEEE 802.11, IEEE 802.16 and IEEE 802.20), DVB standards (DVB-S/S2, DVB-RCS, DVB-T, DVB RCT and DVB-H) and ETSI Broadband Radio Access Network standards (HiperACCESS and HiperMAN), none of them perfectly suits for the use on HAPs [11]. Thus, the choice of the most suitable standard is driven not only by HAP characteristics, but also by targeted services and applications to be provided, operating frequency band, etc.

Let us consider our specific scenario, where the GEO-to-terminal-user link is split in the Sat-HAP link, with prevalent free-space propagation path loss, and in the HAP-user link, with prevalent atmospheric losses. In such a case, the DVB family of standards represents an appealing solution for the Sat-HAP link, particularly satellite DVB standards (i.e., DVB-S, DVB-S2, DVB RCS), due to their inherent suitability for the satellite operating environment. With advanced payload on the HAPs, it is even possible to make the conversion of communication standards (signal formats). In such a case, HAPs have also to provide a transformation between standards on the return link.

The recently-standardized DVB-S2 might also be used in the HAP-to-user link. This technology exploits *Adaptive Coding and Modulation* (ACM) to achieve a good spectral efficiency while maintaining an excellent channel quality by selecting among different MODCOD (*MODulation and CODing*) combinations on the basis of threshold levels on the signal-to-noise ratio. The criteria for choosing the MODCODs in the different channel conditions are not yet defined; anyway, they are surely related to long-term variations of the received signal strength due to cloud attenuation and rain fades. On the other hand, short-term variations are due to scintillation, and they can be considered to have a Gaussian distribution. During these short-term variations, the received signal strength might go below the threshold associated with the operating MODCOD. With high probability, the duration of these short outages produces the loss of a burst of bits. The train lost bits might have different lengths, according to the bit rate and MODCOD; in any case this causes the relevant packet to be discarded. Let us assume that the thresholds and MODCODs are selected so that (after decoding) we can make the conservative assumption that packets (i.e., IP datagrams) are lost according to a random memoryless process characterized by an average PER. Such an assumption is reasonable in clear sky conditions, so that scintillation causes signal fluctuations that are not very significant (generally fractions of a dB). Moreover, we refer to cases with fixed user terminals, which employ highly directional antennas (a preferable assumption for fixed-to-HAP links). With these assumptions, the channel is practically invariant and there is only one direct radio ray coming in the main lobe of the antenna, while reflected rays, if any, are all coming in side-lobes and can be neglected for our purposes.

3. Reliable Multicast

Reliable multicast protocols make sure that all destinations of the multicast transmission correctly receive the transmitted data. This target can be achieved by operating at different protocol levels and in different ways. *Retransmission* techniques allow that lost packets are retransmitted to the receivers, while transport layer *coding* schemes create redundant packets that permit to reconstruct the lost packets. In this Section, we survey some principles and solutions for reliable multicast and propose a new reliable multicast protocol that may be suitable for the application to different hybrid HAP-satellite scenarios, which are described at the end of this Section.

Regarding the retransmission schemes, reliable multicast protocols need to avoid the use of ACK-based mechanisms because they introduce heavy feedback traffic towards the sender, thus increasing the congestion of uplink that, typically, has a reduced capacity with respect to downlink. In particular, here the interest is in using NACK-based approaches. Two existing protocols are taken as references in what follows: the *Pragmatic General Multicast* (PGM) protocol [12] and the *Light-Weight Reliable Multicast Protocol* (LRMP) [13],[14]. Both of them use a NACK-based retransmission scheme to guarantee full reliability. The former protocol needs some modifications to be implemented in the routers in order to support the protocol. NACKs are transmitted back in unicast, thus the path to the source has to be determined. This is done by means of *Source Path Messages* (SPM) broadcast by the source. With this information each node knows to which neighbour a NACK has to be forwarded and to which branches of the multicast tree data have to be re-transmitted; this approach allows bandwidth saving in the other branches. The latter protocol adopts a distributed algorithm to organize the retransmission of lost packets, and it allows local error recovery. Intermediate routers store forwarded packets, so that they can perform local retransmissions when they receive a NACK (the closest router to the destination manages the retransmission). Considering the high GEO latency, local error recovery is very useful in a HAP-satellite scenario, since it allows drastically reducing the time needed to recover data, and it favors the protocol scalability by reducing the number of NACKs sent back to the earth hub station.

In combination with the traditional retransmission scheme a transport layer coding scheme can be added, in order to achieve reliability with a reduced number of retransmissions. This might be particularly useful if resources on the return link need to be saved (smaller number of NACKs or no NACKs are needed at all), or when multiple lost packets are recovered with the retransmission of redundant packets. The original file (to be multicast) is divided into k individual packets, and h redundancy packets are added to it, thus resulting in the transmission of $n = k + h$ packets at the IP level. These packets are finally transferred to the physical layer, which adds independent channel coding to each of them, according to the adopted communication standard (e.g., DVB-S2). This principle is described in Figure 2. At the physical layer the bits affected by low noise levels can be corrected by the physical layer FEC, so that the related packets are passed to the transport layer as "correct" with assumed 100% probability. If the noise level exceeds the correcting capability of the physical layer, the bit failures cannot be recovered. Since IP datagrams are discarded if checksum fails, at the transport layer erroneous packet are not propagated from the IP layer. Hence, at the transport layer we have an erasure channel. The system can use transport layer coding to recover from these erasures. Therefore, the transport layer "sees" either correct packets or erased ones, which is an ideal situation to operate for codes. Using *Maximum Distance Separable* (MDS) codes, like the Reed-Solomon, it is possible to reconstruct the original information if at least k out of n packets of the coded file are received. Therefore, the receiver can cope with erasures, as long as they result in a total loss not exceeding h packets, independently from where erasures occurred. *Low Density Parity Check* (LDPC) codes might be also used for their low complexity [15],[16].

The following sub-Section provides performance evaluations on erasure codes to prove the potentialities for their use at the transport level. Whereas, sub-Section 3.2 describes the basic mechanisms of our proposed RMPH protocol and sub-Section 3.2 shows different scenarios for the implementation of the RMPH scheme in satellite-HAP integrated architecture.

3.1 *Utilization of Erasure Codes at Transport Layer*

As stated above, we can apply erasure codes at the transport layer of the transmitting station, based on Vandermonde matrices (FZC codes) [17]. If applied to data link layer, FZC codes are particularly well suited for mobile users, since they work well on correlated channels. On the other hand, if applied by the sender at the transport layer, their effectiveness is independent of the channel loss type. Such an approach generally could permit a simplification of both the sender and the receiver, since it might render a feedback channel unnecessary; moreover, the time needed to recover the missing packets is reduced. This technique is attractive for multicast applications, since different loss patterns can be recovered from using the same set of transmitted data.

In what follows, we evaluate the performance of erasure codes, when applied at transport level, with respect to not applying any correction at all. The link considered is between the HAP and the user terminals. We selected three mean values of PER (10^{-2}, 4×10^{-3} and 10^{-3}) with independent packet losses, and a packet length of 1500 bytes.

Figures 3 and 4 depict the resulting PER and packet delivery delay for the three values of PER considered, when transport layer erasure codes are applied. Simulations have been done for the packet block length $k = 10$ and for the encoded block length $n = 11$, 12, and 13, and also for $k = 100$ and $n = 101$, 102, 103, 104, and 105. Results show that the packet loss can be reduced from 1% (no application) to 0 by changing the n/k coding ratio. If we use an 11/10 coding ratio (which requires an increase in bandwidth occupancy of 10%), we obtain a PER below 0.1%, and a mean packet delivery delay of 6.6 ms. When a 103/100 coding ratio is applied, we obtain a PER below 0.08%, with only a 3% increase in bandwidth occupancy, and a mean packet delivery delay of about 61 ms. Generally speaking, there is a tradeoff between redundancy, PER, and delay. The delay is an important estimation parameter for real-time streaming applications. For videophone applications, for example, ITU-T Recommendation G.114 [18] gives a delay limit of 150 ms as preferred; in our case, this limit is respected even if we use a block length of 100 packets. However, we can achieve a null PER with an increase in bandwidth occupancy of 30%, if we use a block length of 10 packets, or with an increase in bandwidth of only 6%, if we use a block length of 100 packets. In other words, our simulations show that, with the same bandwidth occupancy, the PER is better if we use a block length of 100 with respect to using a block length of 10. Referring to a PER value of 4×10^{-3}

(such value will be used for numerical evaluations; see Section 4) a coding ratio of 13/10 or 104/100 is sufficient to get a practically null packet loss.

3.2 *Reliable Multicast Protocol for HAP (RMPH)*

We assume to have a source of the contents, co-located with the earth hub station, which sends files of different sizes to the multicast group. Each file is split into a number of packets of the same size, which are passed to the transport protocol. We describe here the basic characteristics of the RMPH multicast protocol that assures reliability in delivering files. The proposed protocol contains some basic ideas of the PGM protocol (see Section 3), but it also integrates transport level coding, similarly to two reliable file transfer protocols: ALC (*Asynchronous Layered Coding* [19]) and FLUTE (*File Delivery over Unidirectional Transport* [20]). Our protocol presents three constituting elements (described below): the Sender, the Receiver (i.e., user terminal) and the Forwarder.

Sender

The sender uses three kinds of packets: DATA packets, for actual data transfer, RDATA packets, for retransmissions, and NCF (NACK confirmation) packets, to confirm a NACK reception. An NCF is broadcast after each NACK reception to inhibit the transmission of additional NACKs (related to the same packet) from different users. After that, the transmission of RDATA is delayed of a random time with the aim of grouping multiple requests and answering with a single retransmission. If a transport layer coding is used, the sender also conveys information in the RMPH header about the used code, the code rate, and the size of the file being transmitted.

Receiver

When multicasting a stream of files, where each file is split into a number of packets, the receiver has to guarantee that each file is correctly and completely received. If the loss of some packets does not allow a correct reception, the receiver asks for retransmissions. Reliability can be granted only for files for which at least one packet was received[2]. Since transmissions of different files may be layered or overlapping, packets belonging to different files can arrive in a mixed order.

[2] The receiver has to be aware of the file transmission; this is possible if at least one packet of the file is correctly received.

For this reason, the receiver has to set a timer to establish when it decides not to wait more for the missing packets in a transmitted file, so that a NACK procedure is started. NACK transmission then is scheduled according to a random delay, to avoid the NACK implosion phenomenon. After the NACK transmission, the receiver waits for an NCF message, and, then, for the retransmitted data, RDATA.

Forwarder

This novel entity introduced by our protocol is thought to be an intermediate router in the multicast tree; in a HAP-satellite scenario, this entity should reside in the HAPs. In traditional multicast protocols, intermediate nodes' task is limited to forward transparently packets. In RMPH, forwarders can also buffer packets, so that they can perform local retransmissions, if required. This reveals to be very useful in a HAP-satellite scenario; forwarders need to identify RMPH packets to be able to retransmit the correct data. This node has functionalities of both the receiver (to identify packets and files and to look for the appropriate packets in the buffer), and the sender (to retransmit data on request). It has to be transparent to the receiver. If a NACK requires a packet that is not in the buffer anymore, the NACK is propagated to higher hierarchical levels (towards the sender). When using transport layer coding, the forwarder node can decide to decode files, and only after that it can decide to forward the packets to the receivers. This has two advantages: packets might be forwarded by applying a different level of protection (e.g., with a different code rate), and it is useless to forward packets if the related file has not been correctly decoded; in such case, the forwarder itself can ask for retransmissions, and only after the correct reception of the whole file it can decide to start forwarding.

The Appendix contains more details on the settings of the different parameters that characterize the RMPH protocol.

3.3 Alternative System Choices for HAP role

In RMPH, reliability can be achieved, as already stated, by means of transport level coding, a retransmission technique, or a combination of both. In this Section, we analyze how these different schemes and configurations can be integrated into a hybrid architecture including the earth hub station, HAPs, and the user terminals (the GEO satellite is transparent and does not directly appears in this picture). We propose three scenarios, which represent three alternative system designs with different HAP design complexity levels. The same scenarios are considered for the simulations, whose results are shown in Section 4.

Reference scenario: HAP supporting up to layer 3 protocols; no retransmissions (classical multicast scheme)

Data received by HAPs from the satellite is simply relayed to the terrestrial users. There are no retransmissions of lost data, but the sending earth hub station utilizes a transport layer coding to improve the reliability of data. This scenario does not foresee a return channel, reliability is partial and achieved only by means of a transport layer coding. This implies a very simple implementation of the end-user terminals, which do not have NACK to transmit back, and of the HAPs, which do not have to manage the transport layer protocol . Note that such reference scenario is well suited for streaming video multicast transmissions, where only partial reliability is requested. The drawback is that resources may be inefficiently used. The first reason is that it is difficult to dimension the amount of transport-layer redundancy to send. If a too robust coding is applied, there might be a waste of bandwidth, whereas, on the other hand, if redundancy is not adequate, entire files could be lost. The second reason for an inefficient use of the bandwidth is that the same level of protection is applied on both Sat-HAP and HAP-user links. The major channel impairments are in the lower part of the atmosphere, so that transmissions on the HAP-user link should be better protected than the other ones, thus allowing an efficient use of redundancy.

Scenario A: HAP with a transport layer buffer and use of retransmissions (basic RMPH scheme)

The source hub earth station sends data to which a transport layer coding is applied. HAPs do not decode this redundancy, but they store data in a transport layer buffer before forwarding them to the receiving user terminals. Differently from the previous case, in this scenario it is possible that users ask for retransmissions of corrupted data. Hence, in the case that users are unable to decode data, they send NACKs. If data are available on the HAP buffer, local retransmissions are performed (i.e., NACKs are not propagated to the sending hub), otherwise the HAP asks for retransmission directly from the earth hub station.

This scenario presents the hardware and software complexity of having a return channel both in the end-user terminal and in the HAPs. The significant advantage of employing retransmissions is that virtually all files can be correctly received by all the users. The novelty here is that, instead of asking for retransmissions from the hub (which entails a GEO satellite round-trip-propagation delay), the HAP has a buffer and thus can retransmit some packets. This approach reduces the average times needed for retransmissions. The drawback is that HAPs need some on-board processing to store all in-transit packets in the buffer. This needs the HAP to implement up to the transport layer of the protocol stack in order to be able to identify packets. As in the previous scenario, the same level of transport layer protection is applied on both parts of the Sat-HAP and HAP-user links.

Scenario B: HAP with transport layer decoding and buffer; use of retransmissions (enhanced RMPH scheme)

In this last and most complex scenario, HAPs decode received data and the transport layer redundancy sent by the hub. If data are correctly decoded, a new coding rate is applied when files are forwarded to the user terminals (a higher protection can be applied on the HAP-user link). If the file cannot be decoded, a file retransmission request (i.e., a NACK) is issued to the hub by the HAP. As in the previous scenario, this is completely transparent to users. The significant advantage of this scenario, with respect to scenario A, is that HAPs can use a different coding protection level to adapt the parameters to the different characteristics of the radio channel toward users. If a file is not correctly received, the users on the earth send back a NACK, and lost packets can be recovered by means of a local procedure that entails retransmissions only from HAPs.

4. Results and Comparisons

A simulator has been built in the ns-2 environment that entails an earth hub station, a GEO satellite, 2 HAP stations, and up to 85 fixed users per HAP. The following characteristics have been considered:

- The traffic source at the hub generates files according to a Poisson process with mean arrival rate of 8.95 file/s; files have a truncated geometrical distribution with mean length of 20938 bytes. The distribution is truncated so that there are at most 64 blocks of 1444 bytes in each file. A transport layer header due to the RMPH protocol of 36 bytes (as in the PGM protocol) and an IP header of 20 bytes is added to each block to form a packet (i.e., an IP datagram) of 1500 bytes.
- The link from the earth hub station to HAPs (through the GEO bent-pipe satellite) has a capacity of 10 Mbit/s and a PER at the IP level equal to 10^{-3}.
- The HAP-user link has a capacity of 10 Mbit/s and a PER equal to 4×10^{-3} (this value depends on the link budget for the connection between users and HAPs; the link budget in turn depends on system design choices and transmission power level. In our simulations, we used a sufficiently high PER value to evaluate the capability of the system to recover data through either retransmissions or coding).
- The link from the users to the HAP (uplink) has a capacity of 5 Mbit/s; PER has been neglected in this case, considering that, for these short packets, a powerful protection can be adopted.
- Packet losses at the IP level are memoryless and uniformly distributed.
- HAPs are at an altitude of 20 km.
- Users are uniformly distributed under the coverage area of each HAP, defined by a contour with 30 degrees minimum elevation angle.

The RMPH protocol parameter values are described in the Appendix. Moreover, the following performance indexes have been measured by simulations for the three system configurations described in sub-Section 3.3:

- β_{sender}: measure of the efficiency of transmissions made by the hub. It is obtained as the ratio between the sum of the lengths of all the (correctly) files transmitted by the hub and the length of all packets sent the first time and all packets resent (due to the NACK scheme) by the hub.

- β_{HAP}: measure of how efficiently the radio transmission resources at HAPs are employed by multicast transmissions. It is obtained as the ratio between the sum of the lengths of all the files (correctly) transmitted by the HAP and the length of all packets sent the first time and all packets resent (due to the NACK scheme) by the HAP.

- $T_{file\_decoding}$: mean time needed to decode a file by a receiver. It is measured from the arrival of the first packet of the file to the arrival of the last packet that permits to recover correctly the file. Note that the lower bound to $T_{file\_decoding}$ is obtained considering that a file is received without needing retransmissions; the corresponding value is 17.4 ms (mean file transmission time at 10 Mbit/s).

- ***Reliability***: percentage of correctly received files at the user terminals.

Figure 5 shows β_{sender} and *Reliability* behaviors as functions of the code rate employed by the hub (cr_s) at the transport level. Since in the reference scenario (dash-dot lines) there are no retransmissions, β_{sender} obviously increases as the code rate increases (this curve simply represents the code rate used by the hub). Of course, this is not an advantage of such a configuration with respect to the other ones, since we note from the reliability graph (still in Figure 5) that there is a significant increase in lost files as the code rate increases (i.e., as the coding protection decreases). Scenario A (continuous line) allows a slightly worse efficiency, but it has the merit to permit the reliable transfer of all the files. Scenario B (simulation results highlighted by square markers) obtains a very good efficiency and reliability. As expected, we may conclude that scenarios A and B are the most interesting configurations for the reliability standpoint, and we will investigate further on them in the following graphs.

Figure 6 presents the β_{HAP} behavior as a function of the code rate employed at the HAP (cr_h) for both A and B scenarios. Note that the code rate at the HAP level is the same of that at the hub level for scenario A. As the code rate reduces (i.e., cr_h tends to 1), the file transmission is less protected and more packet retransmissions are required. The behavior of β_{HAP} is affected by opposite effects: the reduction in redundancy tends to increase efficiency, but more retransmissions are requested which in turn reduce efficiency. The resulting combined effect is that β_{HAP} curves exhibit a maximum that is an optimum condition for the transport layer coding configuration. This occurs in both A and B scenarios. However, scenario B allows a better efficiency than scenario A, because scenario A employs a decoding procedure at the HAP that permits to reduce the number of retransmissions requested by users.

Figure 7 shows $T_{file\_decoding}$ as a function of the code rate employed at the HAP (cr_h) for both A and B scenarios. As in the previous graph, the code rate at the HAP level is the same of that at the hub level in scenario A. Note that $T_{file\_decoding}$ is measured at the user terminals. In Figure 7, $T_{file\_decoding}$ increases as the code rate reduces for scenario A, since more retransmissions are needed. On the other hand, we have a slower increase in scenario B. This is due to the fact that, in this case, local recovery procedures of corrupted files can be employed, thus entailing lower delays. When the code rate is too low, there is not sufficient protection and the time to receive a file suddenly increases due to the frequent need of retransmissions. There is a kind of threshold effect and the code rate threshold is around 0.85-0.9, which corresponds to the maximum in Figure 6. This point reveals the a change of a trend, where the increase in retransmissions becomes prevailing with respect to the reduction of redundancy.

Figure 8 deals with $T_{file\_decoding}$ as a function of the number of users per HAP for both A (cr_h = cr_s = 0.85) and B (cr_h = 0.85 and cr_s = 0.9) scenarios. We may note that in scenario A, $T_{file\_decoding}$ has a (linear) increase as a function of the number of users, since there is an increase in the number of issued NACKs. The higher delays of scenario A with respect to the B one are due to the fact that, in scenario A, corrupted packets can entail to request retransmissions to the hub, thus increasing the delay.

In concluding this Section, it is important to remark that scenario B can guarantee an efficient use of resources and low file decoding times, when an appropriate tuning of the coding protection levels on the different links is applied. Hence, our proposed architecture combined with the RMPH protocol represents a promising solution to achieve reliable multicast transmissions.

6. Conclusions

Multicast traffic is an application for providing the same stream of data to a multitude of users. In such a context, GEO satellites can play an interesting role since they allow the coverage of broad areas on the earth. We have proven that multicast transmissions via satellite can be made even more efficient if the system architecture is combined with a HAP communication segment and if our proposed RMPH scheme is employed. In particular, if HAPs are enabled with an increased level of complexity (from layer 3 up to layer 4) it is possible to obtain an efficient utilization of resources and scalability for the number of supported users. We have also shown the importance of including a transport layer coding to protect multicast traffic in order to limit as much as possible the load of packet retransmissions.

Appendix

In the RMPH protocol, distinct parameters have been used at the user terminal, the HAPs, and the earth hub station. This Appendix aims at describing these parameters and explaining the assigned numeric values. Let RTT_h denote the maximum *Round Trip Time* between users and HAP; let RTT_s denote the maximum RTT value between users and the earth hub station (in such a case the link is through a GEO satellite). In particular, we have considered RTT_s = 500 ms and RTT_h = 0.27 ms. In our RMPH protocol implementation, NACKs are collective, in the sense that there is a single NACK for a whole file, describing all the packets that need to be retransmitted for that file.

Parameters employed by the multicast protocol at the user terminal:

- *nak_bo_ivl*: amplitude of the time window within which (uniform distribution) the user sends the NACK as soon as it realizes that a packet is lost. Note that *nak_bo_ivl* must be selected much greater than the RTT_h between the user and HAP**,** to avoid excessive transmissions of NACKs related to the same file (we hope that before this waiting time expires for many users, the NCF is received, thus stopping the issue of further NACKs).

- *nak_rpt_ivl*: maximum waiting time to receive the NCF message from the HAP (or from the satellite) that is related to a sent NACK. If *nak_rpt_ivl* expires for a user without receiving the

NCF, the user has to re-send the NACK. Such a waiting time is selected as a function of the maximum RTT value in the network (see below).

- *nak_rdata_ivl*: maximum waiting time to receive the RDATA packet related to the sent NACK. If *nak_rdata_ivl* expires for a user, a new NACK is sent. The value of this parameter is selected as a function of the *rdata_delay* parameter of the sender (see below).
- *wait_data*: timer stared at the reception of each packet of a file; if no new packet is received within this timeout, the user starts the procedure to send a NACK within the *nak_bo_ivl* interval. Note that considering the transport level coding, it may occur that $k-1$ packets of a file are correctly received and that the user waits for the reception of the last correct packet that permits to decode the entire file. This timer is useful to limit the waiting time to decode a file; this is particularly important when the number of received packets in a file is insufficient to decode it. The setting of this parameter is detailed in what follows.

Parameters used by the multicast protocol at the HAPs:

- *buffer_size*: size of the buffer employed on the HAP to store packets. This size is defined in the simulations so that we can neglect the loss of packets due to buffer congestion.
- *rdata_hap_delay*: waiting time before the HAP sends the RDATA packet after having received the first NACK of the related file. In scenario B, this time is always employed for retransmissions to users; in scenario A it is used only if the HAP has a correct copy of the packet. In the retransmission of data this delay is needed to collect at the HAP all the NACKs relevant to the same packet in order to make a single retransmission. The setting of this parameter is detailed in what follows.
- *hap_nak_bo_ivl*, *hap_nak_rpt_ivl*, *hap_nak_rdata_ivl*, and *wait_data* are used only in scenario B, and are defined as the corresponding ones for the user terminal.

Parameter used by the multicast protocol at the earth hub station:

- *rdata_delay*: waiting time before sending the RDATA packet after receiving the first related NACK of a file. This parameter is analogous to *rdata_hap_dela,y* since the multicast protocol is applied at two layers: the multicast transmissions from the hub to the HAPs, and the multicast transmissions from the HAP to the users. In the retransmission of data, this delay is needed to collect at the hub all the NACKs relevant to the same packet. The numerical setting of this parameter is detailed below.

The *nak_rdata_ivl* parameter of the receiver must be set to a value greater than or equal to the *rdata_delay* of the sender, otherwise the receiver can be forced to re-send a NACK since the RDATA packet is correctly received too late.

From Figure A.1 is it easy to derive the value to be used for parameter *nak_rpt_ivl* at the user terminal. From the NACK transmission instant to the receipt of the NCF message at the user, there is a delay equal to RTT_s for scenario A and equal to RTT_h for scenario B. For the sake of simplicity and according to the previous *nak_rpt_ivl* definition, we have always considered *nak_rpt_ivl* = RTT_s; such assumption is conservative in scenario B. We still employ Figure A.1 to explain the *rdata_hap_delay* value that is used when HAPs can perform retransmissions (i.e., scenarios A or B). Let us assume that user #1 is the first one that sends the NACK for a file. Further NACKs of the same file can be sent only by those users that schedule the NACK transmissions in the time interval between the NACK sent by user #1 and the NCF reception[3]. Hence, we need *rdata_hap_delay* ≥ RTT_h. Similar considerations can be adopted to define the *rdata_delay* value used by the earth hub station: *rdata_delay* ≥ RTT_s. However, the further considerations below are needed to refine the selection of both *rdata_hap_delay* and *rdata_delay* values.

It is important to note that not all users (that need to ask for retransmissions related to a given file) become aware of the need to send a NACK at the same time, as assumed in Figure A.1 for the sake of simplicity. This is due to the use of both the transport layer coding and the cumulative NACK scheme (i.e., NACK on a file basis). Hence, the selection of both *rdata_hap_delay* and *rdata_delay* values has to consider these aspects. Let us refer a generic code (k, n); a user (also a HAP, in scenario B) is able to decode a file as soon as it receives the first k correct packets of this file. To avoid that a user waits for a packet (or packets) an infinite time to conclude successfully the file decoding (this could be the case when an excessive number of packets is lost in the file), we use a timer, named *wait_data* that is reset at the correct reception of new packets of the same file. If this

[3] In Scenario A, the HAP sends the NCF message if it has the correct version of the packet, otherwise the retransmission request is propagated to the earth hub station that is in charge to send the NCF. Whereas, in scenario B, the HAP sends NCFs to users, but also the earth hub station sends NCF to the HAPs, since the multicast traffic is at two levels.

timer expires, a NACK procedure is started for the retransmission of the missing packets in the file. Let us refer to a file with L packets (in this study L is taken equal to the maximum file length of 64 packets). Let τ denote the time to transmit a packet; since, all links from the hub to the users have a capacity of 10 Mbit/s, a packet of 1500 bytes requires a time τ = 1.2 ms to be transmitted. Moreover, the entire file of L packets is sent in a time equal to $L\tau$. The maximum delay (worst-case) between the transmissions of two NACKs (related to the same file) by two users is characterized as follows:

- The i-th user receives only the first packet of a file, while the j-th user receives only the last packet (i.e., the L-th one) of the same file;
- Moreover, let us assume that the i-th user sends the NACK with the minimum delay (i.e., *wait_data*), and that the j-th user sends the NACK with the maximum delay (i.e., *wait_data*+*nak_bo_ivl*).

Hence, the time interval between the two NACKs is equal to $(L-1)\tau + nak\_bo\_ivl$. Hence, we can conclude that the following dimensioning formulas can be applied for *rdata_hap_delay* and *rdata_delay*:

$$rdata\_hap\_delay = \min\{\mathrm{RTT_h}, (L-1)\tau + nak\_bo\_ivl\} \quad \mathbf{(1)}$$

and

$$rdata\_delay = \min\{\mathrm{RTT_s}, (L-1)\tau + nak\_bo\_ivl\} \quad \mathbf{(2)}$$

Note that the above formulas do not depend on the *wait_data* value. Moreover, since, *nak_bo_ivl* must be set much greater than $\mathrm{RTT_h}$, condition (1) simply becomes: *rdata_hap_delay* = $\mathrm{RTT_h}$. In conclusion, Table A.1 below describes the numeric values employed for the different parameters.

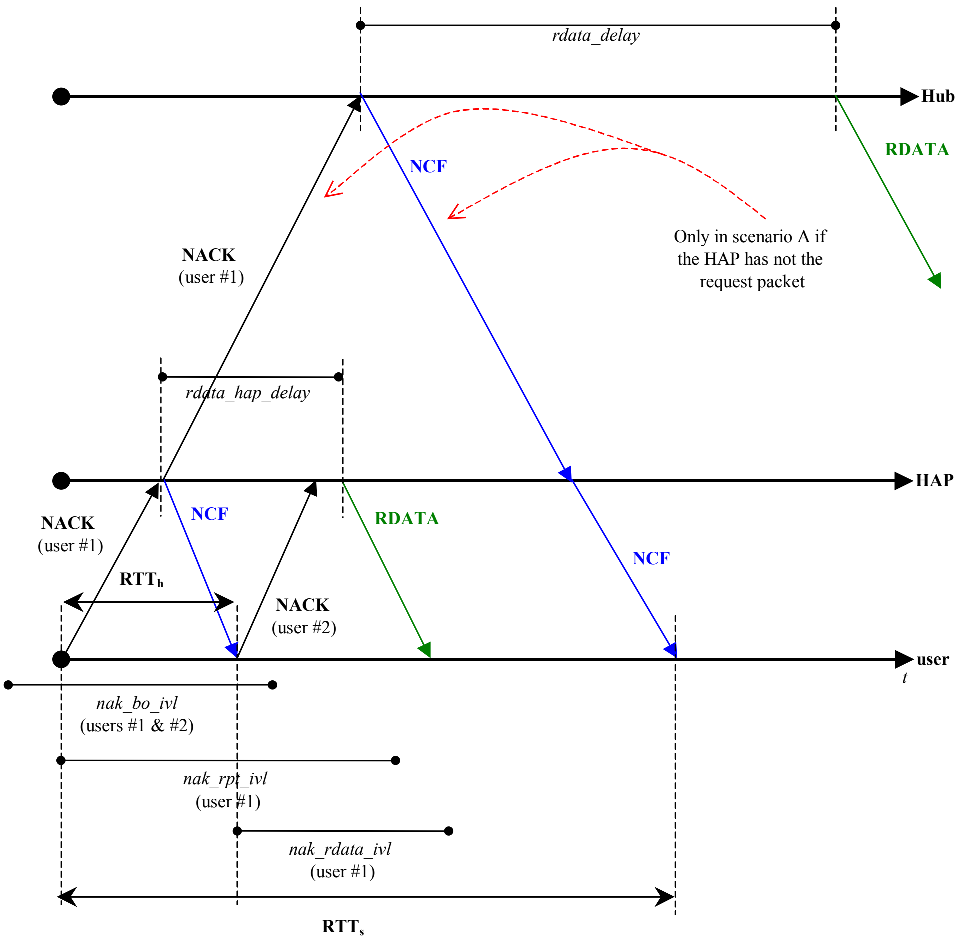

Figure A.1: Time diagram related to the transmission of NACKs and NCF messages.

Table A.1. Parameter values of the multicast protocol, RMPH (parameters with asterisk are only for scenario B).

Hub	*rdata_delay*	86 ms
	RMPH_packet_size	1444 byte (payload)
User terminal	*nak_bo_ivl*	10 ms
	nak_rpt_ivl	500 ms
	nak_rdata_ivl	86 ms
	wait_data	11 ms
HAP	*buffer_size*	128 pkt
	rdata_hap_delay	0.27 ms
	*hap_nak_bo_ivl**	10 ms
	*hap_nak_rpt_ivl**	500 ms
	*hap_nak_rdata_ivl**	86 ms
	*wait_data**	11 ms

Bibliography

[1] M. Karaliopoulos, K. Narenthiran, B. Evans, P. Henrio, M. Mazella, W. De Win, M. Dieudonné, P. Philippopoulos, D. I Axiotis, I. Andrikopoulos, I. Mertzanis, G. E. Corazza, A. Vanelli-Coralli, N. Dimitriou, A. Polydoros, "Satellite Radio Interface and Radio Resource Management Strategy for the Delivery of Multicast/Broadcast Services via an Integrated Satellite-Terrestrial System", *IEEE Communications Magazine*, pp. 108-116, September 2004.

[2] D. J. Bem, T. W. Wieckowski, R. J. Zielinski, "Broadband Satellite Systems", *IEEE Communications Surveys and Tutorials*, 1st quarter 2000.

[3] 3GPP TS 23.246, "Multimedia broadcast/multicast service; architecture and functional description", Release 6.

[4] D. Grace, J. Thornton, G. P. Whire, C. Spillard, D. A. J. Pearce, M. Mohorcic, T. Javornik, E. Falleti, J. A. Delgado-Penin, E. Bertran, "The European HeliNet Broadband Communications Application - An Update on Progress", *The 4th Stratospheric Platform Systems Workshop 2003* (SPSW 2003), 26-27 February 2003, Shinagawa, Tokyo, Japan.

[5] S. Karapantazis, F.-N. Pavlidou, "Broadband Communications via High Altitude Platforms: A Survey", *IEEE Communications Surveys and Tutorials*, first Quarter 2005 issue, [Online], Available: http://www.comsoc.org/livepubs/surveys

[6] J. Thornton, D. Grace, M. H. Capstick, T. C. Tozer, "Optimizing an Array of Antennas for Cellular Coverage From High Altitude Platform", *IEEE Trans. on Wireless Communications*, vol. 2, no. 3, May 2003, pp. 484-492.

[7] M. Oodo, R. Miura, T. Hori, T. Morisaki, K. Kashiki and M. Suzuki, "Sharing and Compatibility Study between Fixed Service Using High Altitude Platform Stations (HAPs) and Other Services in the 31/28 GHz Bands", *Wireless Personal Communications*, Kluwer Academic Publishers, vol. 23, no. 1, Oct. 2002, pp. 3-14.

[8] V. F. Milas, P. Constantinou, "Interfrence Environment between High Altitude Platform Networks (HAPN), Geostationary (GEO) Satelite and Wireless Terrestrial Systems", to appear in *Wireless Personal Communications, Kluwer Academic Publishers*, special issue on "High Altitude Platform (HAP) Systems: Technologies and Applications".

[9] D. Grace, N. E. Daly, T. C. Tozer, A. G. Burr, D. A. J. Pearce, "Providing multimedia communications from high altitude platforms", *International Journal of Satellite Communications*, Wiley InterScience, vol. 19, no. 6, Nov./Dec. 2001, pp. 559-580.

[10] U.-C. Fiebig, "A Time-Series Generator Modelling Rain Fading", *Proc. Open Symposium on Propagation and Remote Sensing*, URSI Commission F, Garmisch-Partenkirchen, 2002.

[11] D. Grace, M. Mohorcic, J. Horwath, M. Capstick, M. Bobbio Pallavicini, M. Fitch, "Communications from Aerial Platform Networks delivering Broadband for All - An Overview of the CAPANINA Project", *Korean Workshop on HAPs*, Korea, November 2004.

[12] T. Speakman, J. Crowcroft, J. Gemmell, D. Farinacci, S. Lin, D. Leshchiner, M. Luby, T. Montgomery, L. Rizzo, A. Tweedly, N. Bhaskar, R. Edmonstone, R. Sumanasekera, L. Vicisano, "PGM Reliable Transport Protocol Specification", *IETF RFC 3208*, December 2001.

[13] S. Floyd, V. Jacobson, S. McCanne, C. Liu, L. Zhang, "A Reliable Multicast Framework for Lightweight Sessions and Application Level Framing", *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 784-803, December 1997.

[14] Tie Liao, "Lightweight Reliable Multicast Protocol Specification", *Internet Draft* draft-liao-lrmp-00.txt, October 13, 1998, [Online], Available: http://webcanal.inria.fr/lrmp.

[15] R. G. Gallager, "Low density parity check codes," *IEEE Transactions on Information Theory*, vol. 8, pp. 21–28, January 1962.

[16] D. Mackay and R. Neal, “Good codes based on very sparse matrices,” *in Proceedings of 5th IAM Conference: Cryptography and Coding*, LNCS No. 1025, 1995.

[17] L. Rizzo, “Effective erasure codes for reliable computer communication protocols”, *ACM Computer Communication Review*, April 1997.

[18] “International telephone connections and circuits General Recommendations on the transmission quality for an entire international telephone connection”, ITU-T Recommendation G.114.

[19] M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, and J. Crowcroft, “Asynchronous Layered Coding (ALC) Protocol Instantiation”, *IETF RFC 3450*, 2002.

[20] T. Paila, M. Luby, R. Lehtonen, V. Roca, and R. Walsh, “FLUTE - File Delivery over Unidirectional Transport.” draft-ietf-rmt-flute-08.txt (IETF Work in Progress), 2004.

Figures

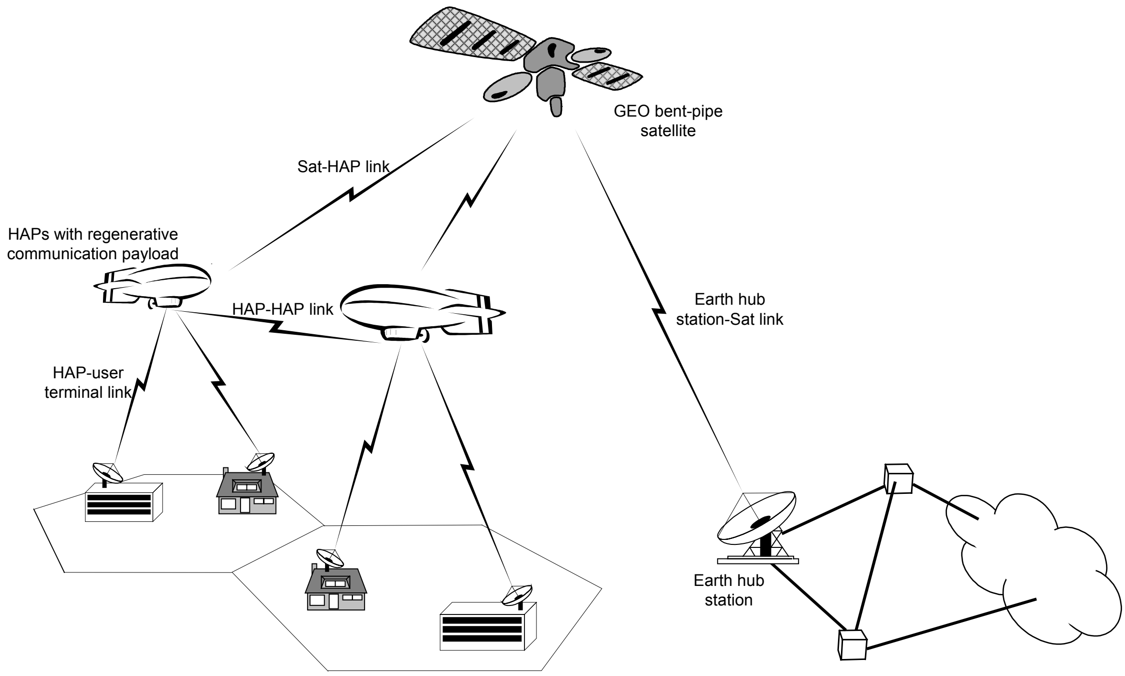

Figure 1: Envisaged multicast scenario.

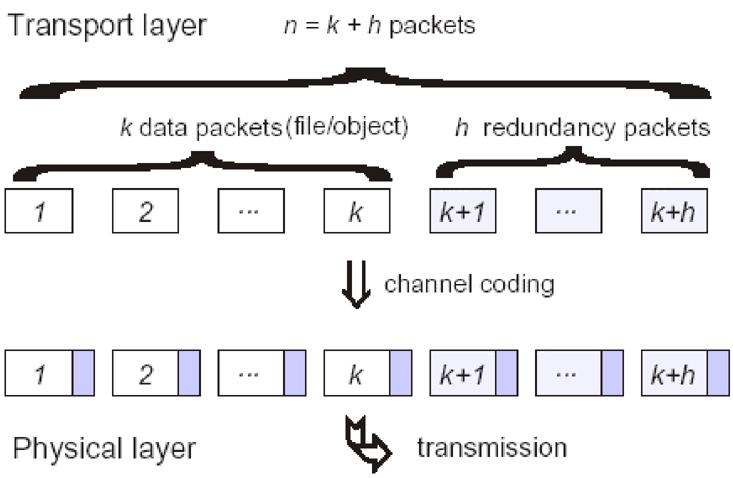

Figure 2: Transport layer coding.

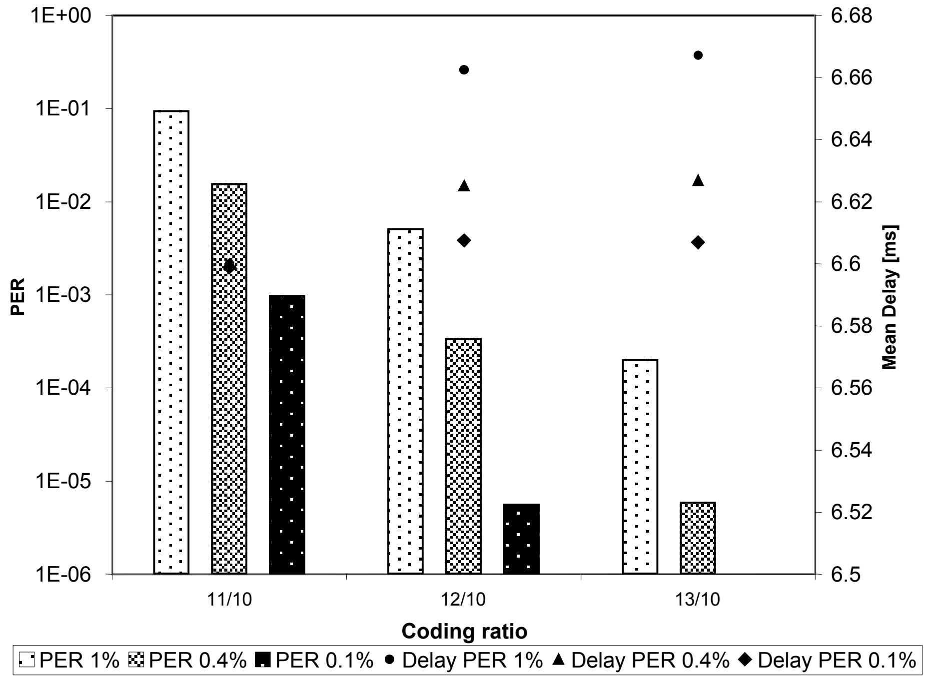

Figure 3: PER and mean delay after coding versus coding ratios, for a block length of 10 packets and different raw PER values.

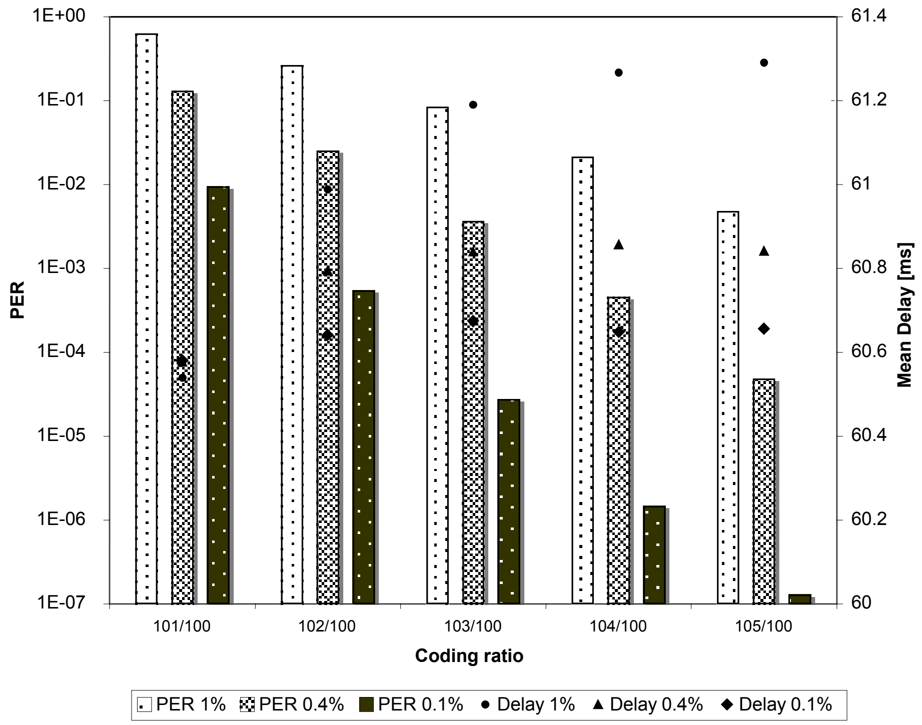

Figure 4: PER and mean delay after coding versus coding ratios, for a block length of 100 packets and different raw PER values.

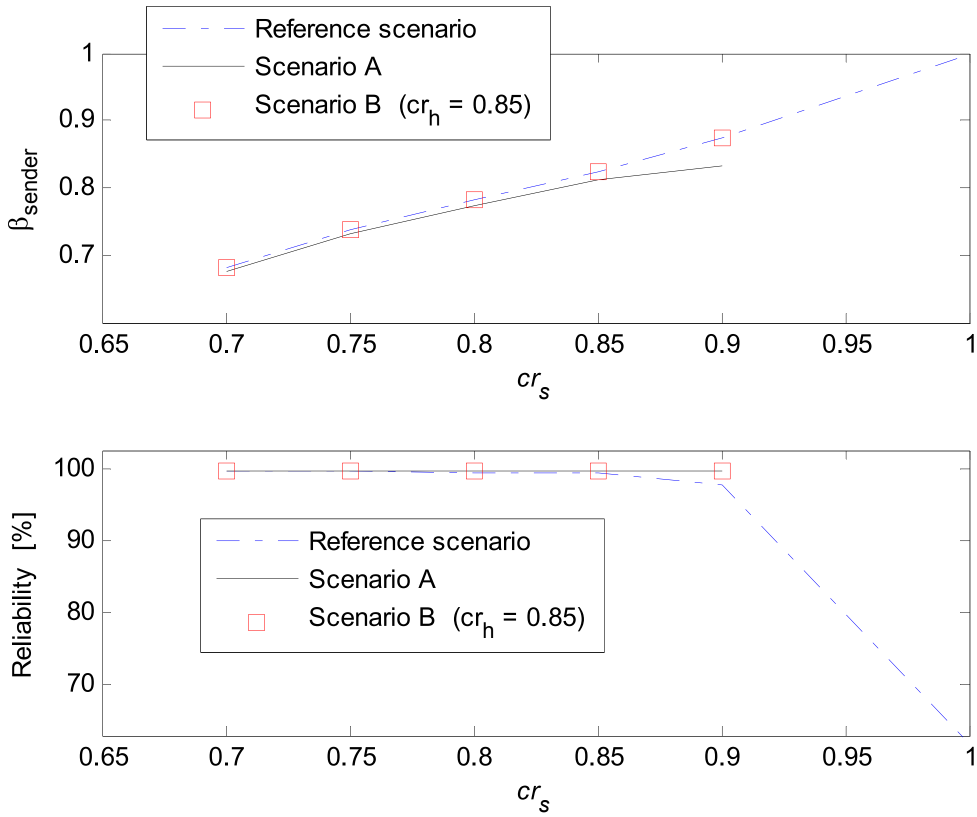

Figure 5: Top graph: β_{sender} behavior as a function of the code rate employed at the sender (cr_s). Bottom graph: Reliability (%) behavior as a function of the code rate employed at the sender (cr_s). 85 users per HAP.

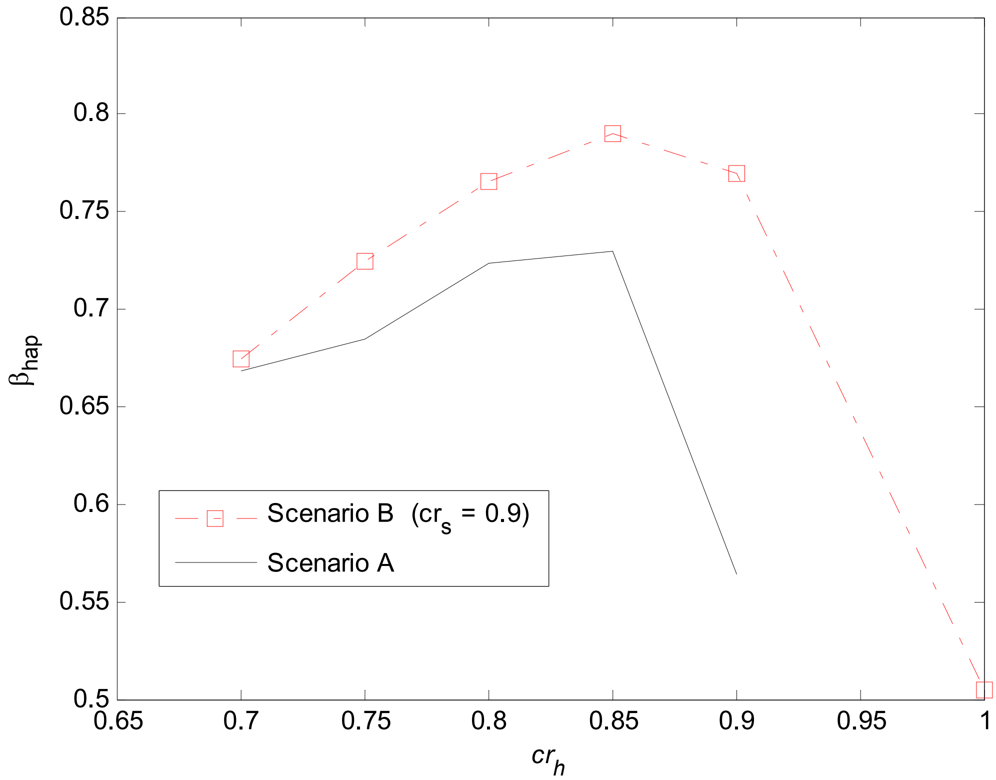

Figure 6: β_{HAP} behavior as a function of the code rate employed at the HAP (cr_h) for both scenarios A and B for 85 users per HAP.

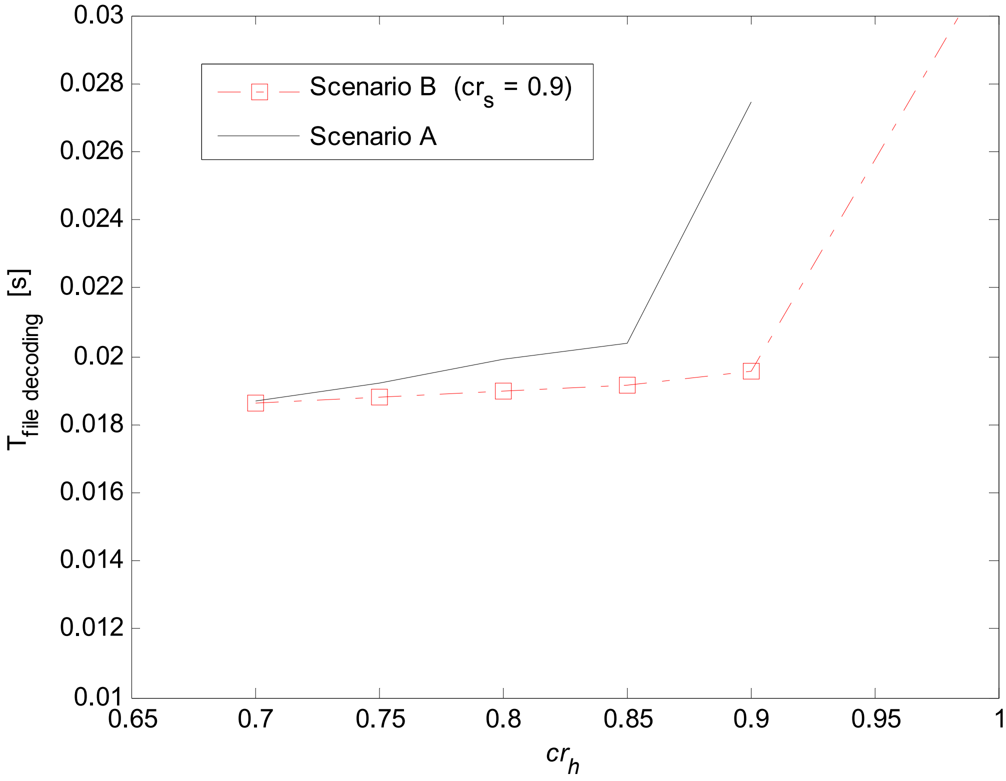

Figure 7: $T_{file\_decoding}$ as a function of the code rate employed at the HAP (cr_h) for both scenarios A and B with 85 users per HAP.

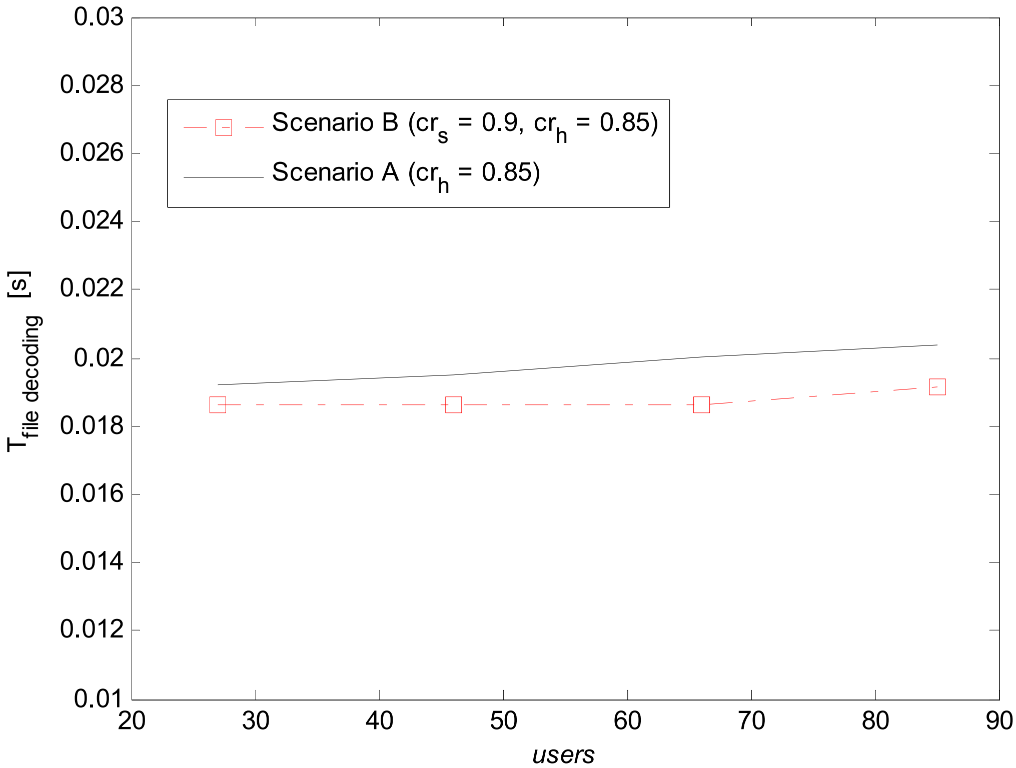

Figure 8: $T_{file\_decoding}$ as a function of the number of users per HAP for both scenarios A ($cr_h = cr_s =$ 0.85) and B ($cr_h = 0.85$ and $cr_s = 0.9$).