

Research Article

Video Streaming Transfer in a Smart Satellite Mobile Environment

Nedo Celandroni,¹ Franco Davoli,² Erina Ferro,¹ and Alberto Gotta¹

¹ CNR-ISTI, Area della Ricerca del C.N.R., Via Moruzzi 1, 56124 Pisa, Italy

² CNIT-University of Genoa Research Unit and Department of Communications, Computer and Systems Science (DIST), University of Genoa, Via Opera Pia 13, 16145 Genova, Italy

Correspondence should be addressed to Franco Davoli, franco@dist.unige.it

Received 27 February 2009; Accepted 10 July 2009

Recommended by Sandro Scalise

Transportation media are becoming “smart spaces”, where sophisticated services are offered to the passengers. We concentrate on video streaming provided on buses that move in urban, suburban, or highway environments. A content provider utilizes a DVB-S2 satellite link for transmitting video streams to a bus, where they are relayed to passengers’ devices. We say that a bus works in smart mode if it takes advantage of the knowledge of the exact points where fixed obstacles will prevent receiving the satellite signal for a certain time period. This information is sent to the hub via a return channel. The hub, in its turn, suspends the transmissions to that specific bus for the given time interval, thus avoiding information losses and unnecessary bandwidth occupation. Buffering video packets, without any quality of service (QoS) degradation, seamlessly compensates channel blockages up to a given duration. We determine the most appropriate transmission parameters for video streaming with good video QoS in a mobile satellite environment; moreover, we evaluate how “smart” the system can be in terms of bandwidth saving, by comparing it with the situation where the bus does not exploit the description of its route, still maintaining the same QoS requirements.

Copyright © 2009 Nedo Celandroni et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

We consider a new utilization of buses, transforming them from pure travel media for transporting passengers to a mobile communication system, able both to contribute services for the public administrations (environmental and traffic monitoring, urban road surveillance, sensing data collection while passing, etc.) and to offer services to passengers, such as news, advertisements, trains/flights timetables, city and tourist info, bus/train routes and timetables, Internet access, music, instant messaging, video, TV on demand, and so on. In this paper, we concentrate on this second aspect, that is, the multimedia streaming services for passengers who utilize long-haul buses.

The system assumptions are the following (Figure 1). We consider a geostationary satellite system with a terrestrial hub station acting as a network control center (NCC). The hub transmits a large amount of multimedia data (video streams and the other information flows, previously mentioned) in time division multiplexing (TDM) mode to a fleet of vehicles,

using the DVB-S2 standard [1]. The return link (from the vehicles to the hub) is assigned to a mobile network link, (from 2.5 G up to 3.5 G service provider), which guarantees a widespread coverage and sufficient bandwidth for signaling toward the hub. We assume that the hub is able to change the average speed of each streaming connection by changing the transmission frequency of the relevant packets. In Section 3, we will address two different rates for each video streaming transfer, r_n (normal rate) and r_h (higher rate), respectively.

The transmission of the video streams to a fleet of buses is done in a “personalized” mode: a multimedia flow is addressed to a specific bus of the fleet either in response to an explicit request, or simply to disseminate information. The Relay Media Proxy allocates a separate portion of memory (i.e., buffers) for each flow, in order both to serve the flows and to manage the store and forward mechanisms that will be detailed in the following. In this paper, we consider two types of video streaming transmissions, which are real-time Video on Demand (VoD), based on the Real-time Streaming Protocol (RTSP) [2], and Live Video Streaming

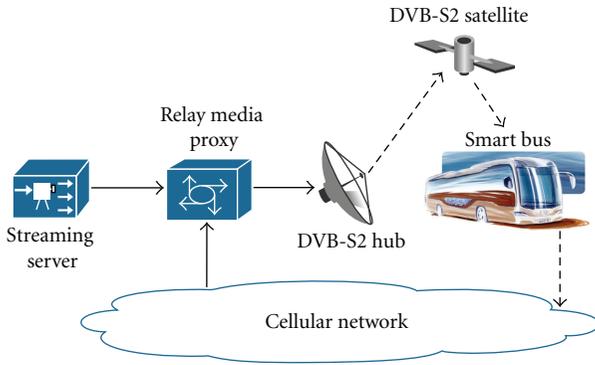


FIGURE 1: The video streaming transmission in the “smart” bus environment.

(LVS), such as IPTV [3]. Each bus relays the received video to its passengers as an on-board service.

We also assume that each bus is able to know and exploit some pieces of information relevant to its route. Bus routes can traverse different environmental settings, each one characterized by its typical obstacles: (i) urban, with tall buildings; (ii) suburban (small villages and towns, as well as the suburbs of big cities); (iii) rural (open areas with tree alleys and forests); (iv) highway (open areas with the presence of relatively frequent bridges and tunnels). In any of the mentioned environments, the buses’ paths can be well traced by any type of GPS device available on the market; each environment can be characterized by a specific channel model, which takes into account multipath fading and shadowing phenomena in a static seasonal environment [4]. The communication from the hub toward buses is therefore adapted to the channel type, by using the most appropriate transmission parameters, as detailed in the following sections. The authors are well conscious that for some trips the usage of the train is surely more convenient than the bus; nevertheless, the connection between distant cities by means of extraurban buses is a reality both in Europe and in the USA, and this is the case we address in our paper. In this scenario, the only possible means for connecting a bus with the hub is the satellite. As far as the authors know, this research topic has not yet been addressed in the literature.

The behavior of the mobile satellite channel, as described by the Markov models in [5], is such that relatively long periods of signal blockage are possible. Such blockage events can be counteracted—in addition to the DVB-S2 adaptive coding and modulation (ACM) applied at the physical layer—by adopting Upper-Layer Forward Error Correction (UL-FEC) and interleaving at the packet level. A possible drawback of these techniques is the introduction of a sensible delay in the playback of the stream. In any case, even in the presence of such strong countermeasure actions, the occurrence of particularly long blockage periods may lead to a complete channel outage, since the loss of all packets transmitted during a long period may overcome the correcting capacity of the upper-layer code. When the routes are well known, as when traced by means of GPS, these periods can be predictable with good precision. Therefore,

in unicast transmission mode, our “smart bus” can alert the hub in advance, thus making it stop sending packets, which would otherwise be lost, for the duration of the blocking time interval. We denote this operational condition as *Smart Mode* (SM); we say that a bus works in *Normal Mode* (NM) if it does not take advantage of the route information. However, UL-FEC and packet interleaving are applied in both modes.

Thus, our smart bus interacts with both the GPS and the hub. The bus may reach the latter either via a cellular network or via a satellite return channel. When a bus, by means of the GPS, detects the presence of a bulky obstacle along the route (tunnel, long/tall buildings, long foliage lines, etc.), it advises the hub, via the return channel, that it would lose packets, starting from an estimated time instant, which accounts for uncertainties in the position estimation, after a round-trip delay time; such an estimation depends on the precision of both the current vehicle’s position and speed. Thus, the hub can temporarily stop sending packets to this bus. At the end of the “blockage-time”, the smart bus will again advise the hub to resume the transmission. Of course, many factors contribute to the estimation of the transmission suspending time: bus speed, environment type, constraints imposed by the traffic class, and so forth. The error made in predicting such an instant is due to the transmission delay between the bus and the satellite. We compensate for such an error by introducing a suitable guard time, which slightly worsens the system performance.

The aim of this paper is determining the most appropriate UL-FEC parameters, together with the interleaver depth, for video streaming transmissions in a satellite mobile channel, and selecting the buffering strategies to take advantage of the knowledge of no-reception time periods. Furthermore, this paper highlights “how smart” the system is in terms of saved bandwidth, by comparing the smart behavior with that of a bus that does not take advantage of the route information. The determination of the above quantities only involves the study of a single channel between a generic bus and the hub. Therefore, we concentrate on this, and do not deal with the multiplexing and multicast transmission problems connected with the delivery of multiple video streams to a multitude of buses.

To reach our goals, we first model the satellite mobile channel for different environments where the bus can move (Section 2); then, we study the mechanisms for video streaming transmission (Section 3), and we make some considerations on the application of the interleaving technique (Section 4). How smart the system is, in terms of saved bandwidth, is shown in Section 5. Our considerations on the results obtained are summarized in Section 6.

2. The Mobile Channel Model

One of the goals of this paper is to quantify packet error rates experienced in the nonblocked states of the channel, in order to make possible choosing the best operating conditions for the video streaming transmission. This means choosing the modulation and coding scheme (MODCOD), and the Reed-Solomon Erasure (RSE) [6] block length and code rate, for the E_s/N_0 (symbol energy over two-sided noise power

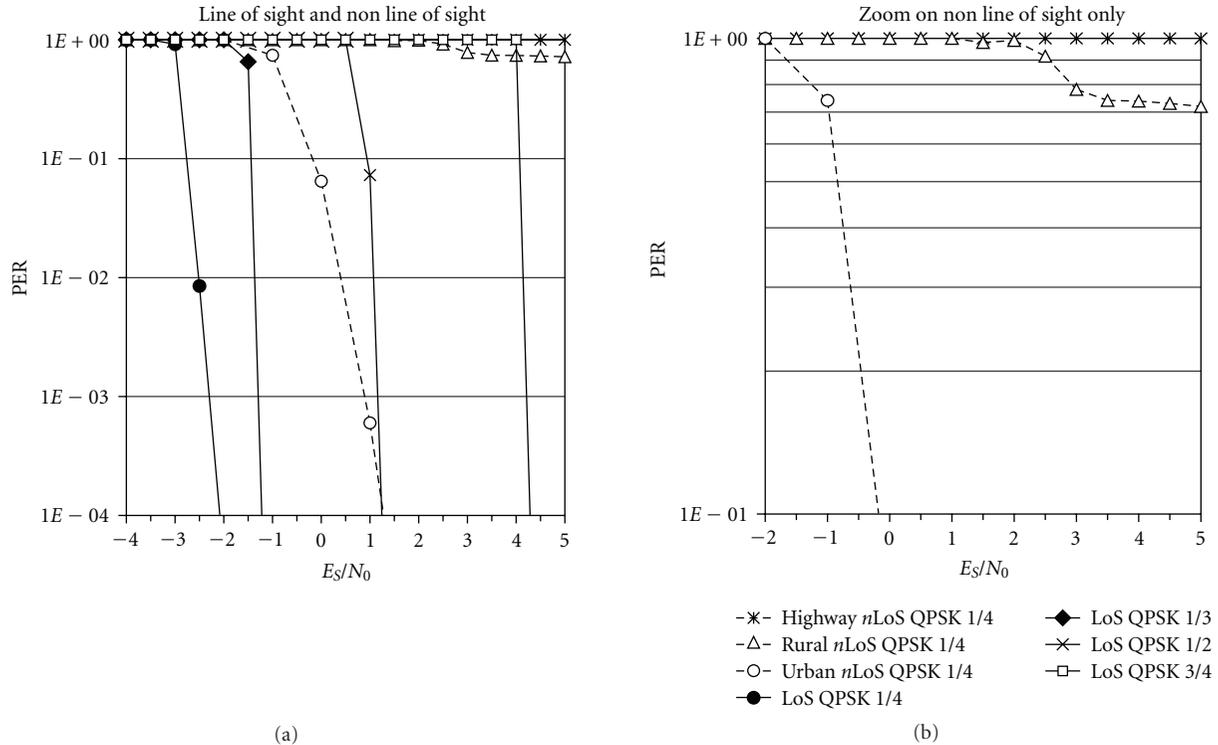


FIGURE 2: Packet error rate for mobile DVB-S2 in LoS and shadowed states.

spectral density) ratio available at the receiver. To model the mobile satellite channel at the packet level, we adopt the 3-state Discrete-Time Markov Chain propagation channel model provided by the European Space Agency (ESA) for the Ku-band satellite channel availability to mobile users in four environments [4, 5]: urban, suburban, rural, or highway. Suburban and rural environments present quite similar characteristics, as it results from the transition matrix in Table 1, which shows that the transition probabilities—and, thus, the steady-state probabilities—are almost the same for these two environments. For this reason, between the two, we chose considering the rural case only. Urban and highway have completely different characteristics, as they differ in the average vehicle speeds (higher in the highway, lower inside towns), other than in the horizon profile, mainly owing to buildings; these concepts are confirmed by the transition probabilities in Table 1.

For each environment, Table 1 reports the state transition matrix of the ESA mobile channel model [4]; the state transition period T_s is equal to 41.7 ms, which corresponds to blocks of 1000 samples at the sampling frequency of 24 kHz, at the average speeds of the vehicle indicated in the table. In each environment, the three states of the Markov chain are named Line of Sight (LoS), Shadowed (Shd), and Blocked (Blk), respectively. The parameters in the four environments essentially depend on the morphology of the landscape and the average speed of the vehicle.

In order to evaluate the performance of our proposed strategies in the mobile DVB-S2 environment, we developed

a simulation model in Simulink starting from the Communication Demos DVB-S2 [7] and according to the parameters reported in Table 1. The simulation model takes into account a standard DVB-S2 transmission/reception chain [1], thus including LDPC [8] and BCH [9] FEC codes in order to simulate the ESA channel model. The DVB-S2 FECFRAME length was set to 64 800 bits, which is the normal frame size in DVB-S2, and includes the BBFRAME (BBheader + Data Field), the LDPC FEC, and the BCH FEC. Figure 2 reports simulation results of the Packet Error Rate (PER) versus E_s/N_0 , relative to a set of DVB-S2 MODCODs in LoS and Shd conditions. We assume to operate with MPEG-2 transport streams (TS), which we will refer to as “packets” throughout the paper. The DVB-S2 BBFRAME can contain more than one of such packets. However, owing to the multiplexing capability allowed by the DVB-S2 standard (Merging/Slicing policy), it is reasonable to assume that a BBFRAME contains no more than one packet belonging to a single stream flow. In this case, the PER relative to the stream coincides with the BBFRAME Error Rate. The range of the x -axis accounts for a link budget that is prevalently limited by the antenna size and by regulatory issues [4], which impose a maximum power flux density on the earth’s surface and, consequently, a maximum effective isotropic radiation power (EIRP) from the satellite. The authors carried out this simulation study because, to the best of their knowledge, no PER evaluation in a mobile environment was available in the literature at the time of writing; in fact, the novelty consists in implementing the DVB-S2 standard in a mobile environment.

TABLE 1: Ku-band land vehicular channel model parameters [4, 5].

Environment	State transition matrix	Steady state probability	Rice factor (K) ¹	Average speed	Doppler shift
	$\begin{pmatrix} p_{\text{LoS,LoS}} & p_{\text{LoS,Shd}} & p_{\text{LoS,Blk}} \\ p_{\text{Shd,LoS}} & p_{\text{Shd,Shd}} & p_{\text{Shd,Blk}} \\ p_{\text{Blk,LoS}} & p_{\text{Blk,Shd}} & p_{\text{Blk,Blk}} \end{pmatrix}$	$\begin{pmatrix} \pi_{\text{LoS}} \\ \pi_{\text{Shd}} \\ \pi_{\text{Blk}} \end{pmatrix}$	$\begin{pmatrix} \text{LoS} \\ \text{Shd} \\ \text{Blk} \end{pmatrix}$		
Highway	$\begin{pmatrix} 0.9862 & 0.0138 & 0.0000 \\ 0.1499 & 0.8378 & 0.0123 \\ 0.0008 & 0.0396 & 0.9596 \end{pmatrix}$	$\begin{pmatrix} 0.8929 \\ 0.0821 \\ 0.0250 \end{pmatrix}$	50.12 0.69 0.00	100 km/h	1342.87 Hz
Rural	$\begin{pmatrix} 0.9795 & 0.0204 & 0.0001 \\ 0.1007 & 0.8277 & 0.0716 \\ 0.0010 & 0.1813 & 0.8177 \end{pmatrix}$	$\begin{pmatrix} 0.7794 \\ 0.1581 \\ 0.0625 \end{pmatrix}$	50.12 1.32 0.00	70 km/h	940.45 Hz
Suburban	$\begin{pmatrix} 0.9796 & 0.0204 & 0.0000 \\ 0.0929 & 0.0929 & 0.0500 \\ 0.0015 & 0.1876 & 0.8109 \end{pmatrix}$	$\begin{pmatrix} 0.7834 \\ 0.1713 \\ 0.0453 \end{pmatrix}$	63.10 1.82 0.00	40 km/h	537.6 Hz
Urban	$\begin{pmatrix} 0.9902 & 0.0097 & 0.0001 \\ 0.0714 & 0.8756 & 0.0529 \\ 0.0000 & 0.0140 & 0.9860 \end{pmatrix}$	$\begin{pmatrix} 0.6014 \\ 0.0825 \\ 0.3161 \end{pmatrix}$	50.12 2.30 0.00	30 km/h	403.2 Hz

¹The Rice factor K is defined as the ratio of signal power in the dominant component over the scattered reflected power. It determines the distribution of the received signal amplitude. The knowledge of the Rician K factor allows determining the bit error rate of a channel, among other metrics.

In LoS, curves are quasivertical, and are not affected by a change in the environment. The operating MODCOD is determined by the available E_s/N_0 , which depends on the symbol rate, given the C/N_0 (carrier power over two-sided noise power spectral density) ratio that results from the link budget. In the Shd state, for all cases, except the urban one, the PER does not sufficiently decrease to acceptable values, for obtainable levels of E_s/N_0 , even for the most robust MODCOD of DVB-S2, that is, QPSK 1/4. In all such cases, then, the Shd state is equivalent to Blk.

Hence, by choosing the appropriate MODCOD for the available value of E_s/N_0 , the channel behavior at packet level can be seen as a 2-state Discrete-Time process. It is worth noting that, in reducing the Markov chain from 3 to 2 states, by merging Shd and Blk states, the Markovian property is generally lost; we will examine this in detail further on in the section. The two states are a “Good” one (LoS) and a “Bad” macro-state (which includes Shd + Blk). The only exception to this assumption is the urban case, in which the Rice Factor K is sufficiently high. In this case, we note in Figure 2 that, when E_s/N_0 is about 1 dB, the PER in the Shd state is close to 10^{-4} (which is equivalent to that of a Good state) for QPSK 1/4; the same PER is obtained in LoS with QPSK 1/2, which allows operating at double speed with respect to QPSK 1/4.

Note that, in DVB-S2, QPSK is the maximum energy efficient modulation scheme, and LDPC = 1/4 is the maximum coding protection.

Then, there are two possible choices. At QPSK 1/4, the LoS and Shd states can be merged into one “Good” state (which we may call BG, for “Big Good”); at QPSK 1/2, the Shd state becomes practically indistinguishable from the Blk one, as in the previous situations. It is worth noting that considering the Shd state as a Good one (which implies higher physical layer redundancy) allows using a lower UL-FEC (RSE) redundancy, with respect to considering the Shd state as a bad one (BB, for “Big Bad”), to the end of obtaining the same final PER. As we mentioned earlier, aggregating two or more states of a Markov chain into a macro-state generally does not preserve the Markovian property in the new reduced process.

Figure 3 highlights the probability mass function (PMF) of the sojourn time in the new macro state in both BB and BG cases (jittered line, obtained by simulation), in comparison with an exponential distribution with the same average sojourn time τ_{BB} or τ_{BG} (straight line). The PMF gives the probability that the discrete random variable is exactly equal to some value.

In addition, we provide further *hypoexponential* fittings of the PMF in the two cases, which are plotted in the two graphs by means of a white interpolating line; the expression of the fitting curve is

$$F(t) = \frac{a}{\tau_1} e^{-t/\tau_1} + \frac{b}{\tau_2} e^{-t/\tau_2}. \quad (1)$$

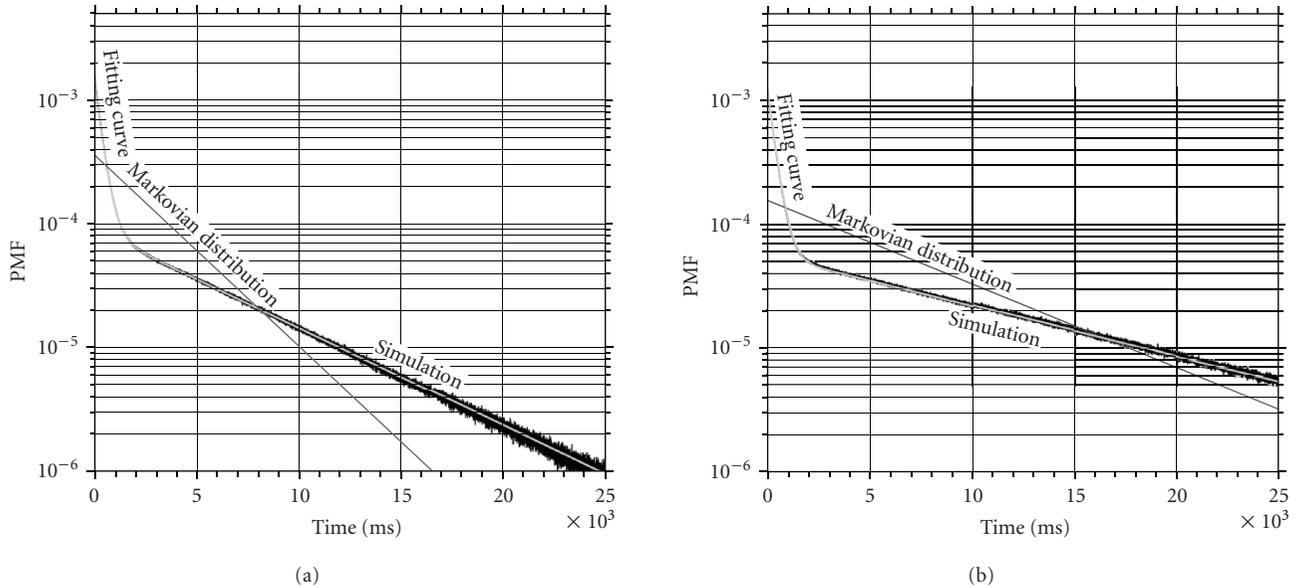


FIGURE 3: Probability mass function of the sojourn time in BB (a) and BG (b) macro-states in the urban environment (the PMF of a Markovian state with the same mean value is also shown; the white line represents the interpolation fitting curve of simulation results).

TABLE 2: Parameters of the hypoexponential distribution.

BB parameters	BG parameters
$\tau_1 = 2.979$ s	$\tau_1 = 2.979$ s
$\tau_2 = 5.495$ s	$\tau_2 = 10.700$ s
$a = 0.463$	$a = 0.604$
$b = 0.529$	$b = 0.399$

Table 2 provides the relative parameters that define the hypoexponential curve for each distribution.

3. Mechanisms for Video Streaming Transfer

Video on Demand is a typical unicast streaming service, similar to a file transfer, but concerning video or audio (such as YouTube), where a user downloads contents while watching them. Generally, this kind of service is quite delay-tolerant; short periods of congestion or link failures may be compensated for with opportune playout buffering, which filters the delay jitter caused by the variability of both source rate and channel bandwidth [10]. Live Video Streaming (LVS) refers to those broadcasting streaming services that produce data that are not prerecorded, and cannot be delayed so much, such as newscasts, alerts, last minute offers, and so on.

Figure 4 shows the Tx/Rx chain of the DVB transport streams. In case of LVS, a video camera, which can be deployed anywhere, transmits a video to the hub; in the VoD case, data can reside in the hub or in a remote streaming server. The Tx and Rx buffers shown in the figure are needed to store packets whenever necessary during the smart mode operations. The buffering of the k packets of each codeword is handled within the RSE encoder and decoder. We do not

deal with the playout buffer, which is contained inside the playout device; we treat the Tx and Rx buffers only.

The smart mode behavior mentioned previously just refers to the set of features driven by feedback commands that allows for the hub stopping packet transmissions during long periods of channel blockage of the bus, and resuming them at normal speed (r_n) or at a higher speed (r_h), when the buffer of the receiving bus has to be refilled. The reason for the two speeds is explained in the following.

At the receiver on the bus, correctly demodulated DVB packets fill the deinterleaver, which flags as *erasures* all packets between two valid ones; that is, when a Transport Error Indicator (TEI) is set to 1 by the demodulator (as it cannot correct errors in the stream), all packets in-between this packet and the next one containing a correct sync byte are flagged as *erasures*. Then, blocks that contain both valid packets and erasures are passed to the RSE decoder. The RSE (n, k) decoder, which is located at the output of the deinterleaver, is able to rebuild k information packets if any k , out of a block of n packets (codeword), are correctly received. The coder first sends k information packets, followed by $n-k$ redundant ones. Thus, if the first k packets are correctly received, no decoding actions are needed. If k or less than k out of n valid packets are received, the decoder delivers a block of k or fewer packets to the playout device (normal mode) or to the Rx buffer (smart mode).

The different characteristics of the LBVS and VoD traffic impose a distinction. We note that, given the presence of the functional blocks shown in Figure 4, the latency in the beginning of video playback (start-up delay) results in the sum of three components: the interleaver filling time, the RSE coding/decoding time, and the Rx buffer filling time. We denote by $stdelay$ the value of the sum of these three time components. In other words, $stdelay$ is the end-to-end delay

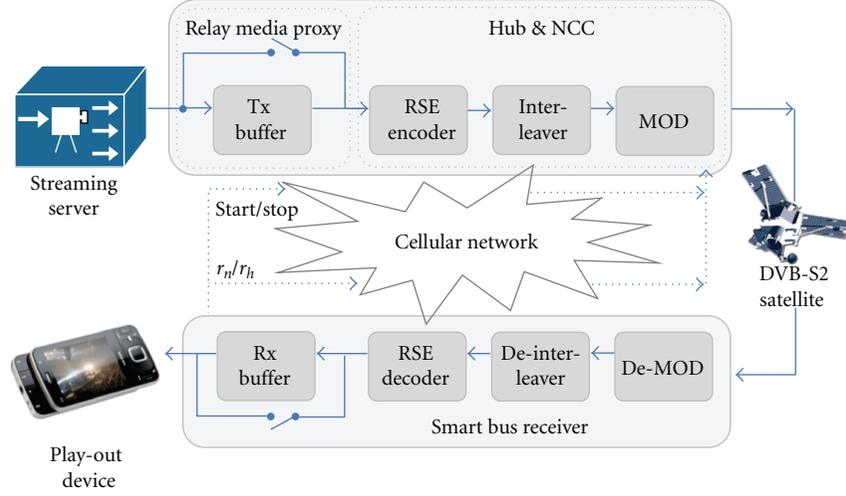


FIGURE 4: Tx/Rx chain for DVB communication to a vehicular device.

in the video playback that we pay to reduce the overall packet loss within the acceptable limit of 2% [11]; $stdelay$ is suitably “spent” in the three components in different proportions, according to the various cases. We call *outage* the set of time periods in which certainly there are no good packets for video playback; this occurs at all times in which the bad channel duration exceeds the $stdelay$ interval. In fact, in this case, invalid packets fill the whole deinterleaver and no good ones get out of the RSE decoder.

By considering a continuous-time approximation of our channel model, we can calculate the outage probabilities, given the threshold $stdelay$ and the probability density functions of the sojourn times in the blocking states associated with the BB and BG cases (t_{BB} and t_B , respectively); we also indicate with τ_B and τ_G the respective average durations of the bad and good intervals:

$$P_{\text{Outage}}^{\text{BG}} = \Pr\{t_B > stdelay\} = \frac{1}{\tau_{\text{BG}} + \tau_B} \times \int_{stdelay}^{\infty} (\vartheta - stdelay) \frac{1}{\tau_B} e^{-\vartheta/\tau_B} d\vartheta, \quad (2)$$

$$P_{\text{Outage}}^{\text{BB}} = \Pr\{t_{\text{BB}} > stdelay\} = \frac{1}{\tau_{\text{BB}} + \tau_G} \times \int_{stdelay}^{\infty} (\vartheta - stdelay) F(\vartheta) d\vartheta, \quad (3)$$

where

$$\begin{aligned} \tau_B &= \frac{T_s}{(1 - a_{33}^{(u)})}, \\ \tau_G &= \frac{T_s}{(1 - a_{11}^{(u)})}, \\ \tau_{\text{BG}} &= \tau_B \frac{(\pi_1 + \pi_2)}{\pi_3}, \\ \tau_{\text{BB}} &= \tau_G \frac{(\pi_2 + \pi_3)}{\pi_1}, \end{aligned} \quad (4)$$

with $\{\pi_1, \pi_2, \pi_3\}$ being the stationary probability distribution of the 3-state Markov chain in LoS, Shd and Blk states, respectively; $a_{ii}^{(u)}$ indicates the i th diagonal element of the transition probability matrix of the discrete-time chain, in the urban environment. By using relations (2) and (3), for the urban environment, we get outage percentages 0.29% and 3.4%, for the BG and BB cases, respectively.

3.1. LBVS Case. In normal mode, the transmission buffer is not present in the hub. In this case, $stdelay$ corresponds to the interleaver filling time at speed r_n plus the RSE coding/decoding time. The packet loss is the one resulting by adopting an RSE (n, k) coder plus an $n \times m$ interleaver of depth m .

In smart mode, the time needed to fill the Rx buffer is equal to $stdelay$, decremented by the interleaver filling time and the RSE coding/decoding time. The interleaver depth is kept smaller than in normal mode, because it must only compensate for short channel interruptions up to a certain threshold (denoted by $maxtb$); emptying the Rx buffer, while the transmission is suspended, compensates for longer interruptions. At the beginning of the process, the Rx buffer filling is done by asking the hub to transmit at speed r_n (and not r_h), because the streaming flow is available at speed r_n . When the Rx buffer is completely filled, the Tx buffer is still empty and the video playback can begin. When, along the route, the bus predicts a channel blockage longer in time than $maxtb$, it sends a stop signal to the modulator in the hub in suitable advance; the hub suspends the transmission and the Tx buffer begins to be filled. At the bus receiver side, the Rx buffer empties out during the whole duration of the blockage period, and the video playback seamlessly continues. When the blockage period has ended, the receiver requires the hub to resume the transmission at speed r_h , up to the complete refilling of the Rx buffer; this time instant coincides with the Tx buffer emptying out, apart from the satellite propagation delay.

3.2. *VoD Case.* In normal mode, all works exactly as in the LBVS case. In smart mode, there is no need of the Tx buffer, because the data are already available on mass storage and they can be passed to the RSE coder at higher speed, whenever needed. The video playback delay can be reduced with respect to the one required in the LBVS case, because the Rx buffer can be filled at speed r_h , even at the start-up time.

4. Some Considerations about Interleaving

In addition to the ACM technique (e.g., MODCOD adaptation), we used UL-FEC and interleaving to improve the transmission efficiency; these techniques are particularly effective in mobile channels, which are characterized by blockage periods. These operations are performed both by adding redundant packets to produce a codeword of suitable length and by scrambling the packet order. Indeed, combining UL-FEC and interleaving allows reducing the codeword size with respect to using UL-FEC alone, just keeping the packet error rate within a target value [12]. In any case, the presence of both interleaving and UL-FEC functionalities introduces a sensible delay in the playback of the stream. In our smart mode, these functionalities are employed to compensate for channel blockages shorter than max_{tb} , while longer blockages are absorbed by the Rx buffer, which is filled before starting the playback. Therefore, there is a compromise between the interleaver depth (which, multiplied by the size of the corresponding codeword, produces the interleaving storage) and the buffer length; both these two items contribute to the maximum start-up delay ($stdelay$).

The effect of the interleaver is the spreading of a train of erasures, or incorrectly received packets, over more codewords, thus improving the error correcting capacity of the RSE code. The higher the interleaver depth, the closer the independent error characteristics are to those of bursty error channels, like the ones we consider in this paper. In fact, upon increasing the interleaver depth, the error characteristics of the environments here considered tend to be as good as in the case of independent errors (or erasures).

Under independent errors, the packet error rate for an RSE (n, k) is given by

$$p^{(\text{ind})} = \sum_{i=n-k+1}^n \sum_{j=0}^{n-k} \left(\frac{i-n+k+j}{k} \binom{k}{i-n+k+j} \right) \times \binom{n-k}{n-k-j} p_e^i (1-p_e)^{n-i}, \quad (5)$$

where p_e is the raw packet error rate. In fact, in a block of n packets, a number of errors i that exceeds the threshold $n-k$ of possible recoverable errors, yields a residual

$$\text{PER} = \frac{i-n+k+j}{k}, \quad (6)$$

where j ($0 \leq j \leq n-k$) expresses all possible divisions of i errors (j and $i-j$) in the redundancy and information fields, respectively.

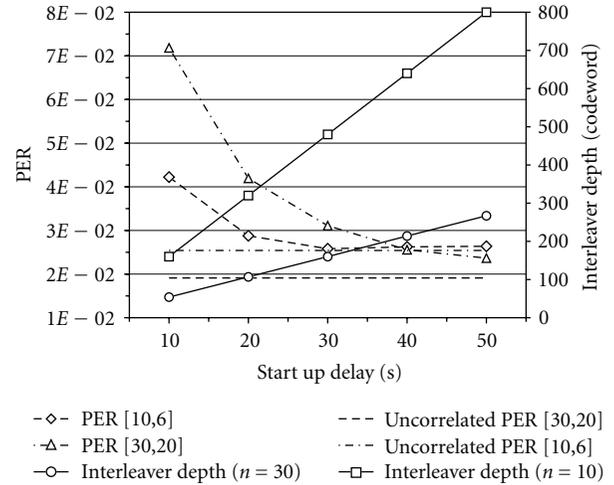
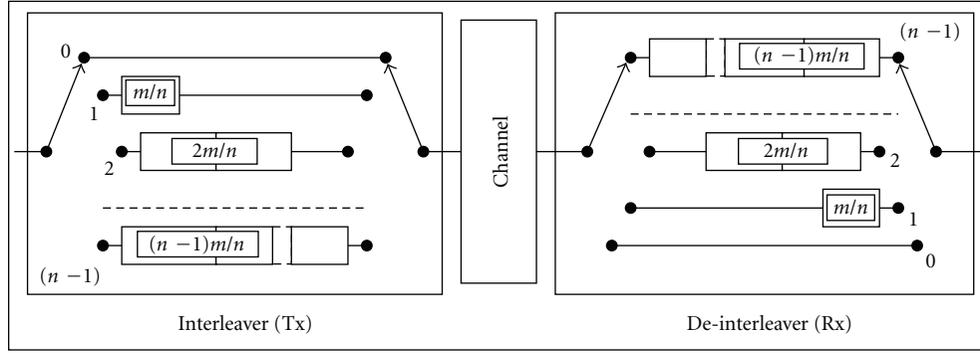


FIGURE 5: Residual PER and interleaver depth versus start-up delay.

In Figure 5, the residual PER characteristics are reported as a function of the start-up delay, which corresponds to a certain interleaver depth (right axis); the figure shows the highway case for different values of the codeword length and coding rate. The comparison is made between sources of independent errors that have the same average PER as the ones of the bursty error channel.

In the case of LBVS, we point out the suitability of using convolutional interleavers, which reduce the overall playout delay with respect to our block-interleaver. Convolutional (or periodic) interleavers were introduced by Forney [13] and similar techniques were proposed, independently, by Cowell-Burton and Ramsey [14, 15]. More recently, they have been investigated in [16, 17] (in the latter, also in relation with turbo codes). As shown in Figure 6, in convolutional interleavers packets from blocks are moved into shift registers before their transmission (interleaver) and before their reblocking at the Rx site (deinterleaver).

The stage of the shift registers goes from $0, 1, \dots$, up to $n-1$ units of m/n packets, in a triangular-wise fashion, and in such a way that the sum of the stages of the Tx and Rx registers that belong to the same transmission chain line is equal to $n-1$ units of m/n packets for all chain lines. The four deviators are simultaneously switched at each packet transmission time in a cyclic fashion and following the ascending order of the chain lines. The time interval between the channel crossings of two consecutive packets belonging to the original block is equal to m packet sending times; thus, an error train of m packets affects one packet of each of m blocks of n packets. All coded blocks of packets are delayed by $n(m-1)$ packet sending times, plus the channel latency. In block interleavers, packets are laid down in the columns of an n row by m column matrix, and sent out from rows. A block is available at the receiver side, in which another $n \times m$ matrix is read from columns, after a delay that is equal to that introduced by a convolutional interleaver. Also the time interval between two consecutive packets is the same in both cases.

FIGURE 6: $n \times m$ convolutional interleaver/deinterleaver.

The difference in the overall playout delay, between the two types of interleavers, arises in the case of LBVS. Here, in fact, the transmission of the Tx matrix of packets of a block interleaver can begin only after that the first k rows of information packets have been collected by the real-time streaming flow. In the convolutional interleaver case, instead, it is sufficient to wait the time relative to n packets sending before beginning the transmission. The rationale behind the last assertion is that the block encoding time is lower than k packet transmission times.

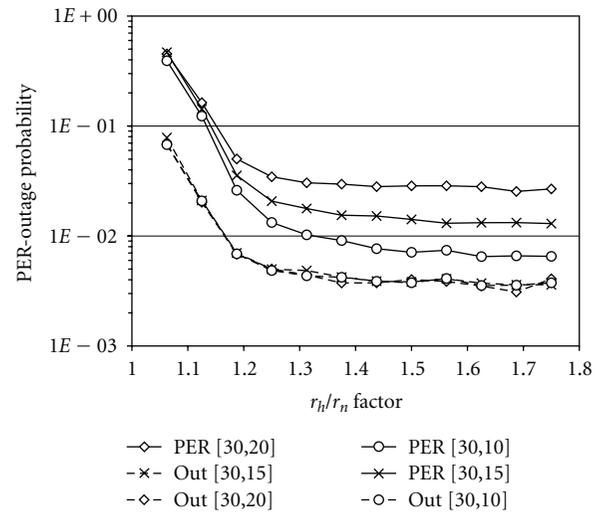
In the VoD case, the video stream comes from a file, stored somewhere, that may be available in the memory at the Tx site in a negligible time; thus, the transmission can start almost immediately in both block and convolutional cases.

5. Benefits of Working in Smart Mode

In order to evaluate the performance of our smart system and choose the best transmission parameters, we carried out a set of simulations. Even if the simulation campaign is not exhaustive, we considered three different environments, which are the most significant ones. In fact, they differ in the landscape profile and in the average speed of the vehicle, specifically: highway, rural (generally suburban cases), and urban. In particular, we divided the urban case in two different subcases, both operating at E_s/N_0 higher than 1 dB:

- (1) *Big Bad* (BB). We assume working with code rate $1/2$ ($E_b/N_0 = 1$ dB, E_b/N_0 being the bit energy over two-sided noise power spectral density); in this case, we cannot distinguish the *Shadowed* state from the *Blocked* one, as they result merged together into a big bad macro state where packets are not received.
- (2) *Big Good* (BG). We assume working with code rate $1/4$ ($E_b/N_0 = 4$ dB); the information bit rate results $1/2$ the BB one, because the redundancy is doubled, but it allows for merging *Line of Sight* and *Shadowed* state into a big good macro states where all packets are correctly received.

As far as the playout parameters are concerned, we chose the start-up delay ($stdelay$) as the maximum time the user can accept to delay the playout. This delay is composed by three

FIGURE 7: PER and outage probability (Out) versus r_h filling rate, for a 30-packet codeword in urban Big Good state; in the legenda, the couple (n, k) indicates the codeword length (n) and the information block length (k) , respectively.

items: (i) $tstore$, that is, the time used by the smart system for prebuffering the video content; (ii) the interleaving time, that is, the delay introduced by the interleaver; (iii) the UL-FEC coding/decoding time.

The value of $stdelay$ strictly depends on the duration of the channel blockage periods. We chose the value of 9 seconds for both the rural and highway environments, whose value is below the key performance parameters suggested by ITU G.1010 [11]. Nevertheless, we were obliged to raise this value to 14 seconds for the urban environment, in order to get an acceptable PER level, which in [11] has been fixed below 2% for a video with data rate between 20 and 384 kps. When the blockage period overcomes $stdelay$, we consider the residual blockage time as an outage.

The value of $tstore$ results from the estimation of the best interleaver depth, which is the number of registers that are filled with a codeword each. Indeed, the best choices for both the interleaver depth and $tstore$ generally depend on the applied redundancy, given a certain codeword length.

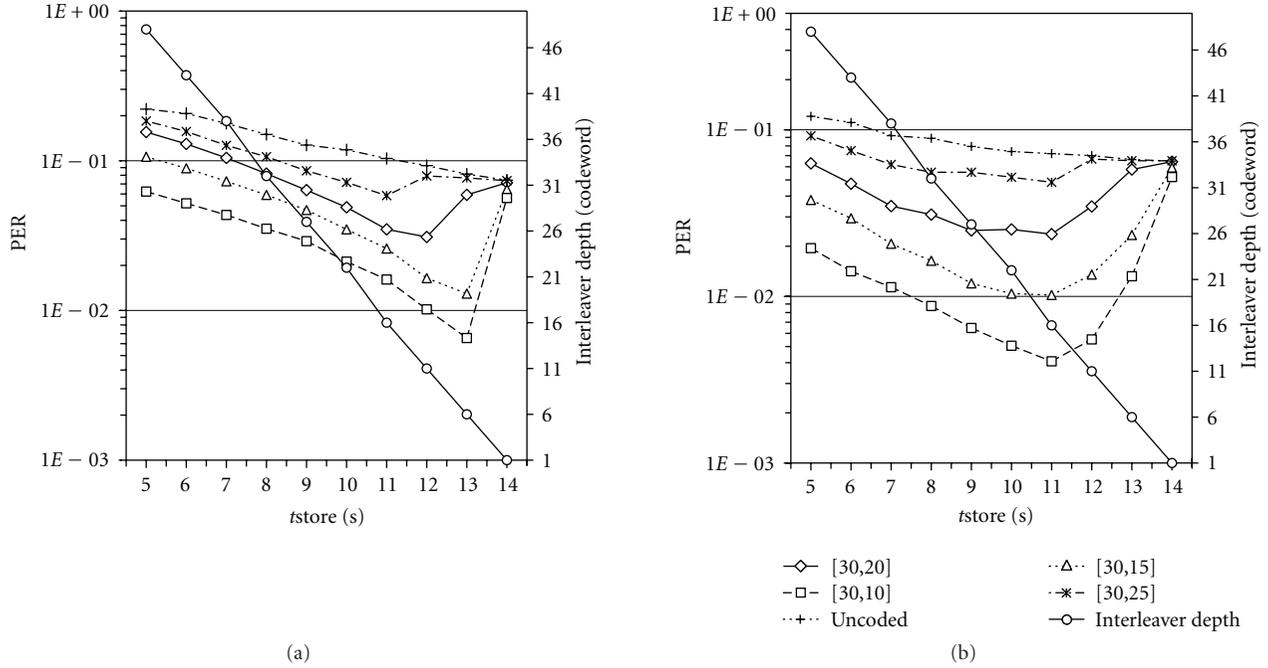


FIGURE 8: Packet error rate and interleave depth versus t_{store} , for a 30-packet codeword in urban Big Bad state (a) and Big Good state (b). In the legends, the couple (n, k) indicates the codeword length (n) and the information block length (k) , respectively.

TABLE 3: Simulation parameters.

Parameter	Value
Packet rate	160 MPEG2-Pkt/s (241 kbps)
Modulation	QPSK
DVB-S2 LDPC coder identifier	1/2 (suburban, urban BB) 1/4 (urban BG)
DVB-S2 BBFRAME Data Field	16008 bits (urban BG) 32208 bits (suburban, urban BB)
$stdelay$	9 s (suburban) 14 s (urban)
$gaptb$	0.5 s
UL-FEC codeword	(30, 10) MPEG2-Pkt

Our simulation parameters are given in Table 3. We used a guard time $gaptb$ in order to manage the reestablishment of the communication between the bus and the satellite after a blockage period longer than 0.5 seconds. We set $gaptb$ to a value of 0.5 seconds, which seems to be reasonable; however, simulations with $gaptb$ set to 1 second did not produce significant changes in the results.

The packet rate reported in Table 3 is relevant to the normal speed (r_n). The assigned r_h should be sufficient to fill the Rx buffer as fast as possible in good periods. The lower r_h , the higher the probability is that the buffer is not fully filled before a blocking period; thus, the system could not manage a blocking period as long as t_{store} , with a resulting increase in both mean PER and outage probability. Figure 7 shows PER and outage probability as functions

of the r_h/r_n ratio. It is worth noting that increasing r_h over certain values does not produce any further benefit in terms of PER and outage; therefore, it results in a waste of resources.

Figure 8 shows that the best choice of t_{store} for a 30-packet codeword results to be about 11 seconds in BG, and ranges between 11 and 13 seconds in BB. By now, t_{store} is optimized for each RSE redundancy level, in order to get always the minimum PER. Once chosen t_{store} , the interleave depth is directly determined, for a given $stdelay$.

Figures 9–12 summarize the performance of the analyzed architecture. All simulation results are obtained with a 95% confidence interval, whose width never exceeds 5% of the estimated value.

We chose two different codeword sizes: 30 and 10 packets, respectively. As resulting from the figures, the PER does not substantially differ, under the same coding rates, between the two values; however, the choice is not straightforward. In fact, while the smaller codeword is less costly in terms of coding/decoding time—and hence in power efficiency—the larger one permits choosing the redundancy with a smaller granularity. For this reason, we report results with both codeword lengths.

Figures 9 and 10 refer to the suburban case, while Figures 11 and 12 to the urban one. Each figure compares PER, computed in non-outage periods, Packet Loss Rate (PLR), which accounts for losses due to both corruption and outage periods, and bandwidth occupancy, as functions of the applied redundancy. In highway and rural environments, outage periods, that is, periods longer than the assessed 9 s of $stdelay$, are really unlikely; hence, PLR is not reported in this case, as it practically coincides with PER.

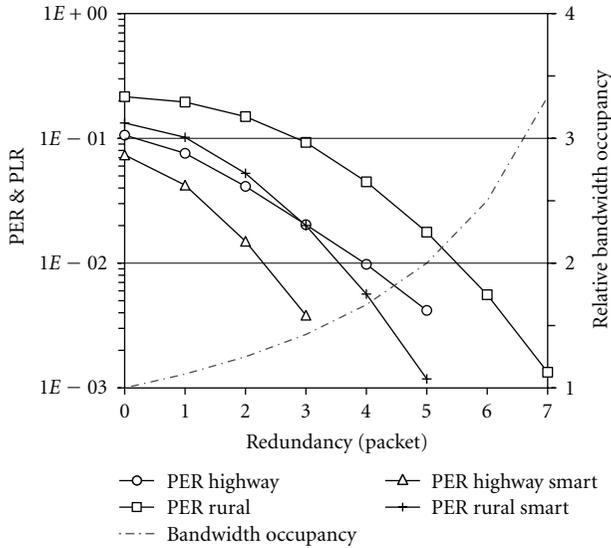


FIGURE 9: PER, PLR, and bandwidth occupancy versus redundancy with RSE (10, k) in highway and rural environments.

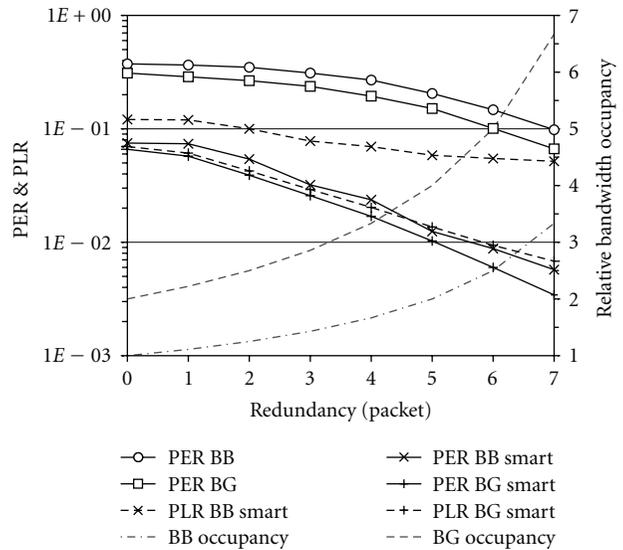


FIGURE 11: PER, PLR, and bandwidth occupancy versus redundancy with RSE (10, k) in urban environment.

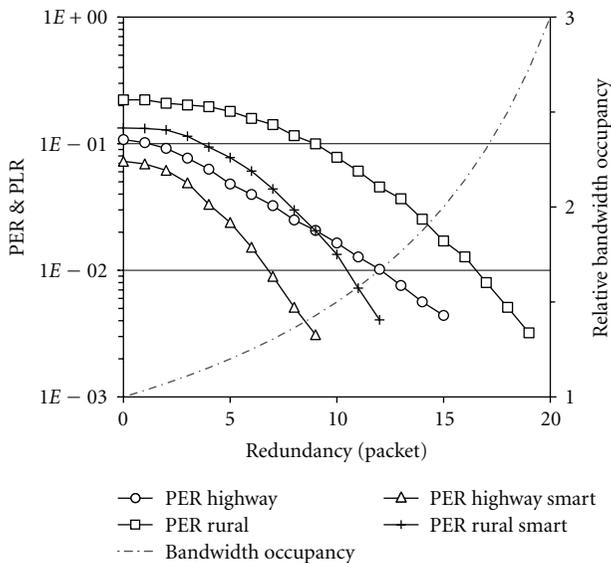


FIGURE 10: PER, PLR, and bandwidth occupancy versus redundancy with RSE (30, k) in highway and rural environments.

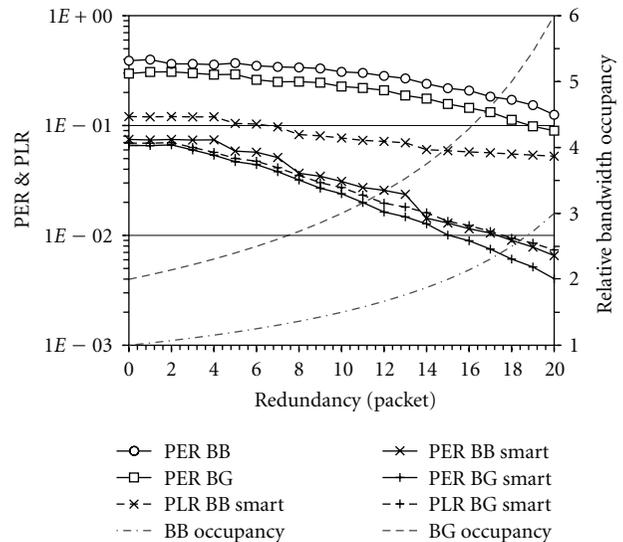


FIGURE 12: PER, PLR, and bandwidth occupancy versus redundancy with RSE (30, k) in urban environment.

In the urban case, instead, very long blockage periods are possible and they impose choosing a longer *stdelay* time, in order to yield an acceptable QoS. The urban case is more complex to be illustrated and more challenging. As previously stated, we can adopt two different MODCODs—1/2 and 1/4, respectively—which produce different overheads. Hence, the relative bandwidth occupancy is very different in the two cases. As a matter of fact, Figures 11 and 12 report two different curves relevant to bandwidth occupancy in BB and BG states. However, the most relevant result is that the smart technique outperforms the sole adoption of UL-FEC

and interleaving in all environments, thus consisting in a notable gain in terms of PLR.

Without the smart technique, it is impossible to get a PER below the acceptable threshold of 2%. There is, instead, a tradeoff between bandwidth occupancy and PLR in choosing between BG and BB. In fact, although the acceptable threshold is reached with 4 and 5 redundancy packets, in the BG and BB cases, respectively, the bandwidth occupancy is much higher in the former case; this is because of the adoption of 1/4 coding rate, which halves the net bandwidth with respect to the BB case. Regarding the PLR,

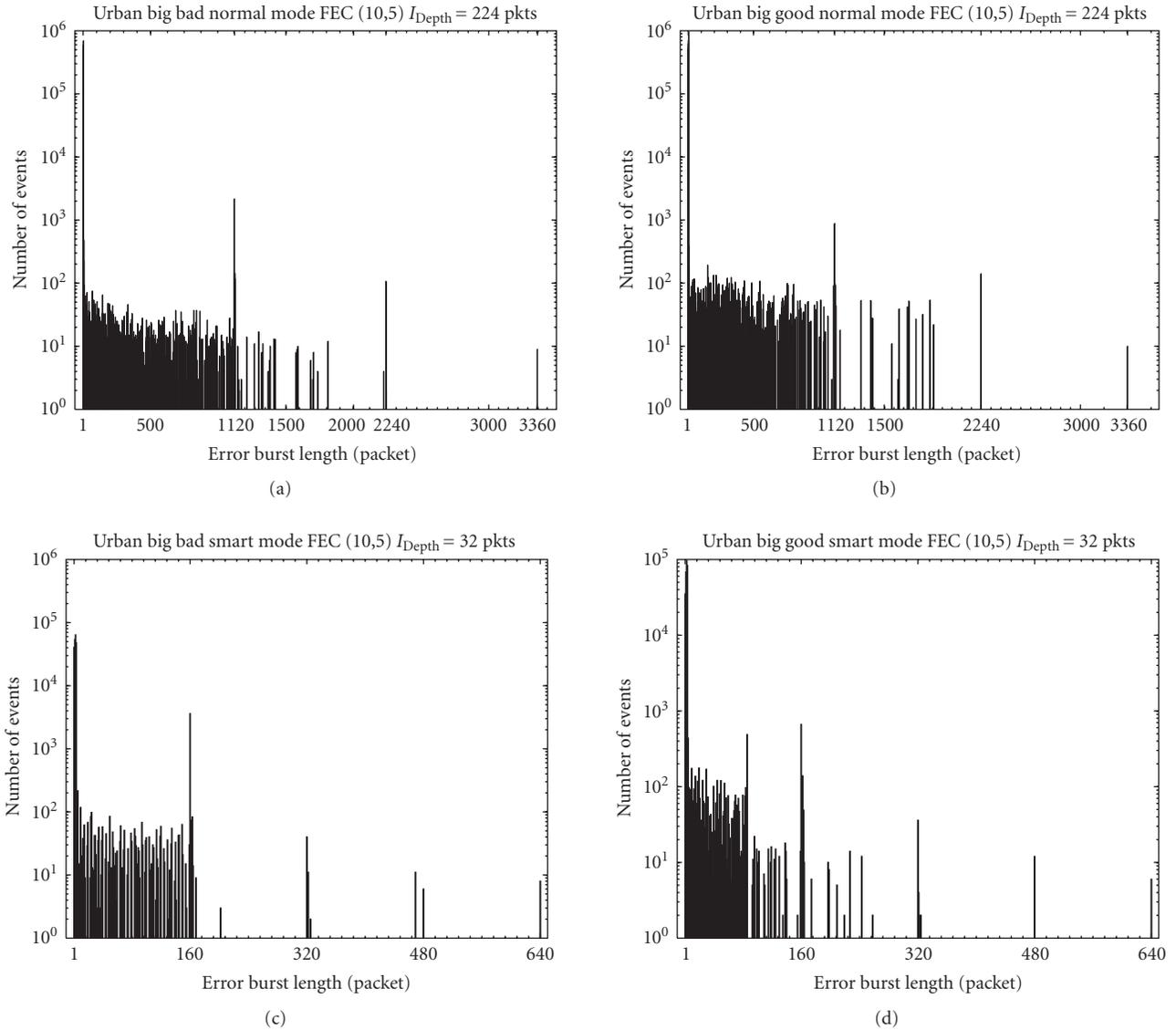


FIGURE 13: Error burst length distributions with FEC (10, 5) in BB and BG and operating in smart mode and normal mode, respectively.

it results much higher than the PER in the BB case, while it results very close to the PER in the BG case. This means that outage periods have a stronger influence in the BB case, due to the heavy tail distribution of blockage times in the bad state of this case (see Figure 3). Very similar results are obtained by using a 30-packet codeword (Figure 12). Here, the acceptable threshold is reached with 10 and 13 redundancy packets in the BG and BB cases, respectively, and, therefore, with coding rates close to the 10-packet codeword case.

Figure 13 reports the distribution of the error burst lengths both in BB and BG cases, when operating in normal and smart modes, respectively. Figure 14 shows the normalized distributions reported in Figure 13 in the most significant range of the x -axis; it highlights that the average error burst length is about four packets with such

parameters. Figure 13 shows some periodic peaks of burst lengths, which are multiples of half of the interleaver size (note that the coding rate is 1/2); this phenomenon occurs when a burst longer than the interleaver size fills the interleaver. In fact, due to the redundancy 1/2 applied in Figure 13, only half of the packets (the information ones) are actually lost. This means that when the interleaver is filled (for example the interleaver of 2240 packets in NM) with erasures, 1120 packets of information are lost (the peak in Figure 13). This event may occur with a probability that can be derived in the PMF of Figure 3, just passing from packet to time domain.

In smart mode, the interleaver is sized in such a way to absorb bursts of errors long up to $maxtb$; in normal mode, $stdelay$ determines the size of the interleaver. Therefore, $maxtb$ and $stdelay$ determine the positions of the peaks in

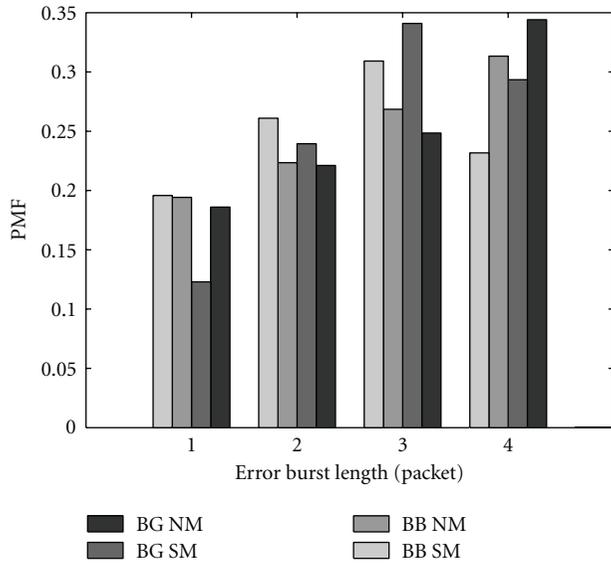


FIGURE 14: Zoom on the distributions of Figure 13 in the x-range of 1–5 packets with FEC (10, 5) and interleaver depth of 32 (SM) and 224 (NM) packets, respectively.

smart and normal mode, respectively. In smart mode, the error burst peaks are more than one order of magnitude shorter than in normal mode, under the same number of events.

6. Conclusions

We proposed a smart architecture for DVB-S2 vehicular connectivity, which allows delivering multimedia contents to fleets of buses. In particular, we studied the video streaming transmissions from a hub station, anywhere located, towards a specific bus reached via a satellite link. The system takes advantage of feedback and prediction of obstacles on the buses' routes provided by GPS navigation devices, installed on board the buses. In addition to classical error recovery techniques, such as forward error correction and interleaving, the proposed smart system enhances the quality of service, by reducing packet error rate and outage periods in both urban and extraurban environments. The proposed technique may be employed as an alternative to gap-fillers, whenever the gap is sufficiently limited in time. This study aims at fostering the development of new communication and entertainment services, as well as monitoring and security issues on future public transportation vehicles, by adapting the most suitable technologies to existing telecommunication infrastructures.

Acknowledgment

This Work was funded by the European Commission in the framework of the SatNEx II Network of Excellence (NoE) FP6 project (Contract no. 027393).

References

- [1] EN 302 307, "Digital video broadcasting: second generation framing structure, coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications," ETSI, DVBS2-74r15.
- [2] "Real Time Streaming Protocol," <http://tools.ietf.org/html/rfc2326>.
- [3] <http://www.openiptvforum.org>.
- [4] ESA Project, "Feasibility study of a mobile Ku-band terminal," Contract No. 15593/01/NL/DS, Final Report, http://telecom.esa.int/feasibility_ndsatcom.
- [5] S. Scalise, H. Ernst, and G. Harles, "Measurement and modeling of the land mobile satellite channel at Ku-band," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 2, pp. 693–703, 2008.
- [6] J. Nonnenmacher, E. W. Biersack, and D. Towsley, "Parity-based loss recovery for reliable multicast transmission," *IEEE/ACM Transactions on Networking*, vol. 6, no. 4, pp. 349–361, 1998.
- [7] <http://www.mathworks.com/products/commblockset/demos.html?file=/products/demos/shipping/commblks/commdvbs2.html>.
- [8] R. Gallager, "Low density parity check codes," *IRE Transactions on Information Theory*, no. 1, pp. 21–28, 1962.
- [9] I. S. Reed and X. Chen, *Error-Control Coding for Data Networks*, Kluwer Academic Publishers, Boston, Mass, USA, 1999.
- [10] K. Fujimoto, S. Ata, and M. Murata, "Adaptive playout buffer algorithm for enhancing perceived quality of streaming applications," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '02)*, vol. 3, pp. 2451–2457, Taipei, Taiwan, November 2002.
- [11] ITU-T Recommendation G.1010, "End-user multimedia QoS categories," <http://www.itu.int/ITU-T/>.
- [12] P. Barsocchi and A. Gotta, "Experimental video broadcasting in DVB-RCS/S2 with land mobile satellite channel: a reliability issue," in *Proceedings of the IEEE International Workshop on Satellite and Space Communications (IWSSC '08)*, pp. 234–238, Toulouse, France, October 2008.
- [13] G. D. Forney Jr., "Burst-correcting codes for the classic bursty channel," *IEEE Transactions on Communications*, vol. 19, pp. 772–781, 1971.
- [14] W. R. Cowell and H. O. Burton, "Computer simulation of the use of group codes with retransmission on a Gilbert burst channel," *AIEE Transactions on Communications and Electronics*, vol. 80, pp. 577–585, 1962.
- [15] J. L. Ramsey, "Realization of optimum interleavers," *IEEE Transactions on Information Theory*, vol. 15, no. 3, pp. 338–345, 1970.
- [16] S. A. Hanna, "Convolutional interleaving for digital radio communications," in *Proceedings of the 2nd International Conference on Universal Personal Communications (ICUPC '93)*, pp. 443–447, Ottawa, ON, Canada, October 1993.
- [17] S. Vafi, T. Wysocki, and I. Burnett, "Convolutional interleaver for unequal error protection of turbo codes," in *Proceedings of the DSPCS & WITSP*, pp. 485–491, Coolangatta, Gold Coast, Australia, December 2003.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

