Multimed Tools Appl
DOI 10.1007/s11042-009-0331-7

A zero-overhead error-correcting nVoD schema

Francisco J. González-Castaño · Rafael Asorey-Cacheda · Héctor Cerezo-Costas · Juan C. Burguillo-Rial · Felipe J. Gil-Castiñeira

© Springer Science + Business Media, LLC 2009

Abstract In this paper we present a novel multicast near-Video on Demand (nVoD) coding schema, which relies on the intrinsic redundancy of the underlying nVoD protocol to provide *implicit error correction*, by employing content segments as blocks for coding operations. As a result, this implicit error correction has zero overhead, unlike the direct application of error-correcting codes, which increase content bitrate in the same proportion as target error probability. The findings in this paper indicate that our proposal outperforms previous approaches with explicit error correction (error protection within content segments) in terms of transmission bandwidth for the same packet loss probability. We present an analytical approach that can be used to tune implicit error correction (coding matrix selection), which we validate with simulations. We also simulate the impact of the coding schema on two different nVoD protocols, fast broadcasting (FB) and recursive frequency splitting (RFS). Finally, we show the benefits of applying this schema to a real scenario with WiMax transport.

F. J. González-Castaño (✉) · R. Asorey-Cacheda · H. Cerezo-Costas · J. C. Burguillo-Rial · F. J. Gil-Castiñeira
Departamento de Enxeñería Telemática, Universidade de Vigo, 36310 Vigo, Spain
e-mail: javier@det.uvigo.es

R. Asorey-Cacheda
e-mail: rasorey@det.uvigo.es

H. Cerezo-Costas
e-mail: hcerezo@det.uvigo.es

J. C. Burguillo-Rial
e-mail: jrial@det.uvigo.es

F. J. Gil-Castiñeira
e-mail: xil@det.uvigo.es

F. J. González-Castaño
Galician Research and Development Center in Advanced Telecommunications, ETSE Telecomunicación, Campus, 36310 Vigo, Spain

Published online: 07 August 2009

Keywords Multicast · nVoD · Error correction

1 Introduction

Significant advances in on-demand media streaming have taken place recently and several protocols for scalable on-demand media streaming have been proposed, including periodic broadcast [17], patching [2, 4] and bandwidth skimming protocols [3, 5]. These protocols are mostly intended for the wired Internet and do not directly deal with the frequent packet losses that occur in wireless networks. In broadcast-oriented protocols, a media file is divided into segments, which are simultaneously broadcast through different channels at different rates according to their impact on content playback delay. Clients thus receive multiple streams at a time, with an aggregate transmission rate that is proportional to the real-time content playing rate.

Typically, Video-on-Demand (VoD) systems serve long contents with high playback rates. Unicast VoD (or true VoD, tVoD) allocates a dedicated stream to every client request. This is desirable to ensure user satisfaction but it is very inefficient in terms of resources required (particularly in terms of network and server I/O bandwidth). Multicast near VoD (nVoD) [8] requires low bandwidth and is well suited for multicast networks. NVoD should ideally approximate the performance of tVoD (it achieves that goal if there are enough resources to guarantee low service delays), but it brings new challenges. One particularly important issue is the time that elapses before a client request is accepted as VoD systems should respond to all client requests in real time. Any noticeable delay before playback starts may dissatisfy users. There have been several proposals to overcome this problem such as best-effort patching [9]—which supports VCR-like interactions and immediate service to user requests—and a mixed batching-patching schema [16], which employs broadcast, multicast or unicast communications depending on the popularity of the contents. Traditional VoD schemas introduce redundancy for error recovery, but such an approach increases bandwidth and may be unfeasible due to high error probabilities [10].

The nVoD coding schemas proposed in this paper inherit all the advantages of traditional server-initiated nVoD systems, such as unlimited scalability and the ability to recover automatically from network failures, but they introduce intrinsic error correction capability at the client side without extra bandwidth cost [1]. At the end of the paper, we show that our schema can recover from a high transmission error probability in a WiMax scenario without impacting channel bandwidth.

The rest of this paper is organized as follows. Section 2 presents our schema, together with an analytical approach to tuning it. Section 3 presents results that validate the analysis in Section 2 and shows the benefits of our approach in a real WiMax scenario, and Section 4 presents our conclusions.

2 Description of the schema

2.1 Segment-mapping algorithm

Our schema relies on a nVoD broadcast protocol. For each content segment, the server reserves in advance an amount of bandwidth that is divided into channels with individual bandwidths that match the content playback rate. The video is

Channel 1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	$\cdots$
Channel 2	S_2	S_3	S_2	S_3	S_2	S_3	S_2	S_3	$\cdots$
Channel 3	S_4	S_5	S_6	S_7	S_4	S_5	S_6	S_7	$\cdots$

Fig. 1 Example of Fast Broadcasting (FB) protocol. The rows and columns represent multicast channels and time slots, respectively. Segment S_1 has a frequency of 1 segment/slot, segments S_2 and S_3 have frequencies of 1/2 segment/slot and segments S_4, S_5, S_6 and S_7 have a frequency of 1/4 segment/slot

split into segments that are periodically transmitted through the different channels according to a *segment mapping algorithm*. Our proposal is compatible with any segment mapping algorithm with fixed-length content chunks. To optimize latency and bandwidth usage, we decided to rely on a Harmonic Broadcasting (HB) protocol [6]. An HB protocol divides a content chunk into equally sized segments across a set of different channels, and assigns each one a retransmission frequency that is the inverse of the segment index. The playback time of each segment, i.e. the slot, is the ratio between the segment bitlength and its consumption rate. Theoretically, given a number of channels l, the number of segments n that HB allocates satisfies:

$$\sum_{k=1}^{n} \frac{1}{k} < l. \tag{1}$$

To minimize bandwidth redundancy in a given implementation, the frequency of a segment must be as close as possible to the inverse of its index ($1/i$ for segment S_i). Figure 1 shows an example corresponding to the FB protocol [7]. There are other HB approaches that outperform FB, such as the New Pagoda Boadcasting (NPB) protocol [11], the Recursive Frequency-Splitting (RFS) protocol [18, 21] and Hybrid Broadcasting [20]. All these schemas require local storage for segments that are not immediately visualized (up to 40% of content). Some algorithms based on the Pagoda approach consider memory management [12–14]. There also exist some recent works that deal with buffer utilization such as [22–24].

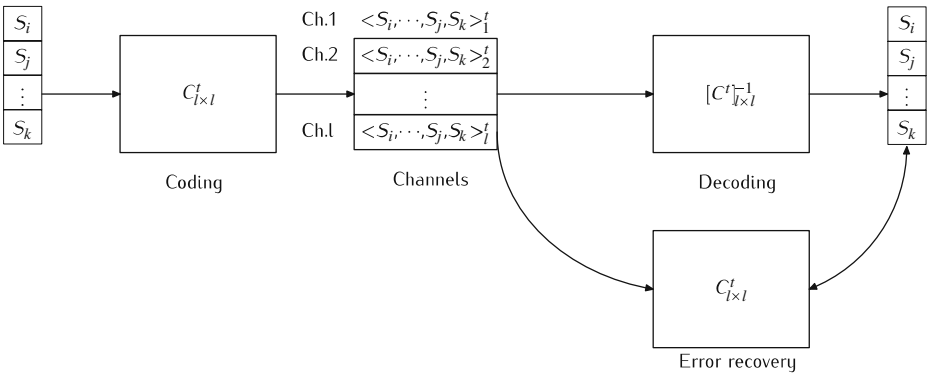

Fig. 2 Implicit error correction coding-decoding block diagram

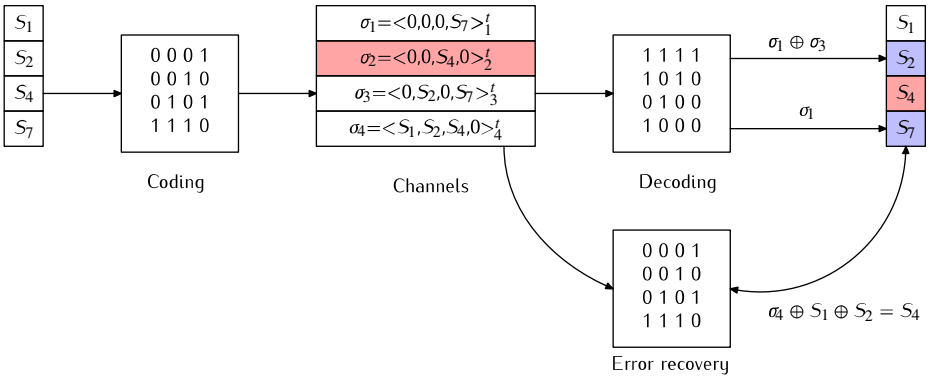

Fig. 3 Implicit error correction. Assuming that S_1 has been previously downloaded and is available, segment packets S_2 and S_7 are normally decoded with the help of $[C^t]^{-1}$. Segment packet S_4 cannot be recovered from σ_2, which is lost. It is obtained instead from the frame in the fourth channel and segment packets S_1 and S_2. There is full packet recovery in the slot, for a 25% frame loss

2.2 Segment coding

We introduce the following basic notation. Video content S is divided into segments, $S = \{S_1, S_2, \ldots, S_n\}$ of equal size. In turn, each segment contains r packets. Thus, $S_i = \{\text{pkt}_i^0, \ldots, \text{pkt}_i^{r-1}\}$. The analytical and simulation studies are performed at packet level rather than segment level (as was also the case in a previous study by our group [1]). For simplicity, since every segment/slot is divided into independent packet subslots, S_i refers to a *segment packet* within segment i hereafter.

The frame $\sigma_m^t = \langle S_i, \ldots, S_j \rangle_m^t$ for channel m and timeslot t, ($S_i \cap \cdots \cap S_j = \emptyset$, $\text{card}(\{S_i, \ldots, S_j\}) \leq l$, where card() is the cardinality of a set) results from the following operation on the original segment packets $S_i \ldots S_j$:

$$\langle S_i \ldots S_j \rangle_m^t = C_{m,i}^t \cdot S_i \oplus \cdots \oplus C_{m,j}^t \cdot S_j,$$
$$\{i, j\} \in \{1, \ldots, n\}, \left\{C_{m,i}^t \ldots C_{m,j}^t\right\} \in \{0, 1\}, \tag{2}$$

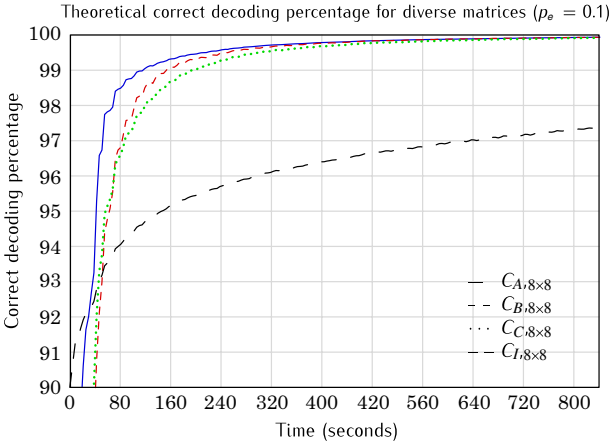

Fig. 4 Theoretical value of correct decoding percentage for $p_e = 0.1$ vs time, for matrices C_A, C_B, C_C and without implicit error correction (C_I), for 1532 segments (4.25 *s*/*slot*)

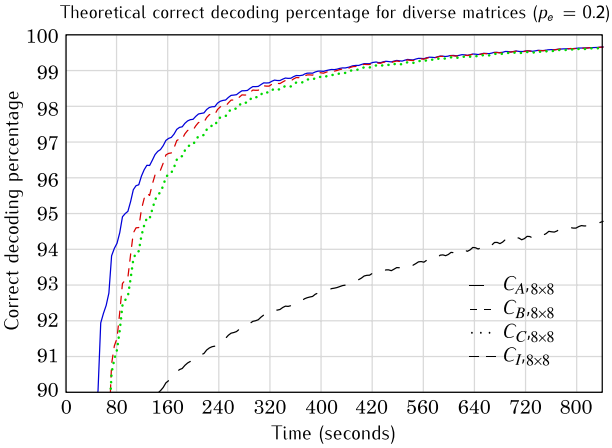

Fig. 5 Theoretical value of correct decoding percentage for $p_e = 0.2$ vs time, for matrices C_A, C_B, C_C and without implicit error correction (C_I), for 1532 segments (4.25 *s*/*slot*)

The coefficients in expression (2) could be integers, but we chose to make them binary for implementation simplicity once we had good results in practice.

Therefore, for l channels, the original segment packets are recovered from the coded ones $\sigma_m^t = \langle S_i \dots S_j \rangle_m^t$, $\sigma^t = (\sigma_1^t, \dots, \sigma_l^t)$, if there exists a matrix $[C^t]^{-1}$ such that:

$$S_i = \left[C^t\right]^{-1} \left[\sigma^t\right]^T \tag{3}$$

For simplicity, the coding coefficients are arranged in a matrix C^t.

$$C^t = \begin{pmatrix} C_{1,i}^t & C_{1,j}^t & \cdots & C_{1,k}^t \\ C_{2,i}^t & C_{2,j}^t & \cdots & C_{2,k}^t \\ \vdots & \vdots & \ddots & \vdots \\ C_{l,i}^t & C_{l,j}^t & \cdots & C_{l,k}^t \end{pmatrix}_{l\times l}, \tag{4}$$

where $C_{x,y}^t$ is the coefficient of channel x and segment y in slot t.

Figure 2 shows the coding-decoding block diagram. If no errors occur, the upper flow in the diagram allows all the original segment packets to be recovered. However, in lossy channels, the decoder uses matrix C to correct errors by taking advantage

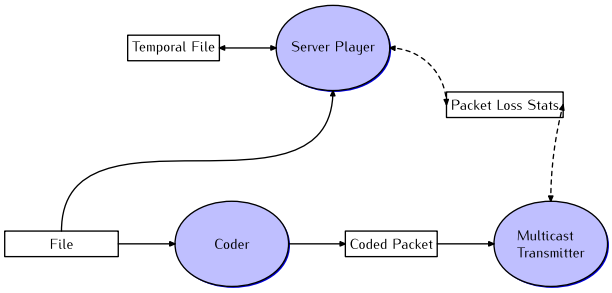

Fig. 6 Multicast VoD server architecture. The server is composed of a coder, a player and a transmitter process

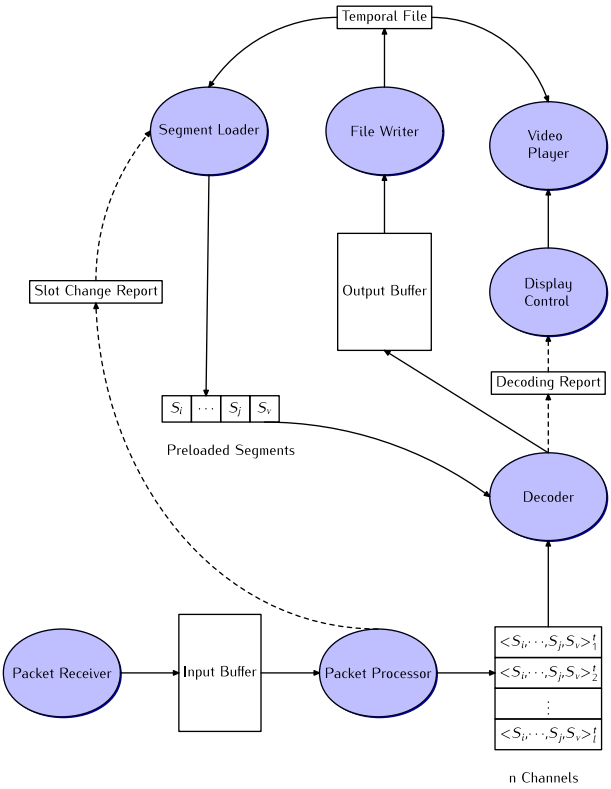

Fig. 7 Multicast VoD client architecture. The client module is composed of eight processes (the decoder is split in two processes) and two tables, one with the transmitted frames and the other with the original segment packets decoded in previous slots

of the information in previously decoded segment packets. Notice that there is no overhead in this process. Figure 3 shows the correcting actions in a scenario with four channels and a lost frame in the second channel.

Our schema can be adapted to VBR nVoD [15] with a receiver buffer of appropriate size.

2.3 Segment coding evaluation—analytical approach

Theoretically, many nonzero coefficients in a column of C imply a higher probability of receiving a frame containing the original segment packet. The system therefore gives greater priority to low-frequency segments by granting them more nonzero coefficients in the corresponding column of C, and then reordering the segment packets given in each slot from higher to lower frequencies. The first segment packet S_1 is always guaranteed to be available in the first slot, and is normally combined with low frequency segments to enhance correction capabilities.

A theoretical evaluation of the matrix in terms of video quality assessment by customers presents a complexity that exceeds the pretensions of this paper. Nevertheless, it is easy to understand that a matrix that recovers more segment packets in a slot is better in terms of user satisfaction. We therefore introduce $\theta_{\kappa,\lambda}$ as the probability of full recovery in a scenario with κ lost frames and λ original segment packets that were obtained in past slots. We assume that the λ available segment

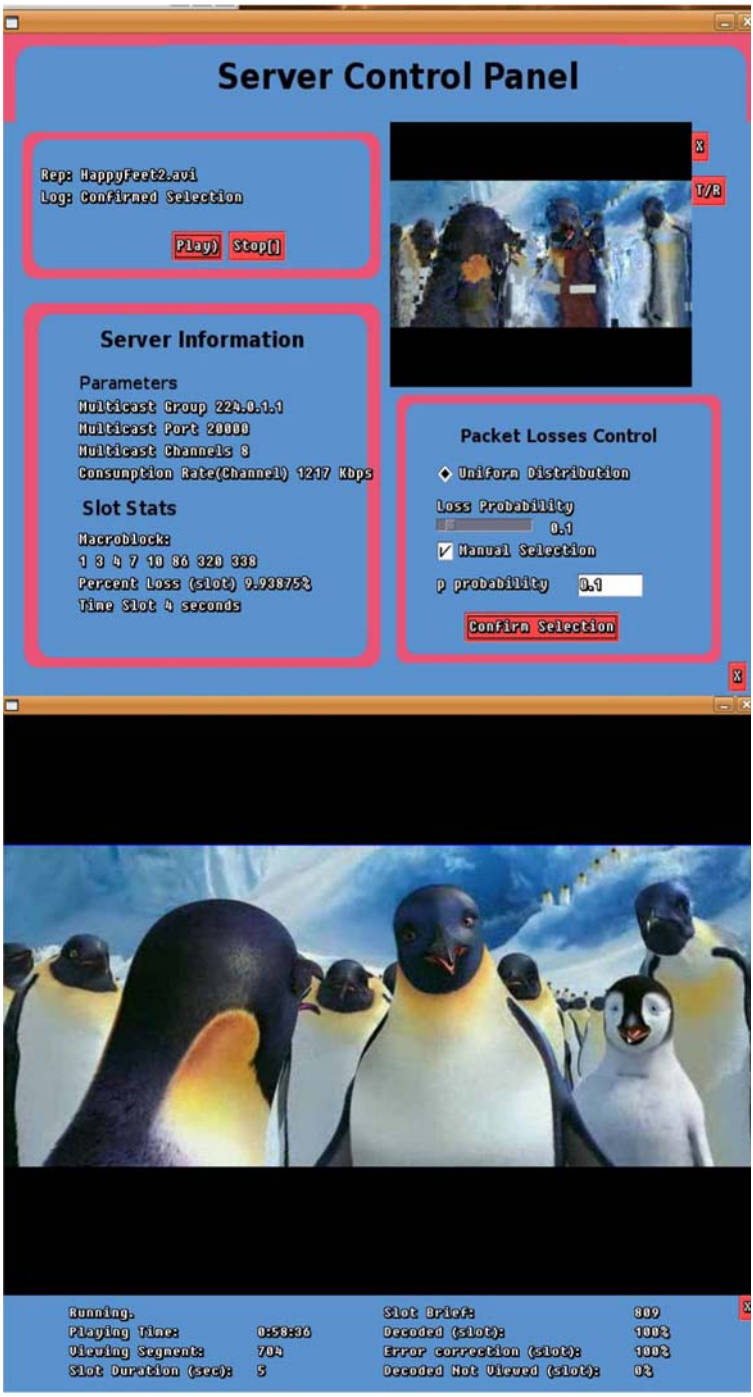

Fig. 8 Server and client SDL interfaces. Scenario with 10% uniformly distributed packet errors. Compare the server player (up, right) with the client player after error correction (down)

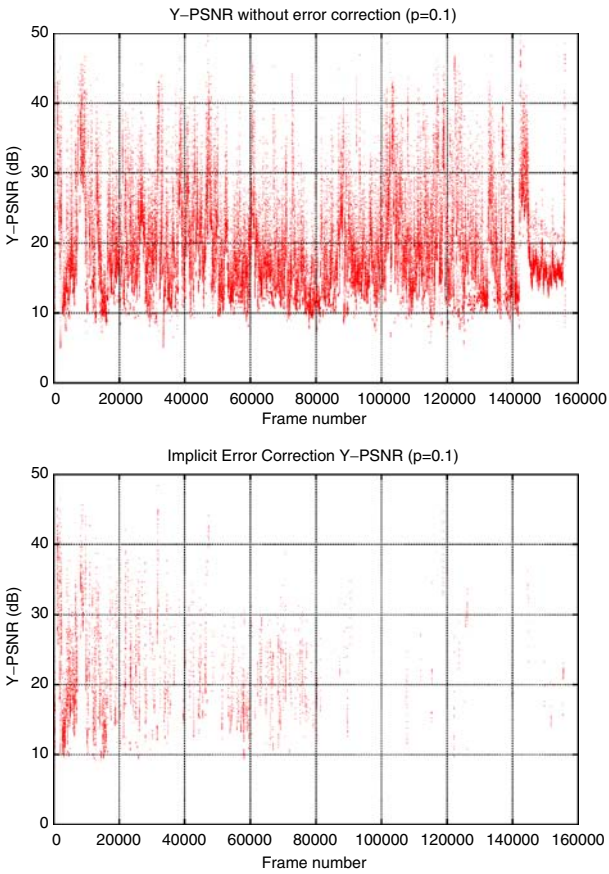

Fig. 9 Y-PSNR for the server player (up) and the client player after error correction (down). Scenario with 10% uniformly distributed packet errors

packets have the highest frequency. Thus, $\theta_{\kappa,\lambda}$ can be computationally determined as the segment recovery probability for all combinations of κ lost frames and a given λ.

The set of coefficients $\theta_{\kappa,\lambda}$ can be arranged in a square matrix Θ whose rank is $l+1$.

$$\Theta = \begin{pmatrix} \theta_{0,0} & \theta_{0,1} & \cdots & \theta_{0,l} \\ \theta_{1,0} & \theta_{1,1} & \cdots & \theta_{1,l} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{l,0} & \theta_{l,1} & \cdots & \theta_{l,l} \end{pmatrix}_{l+1 \times l+1} \tag{5}$$

Θ characterizes the behaviour of the matrix since with a given error pattern and segment mapping algorithm we can determine the percentage of recovered segment packets in a slot.

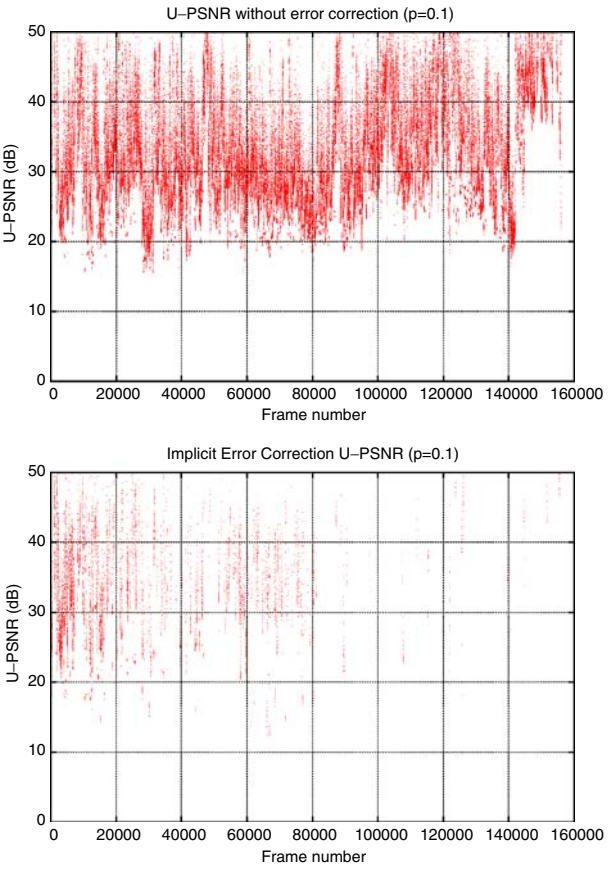

Fig. 10 U-PSNR for the server player (up) and the client player after error correction (down). Scenario with 10% uniformly distributed packet errors

Let $\Pi = \{\pi(0), \pi(1), \ldots, \pi(l)\}$, where $\pi(k)$ is the probability of losing κ frames out of l in the current slot. If p_e is the probability of channel error and it is uniformly distributed, then we obtain the binomial distribution:

$$\pi(\kappa) = \binom{l}{\kappa} p_e^{\kappa} (1 - p_e)^{l-\kappa} . \tag{6}$$

Let $\Psi = \{\psi(0), \psi(1), \ldots, \psi(l)\}$, where $\psi(\lambda)$ is the probability that λ segment packets will be provided in the current slot; this probability depends on the percentage of decoded packets in previous slots. However, to achieve a good coding matrix, we overestimate the availability of a segment packet by assuming it will be available after its second appearance in a frame. In other words, $\psi(\lambda)$ is the probability that $l - \lambda$ segment packets are sent for the first time.

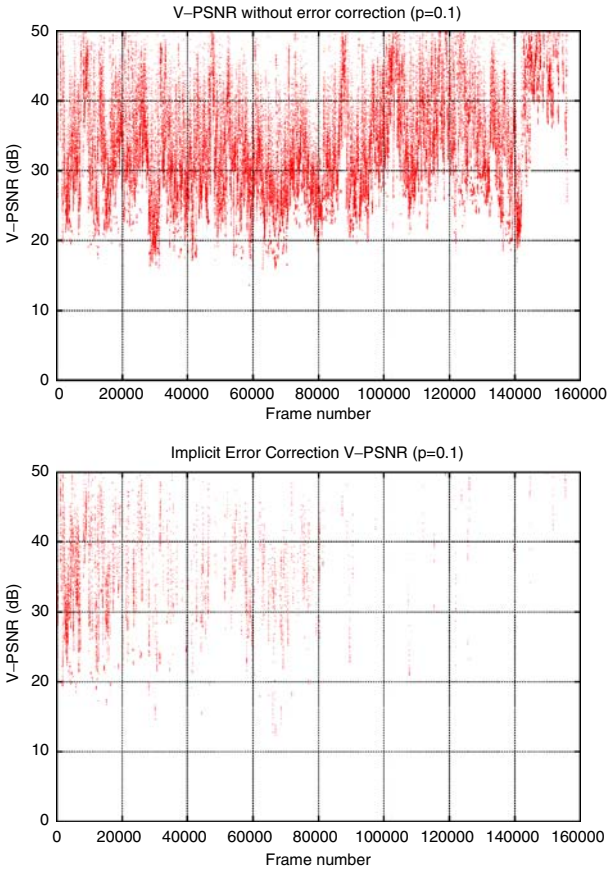

Fig. 11 V-PSNR for the server player (up) and the client player after error correction (down). Scenario with 10% uniformly distributed packet errors

From these assumptions, it is easy to determine the percentage of decoded segment packets in the current slot as:

$$\Pi \cdot \Theta \cdot \Psi^T \tag{7}$$

Figures 4 and 5 represent expression (7) from the beginning of the transmission, for the coding matrices in (8), for a content partition with 1532 segments (4.25 s/slot), corresponding to an underlying RFS protocol.

$$C_A = \begin{pmatrix} 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&0&0&1&0&1&0 \\ 0&0&0&0&0&1&0&1 \\ 0&0&0&1&1&0&1&0 \\ 0&0&1&0&0&1&0&1 \\ 0&1&1&0&1&0&0&1 \\ 1&1&0&1&0&1&1&0 \end{pmatrix}_{8\times 8} \quad C_B = \begin{pmatrix} 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&1&1&0 \\ 0&0&0&0&1&0&0&1 \\ 0&0&0&1&1&0&1&0 \\ 0&0&1&0&0&1&0&1 \\ 0&1&1&1&0&1&1&0 \\ 1&1&1&1&1&0&0&1 \end{pmatrix}_{8\times 8} \quad C_C = \begin{pmatrix} 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&1&1&0 \\ 0&0&0&0&1&0&0&1 \\ 0&0&0&1&0&1&0&1 \\ 0&0&1&0&1&0&1&0 \\ 0&1&0&1&0&1&1&0 \\ 1&1&1&0&1&0&0&1 \end{pmatrix}_{8\times 8} \tag{8}$$

Table 1 Testbed description

PC	CPU	Intel Pentium D 3 GHz	
	RAM	1 GB	
VIDEO	Codec	MPEG4-DIVX	
	Picture Size	652 × 272	
	FPS	23.98	
	Bitrate/channel	1217 Kbps	
nVoD	Channels	8	
	Segment Mapping	FB	RFS
		255 segments	1532 segments
		25.56 *sec/slot*	4.25 *sec/slot*

The different behaviours in Figs. 4 and 5 indicate that the coding matrix can be tuned to optimize the performance of the schema. Note that, even for $p_e = 0.2$, the percentage of correctly decoded segment packets comes theoretically close to 100% in just a few minutes.

3 Tests

3.1 Implementation of the schema

To test our schema in practice we developed a tool composed of two separate modules: a nVoD server and the corresponding client. Both include Simple Directmedia Layer (SDL) video players that display content with and without implicit error correction to allow subjective comparisons.

VoD multicasting carries data at several times the playback rate (over 1 Mbps for an acceptable quality) and the client has to operate at least as quickly. We therefore divided the server and the client into smaller processing units that execute simple operations. Figures 6 and 7 show the architectures of the server and the client, respectively. Figure 8 shows a screenshot of the working system. Compare the server player window (up, right) with the client player window after error correction

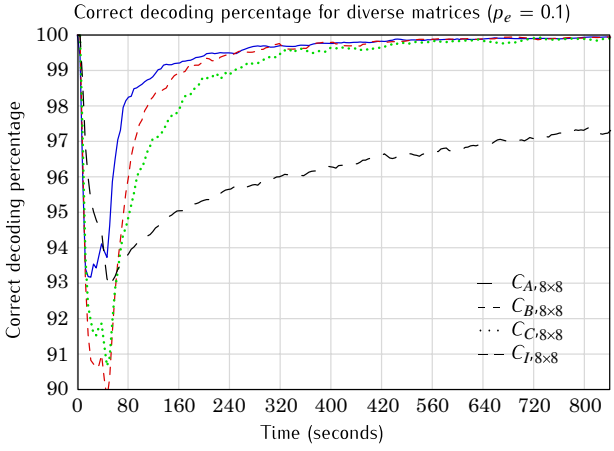

Fig. 12 Correct decoding percentage vs time for matrices C_A, C_B, C_C and without implicit error correction (C_I), for 1532 segments ($p_e = 0.1$)

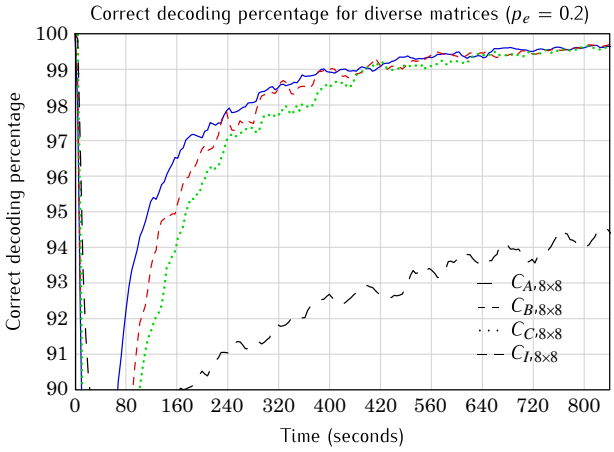

Fig. 13 Correct decoding percentage vs time for matrices C_A, C_B, C_C and without implicit error correction (C_I), for 1532 segments ($p_e = 0.2$)

(down). Figures 9, 10 and 11 show the PSNR with and without our error correcting schema, compared to the original file. The "missing points" in the curves of our schema correspond to full error correction.

3.2 Segment coding evaluation—simulation approach

The goal of the simulations was to study the behaviour of the schema in lossy environments. The results that follow were obtained by considering uniformly distributed errors, with error probabilities (p_e) of about 10% and 20%. Each frame corresponded to a UDP-IP packet, and loss events were also determined at frame level. It was guaranteed that the very first slot would never be affected by errors (in practice, it must be enforced that the first segment is correctly downloaded, and this

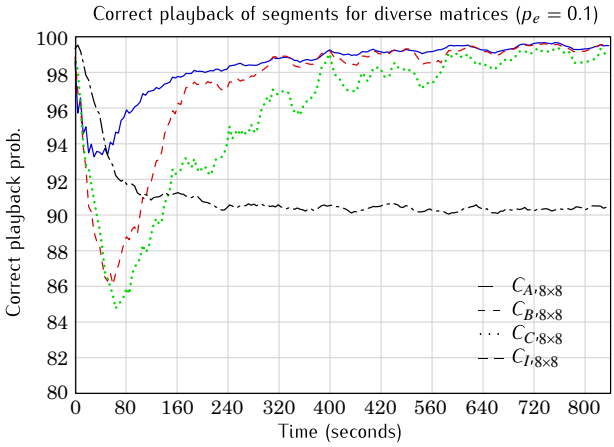

Fig. 14 Correct playback probability vs time for matrices C_A, C_B, C_C and without implicit error correction (C_I), for 1532 segments ($p_e = 0.1$)

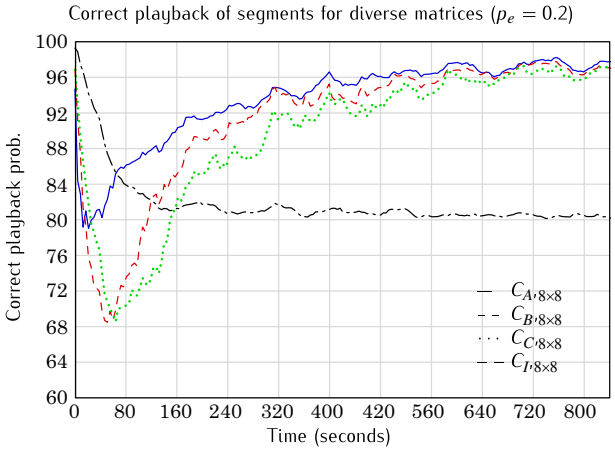

Fig. 15 Correct playback probability vs time for matrices C_A, C_B, C_C and without implicit error correction (C_I), for 1532 segments ($p_e = 0.2$)

can be considered a minor limitation of this schema). Subsequent slots were subject to error probability p_e (uniformly distributed errors).

Table 1 shows the simulation parameters. The system bandwidth is eight times the consumption rate of the content. We considered the three coding matrices in (8), chosen from previous tests with several matrices.

In the simulations, the three matrices in (8) exhibited good behaviour. After two or three slots, the percentage of correctly decoded segment packets grows exponentially, outperforming traditional nVoD schemas without error correction. The matrices differ in the convergence to the theoretical performance bound, as in the analysis with uniformly distributed errors. The percentage of correct decoding is below 100% due to limited matrix efficiency (in practice, a coding matrix does not always allow recovery from κ errors if κ segment packets are available from previous slots). The behavior of the coding matrices is coherent with the theoretical analysis.

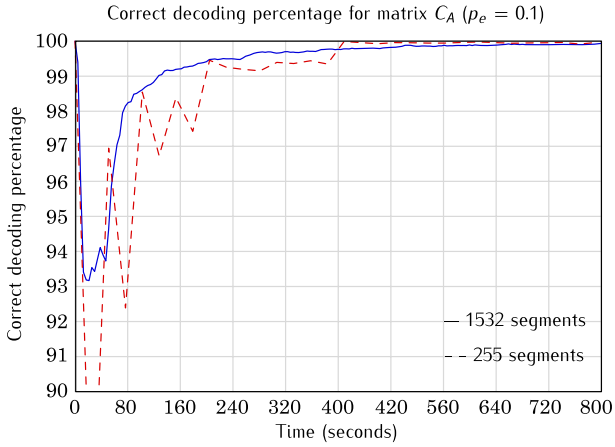

Fig. 16 Correct decoding percentage vs time from the moment the transmission starts, for different numbers of segments, employing C_A, $p_e = 0.1$

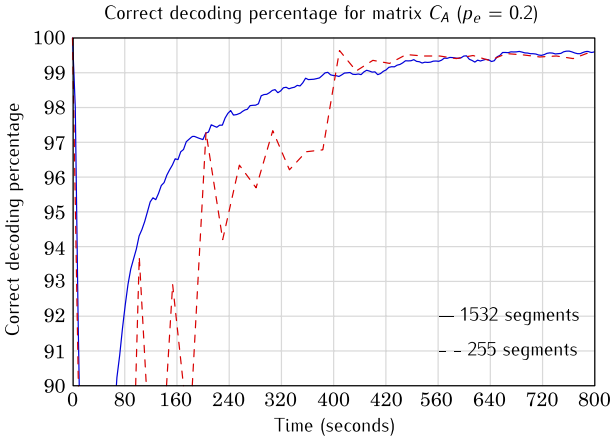

Fig. 17 Correct decoding percentage vs time from the moment the transmission starts, for different numbers of segments, employing C_A, $p_e = 0.2$

This appears if we compare Fig. 4 with Fig. 12, or if we compare Fig. 5 with Fig. 13. Therefore, expression (7) for uniform errors seems helpful in choosing a particular coding matrix from among different alternatives to be applied in scenarios with more realistic error distributions.

So far, we have measured performance in terms of correctly decoded segment packets (which includes redundancy), but subjective quality depends only on the percentage of correctly *played* packets as measured, in our case, by the VLC-based player itself. Figures 14 and 15 show that, after 80 s for $p_e = 0.1$ and after 7 min for $p_e = 0.2$, the playback is correct for 96% of the content, which is satisfactory in terms of subjective quality. However, RFS without implicit error correction attains correct playback probabilities of barely 90% and 80% for $p_e = 0.1$ and 0.2, respectively, for which subjectively satisfactory playback is unfeasible.

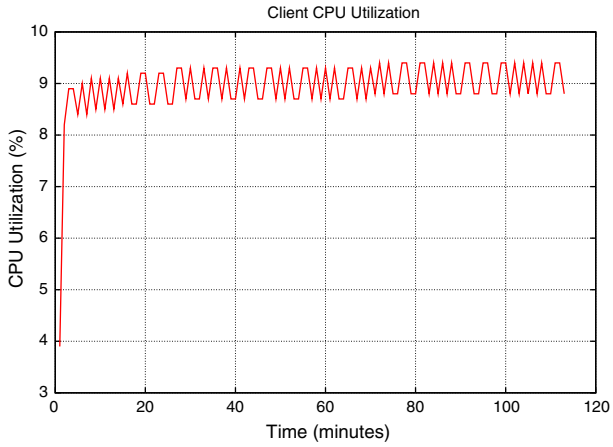

Fig. 18 Client CPU utilization

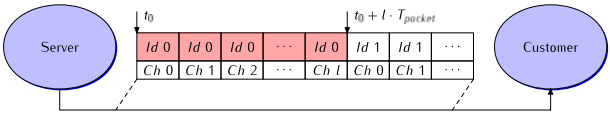

Fig. 19 Sequential packet transmission. Recovery fails completely for l consecutive lost frames. ($Id\ i$ is the frame identifier for each channel)

Figures 16 and 17 compare our approach for two different segment mappings, FB and RFS, with the settings in Table 1. The closer to the harmonic segment-allocation limit, the better the decoding results for a particular mapping. Thus, for 1532 segments with the RFS protocol, the schema performs better than with the FB protocol with a higher segment redundancy (255 segments). The oscillations in these figures, mainly with the FB protocol, are due to the fact that the rate of appearance of new segments in the slots is nonlinear, unlike in the ideal case.

Figure 18 shows the client CPU utilization during content download, using our schema. As it can be observed, after a start-up time, total CPU usage is low and it never exceeds 10%.

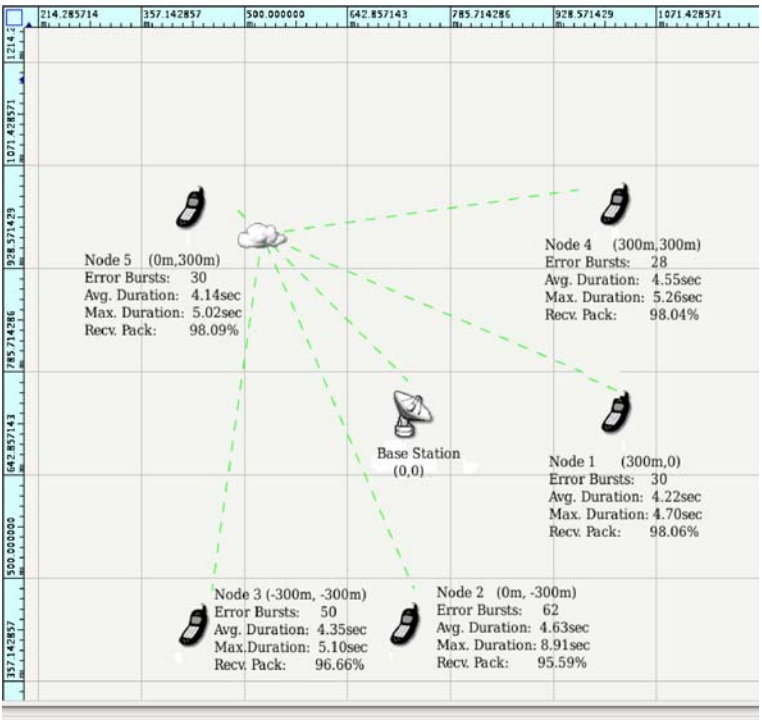

Fig. 20 Node placement and statistics for received frames in WiMax simulation

Table 2 Qualnet configuration parameters

ANTENNA TYPE	OMNIDIRECTIONAL
ANTENNA GAIN	10.5 dB
ANTENNA EFFICIENCY	0.8
PATHLOSS MODEL	TWO-RAY
PHY-NOISE-FACTOR	10
PROPAGATION CHANNEL	5.8 GHz
TX-POWER	36.5 dBm
ROUTING PROTOCOL	OSPFv2
MULTICAST PROTOCOL	MOSPF
WEATHER PATTERN	1.5 mm/h

3.3 Real scenario: WiMax testbed

WiMax is a promising access alternative in certain scenarios. Substandard 802.16 m enables WiMax to achieve 100 Mbps and thus opens the door to multimedia applications such as VoD [19].

In wireless networks, communications between the base station (BS) and the subscribers (SS) may be seriously affected by rain, so that even with careful planning there may be error bursts in some locations that may not be properly handled by the lower layers. In other words, the uniform distribution assumption so far does not hold. In this case, the benefit of implicit error correction is minimal. A burst starting at t_0 for longer than $t_0 + l \cdot T_{packet}$ leads to insufficient information for the decoder and therefore null segment recovery (Fig. 19).

Due to the bursty WiMax error patterns, even with a low error rate, the probability of losing several consecutive packets in different channels is high, which affects our schema. At this point, a further enhancement (also without extra bandwidth overhead) is possible: to interleave frames independently of the coding process, to uniformize errors. This introduces a certain amount of extra playback latency at the beginning of a VoD session, yet this latency is bounded regardless of the number of users in the system. Interleaving time T_I must be carefully selected taking into account the maximum and average number of consecutive lost frames. A long T_I increases the waiting time for the first slot $T_1 = max(T_I * \lceil \frac{T_{slot}}{T_I} \rceil, T_I)$.

For the WiMax tests we employed the Qualnet 4.5 simulator. Figure 20 shows the simulation scenario and Table 2 the Qualnet settings. The parameters of the nVoD protocol are the same as in previous simulations. The BS acts as the multicast server and the SS nodes are the end customers. The terrain model includes a weather

Table 3 Sequential transmission

Segments with errors at the client side

Node	Without interleaving	
	Traditional nVoD	Implicit Error Correction
	Segments with errors	Segments with errors
1	54	54
2	109	109
3	92	92
4	51	51
5	62	62

Table 4 Transmission with interleaving

Node	Interleaving	
	Traditional nVoD	Implicit Error Correction
	Segments with errors	Segments with errors
1	185	5
2	374	16
3	318	7
4	166	6
5	195	6

Segments with errors at the client side

pattern that covers the scenario, moving across from west to east. The interleaving time T_I was set to 22.66 s.

We performed tests for transmissions with and without interleaving, both for a traditional nVoD approach and for our approach with implicit error correction (using the C_A coding matrix, as suggested by the analysis in the previous section) for an underlying RFS protocol with 1532 segments and 4.25 s/slot. In those tests, we measured the performance of the system in terms of correct playback of segments (VLC-based player statistics) at the client side. Tables 3 and 4 present the results of these experiments. As we can see in the histogram of Fig. 21, traditional nVoD

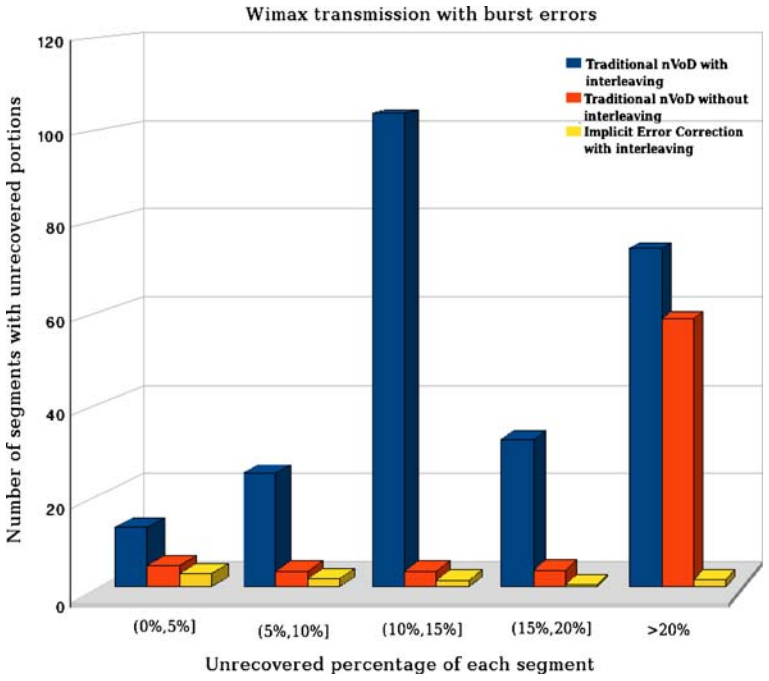

Fig. 21 Histogram of segments displayed with errors. Each group of three columns indicates the results of the WiMax test for three nVoD schemas. The two leftmost columns in every case represent nVoD without implicit error correction with and without interleaving, respectively. The third column corresponds to the schema with interleaving and implicit error correction. The horizontal axis represents the percentage of unrecovered packets within each segment and the vertical axis indicates the number of segments subject to that error level

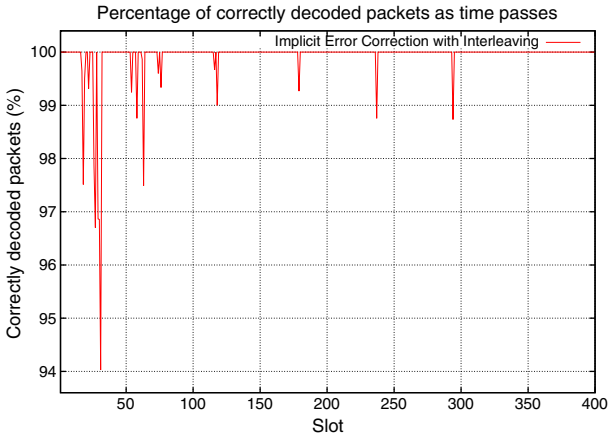

Fig. 22 Percentage of correctly decoded packets as time passes for implicit error correction with interleaving. After slot #400, 100% of the packets are correctly decoded

without interleaving fails completely if the probability of segment packet errors exceeds 20%. The peak in the histogram for the case without error correction and with interleaving, for unrecovered segment portions of 10–15% of total segment length, simply indicates that, in that case, the interleaving process "spreads" a noticeable part of the error (10–15%) among many segments. The only purpose of that case is to illustrate how interleaving uniformizes the error bursts, since there is no error correction at all. According to Table 3, our approach does not work well without interleaving, and implicit error correction is of little use, as expected. Nevertheless, our nVoD schema with implicit error correction combined with interleaving (the third vertical bar in Fig. 21), has a substantial impact on system performance in all error ranges. Figure 22 shows the percentage of correctly decoded packets as time passes. It is straightforward to see the benefits of combining interleaving with implicit error correction.

4 Conclusions and future work

We have presented a proposal for implicit error correction using nVoD protocols and described its benefits. The schema is compatible with any underlying nVoD protocol with equally sized segments. In our tests, the schema allows the decoding of over 99% of segments for a continuous 10% error rate (uniform distribution) after 80 seconds, and matches this performance for a 20% error rate after seven minutes. It inherits all the advantages of traditional nVoD server-initiated schemas, but greatly reduces errors at the client side without extra bandwidth cost. Theoretical and practical results show that coding efficiency and segment mapping are crucial for a fast convergence of the system to its best performance. In this regard, we provide an analytical study that helps in the selection of good coding matrices.

Another point to consider is that wireless channels are subject to frequent unexpected burst errors, due to temporal fading among other factors. We show that, by combining our approach with packet interleaving, it is even possible to recover from a packet error probability of 20% without an increase in bandwidth, contrasting with previous approaches in which bandwidth increases with the error rate.

As future work we plan to extend our schema to other sources such as scalable video, to support heterogeneous clients, and to other networks where packet losses can reach high rates.

Acknowledgements The work described in this paper has been supported by grants MIND-GAP-5 PGIDIT 08TIC010CT (*Xunta de Galicia*, Spain) and CON-PARTE-2 TEC2007-67966-C03-02 (*Ministerio de Educación y Ciencia*, Spain).

References

1. Asorey-Cacheda R, González-Castaño FJ (2008) A multicast nVoD schema with zero-overhead implicit error correction. In: Proc. IEEE intl. communications conference 2008, 19–23 May 2008, pp 2017–2020
2. Azad SA, Murshed M, Dooley LS (2003) A novel batched multicast patching scheme for video broadcasting with low user delay. In: Proc. 3rd IEEE international symposium on signal processing and information technology (ISSPIT), pp 339–342
3. Eager D, Vernon M, Zahorjan J (1999) Bandwidth skimming: a technique for cost-effective video-on-demand. Tech. Rep. CS-TR-1999-1408, Computer Science Department, Univ. of Wisconsin-Madison
4. Hua KA, Cai Y, Sheu S (1998) Patching: a multicast technique for true video-on-demand services. In: Proc. 6th ACM international conference on multimedia, pp 191–200
5. Janakiraman R, Waldvogel M, Xu L (2002) Fuzzycast: efficient video-on-demand over multicast. In: Proc. Infocom 2002, pp 920–929
6. Juhn L, Tseng L (1997) Harmonic broadcasting for video-on-demand service. IEEE Trans Broadcast 43(3):268–271
7. Juhn L, Tseng L (1997) Fast broadcasting for hot video access. In: Proc. 4th international workshop on real-time computing systems and applications, 27–29 October 1997, pp 237–243
8. Ma H, Shin KG (2002) Multicast video-on-demand services. ACM Comput Commun Rev 32(1):31–43
9. Ma H, Shin KG, Wu W (2005) Best-effort patching for multicast true VoD service. Multimedia Tools and Applications 26(1):101–122
10. Mahanti A, Eager D, Vernon MK, Sundaram-Stukel DJ (2003) Scalable on-demand media streaming with packet loss recovery. IEEE/ACM Trans Netw 11(2):195–209
11. Paris JF (1999) A simple low-bandwidth broadcasting protocol for video-on-demand. In: Proc. 8th intl. conf. on computer communications and networks, pp 690–697
12. Paris JF (2001) A fixed-delay broadcasting protocol for video-on-demand. In: Proc. 10th intl. conf. on computer communications and networks, pp 418–423
13. Paris JF (2005) A simple but efficient broadcasting protocol for video-on-demand. In: Proc. 24th IEEE intl. performance, computing, and communications conference, 7–9 April 2005, pp 167–174
14. Peng C, Shen H, Xiong N, Yang LT (2006) Discrete broadcasting protocols for video-on-demand. Lect Notes Comput Sci 4208:642–652
15. Reisslein M, Saparilla D, Ross KW (2004) Periodic broadcasting with VBR-encoded video. Multimedia Systems 9, pp 503–516
16. Sujatha DN, Girish K, Venugopal KR, Patnaik LM (2007) An integrated quality-of-service model for video-on-demand application. IAENG Int J Comput Sci 34(1):1–10
17. Sun Y, Kameda T (2005) Harmonic block windows scheduling through harmonic windows scheduling. Multimed Inf Syst 3665:190–206
18. Tseng Y-C, Yang M-H, Chang C-H (2002) A recursive frequency-splitting scheme for broadcasting hot videos in VOD service. IEEE Trans Commun 50(8):1348–1355
19. Xie F, Hua KA, Jiang N (2007) Achieving true video-on-demand service in multi-hop wimax mesh networks. In: Proc. 32nd IEEE conference on local computer networks, 15–18 October 2007, pp 287–294
20. Yu H-F (2008) Hybrid broadcasting with small buffer demand and waiting time for video-on-demand applications. IEEE Trans Broadcast 54(2):304–311
21. Yu H-F, Yang H-C, Tseng L-M (2007) Reverse fast broadcasting (RFB) for video-on-demand applications. IEEE Trans Broadcast 53(1):103–111

22. Yu H-F, Chen Y-N, Yang H-C, Yang Z-Y, Tseng L-M (2008) An efficient scheme for broadcasting popular videos at low buffer demand. Comput Commun 31(10):2270–2279
23. Yu H-F, Yang H-C, Wang Y-T, Fan P-L, Chien C-Y (2009) Broadcasting scheme with low client buffers and bandwidths for video-on-demand applications. Multimedia Tools and Applications 42(3):295–316
24. Yu H-F, Ho P-H, Yang H-C (2009) Generalized sequence-based and reverse sequence-based models for broadcasting hot videos. IEEE Trans Multimedia 11(1):152–165

Francisco J. González-Castaño received the Ingeniero de Telecomunicación degree from University of Santiago de Compostela, Spain, in 1990 and the Doctor Ingeniero de Telecomunicación degree from University of Vigo, Spain, in 1998. He is currently a Catedrático de Universidad (Full Professor) with the Telematics Engineering Department, University of Vigo, where he leads the Information Technologies Group (http://www-gti.det.uvigo.es). He is also currently with Gradiant, Spain, as the Network Research Director. He has authored over 50 papers published in international journals, in the fields of telecommunications and computer science, and has participated in several relevant national and international projects. He is the holder of three Spanish patents, one European patent, and one U.S. patent.

Rafael Asorey-Cacheda was born in Vigo, Spain in 1977. He was a researcher with the Information Technologies Group, University of Vigo, Spain until 2008. He is currently with Optare Solutions, Spain, as R&D Director. His interests include content distribution, high-performance switching, video transcoding, peer-to-peer networking and wireless networks.

Héctor Cerezo-Costas received the Ingeniero de Telecomunicación degree from University of Vigo, Spain, in 2007. He is currently a researcher with the Information Technologies Group, University of Vigo, Spain. His interests include content distribution and wireless networks.

Juan C. Burguillo-Rial received the M.Sc. degree in 1995 and the Ph.D. degree (cum laude) in 2001 from the University of Vigo, Spain; both in Telecommunication Engineering. He is currently an associate professor with the Telematics Engineering Department at the same university. He has participated in several R&D projects in the areas of Telecommunications and Software Engineering, and has published over one hundred papers in international refereed journals and conference proceedings. His research interests include autonomous agents, distributed artificial intelligence, distributed optimization and telematic services.

Felipe J. Gil-Castiñeira received the M.Sc. degree in telecommunication engineering (major in Telematics) and the Ph.D. degree in telecommunication engineering from the University of Vigo, Vigo, Spain, in 2002 and 2007, respectively. He is currently an Assistant Professor with the Department of Telematics Engineering, University of Vigo. His research interests include wireless and car-to-car communication technologies, embedded systems, nomadic devices, wireless sensor networks, and ubiquitous computing.