IST R&D. FP6-Priority 2.

SPECIFIC TARGETED RESEARCH PROJECT

Project Deliverable

SUIT Doc Number	SUIT_428
Project Number	IST-4-028042
Project Acronym+Title	SUIT- Scalable, Ultra-fast and Interoperable Interactive Television
Deliverable Nature	Report
Deliverable Number	D3.4
Contractual Delivery Date	November 30, 2007
Actual Delivery Date	November 30, 2007
Title of Deliverable	Optimization of the entire video coding system
Contributing Workpackage	WP3
Project Starting Date; Duration	01/02/2006; 27 months
Dissemination Level	PU
Author(s)	Maryse Stoufs (IBBT), Adrian Munteanu (IBBT), Dirk Bakker (IBBT), Stewart Worrall (UoS), Zaheer Ahmad (UoS), Juan Carlos Plaza (UPM), Julian Cabrera (UPM), Daniel Baptista (IT), Antonio Navarro (IT), Yves Dhondt (IBBT)

Abstract

In this deliverable, we experimentally assess the performance of the joint source-channel coding (JSCC) video coding architecture proposed in D.3.3 over packet loss channels using SVC, the scalable extension of H.264/MPEG-4 AVC. We show that our previously developed JSCC-methodology (cf. SUIT_375) delivers competitive results against state-of-the-art Lagrangian-based JSCC-algorithms. Moreover, compared to the state-of-the-art, we demonstrate that our approach significantly reduces the number of computations needed to derive the rate-distortion hulls and that the proposed approach constructs convex rate-distortion hulls for each frame, irrespective of the target rate. This allows the pre-computation of the convex rate-distortion hulls for typical packet loss channels, such that the extraction of a near-optimal JSCC-allocation can be achieved on the fly for any target rate or packet-loss rate. We conclude that the proposed JSCC-methodology provides optimized resilience against transmission errors in scalable video streaming over variable-bandwidth error-prone channels.

We also experimentally assess the performance of the multiple description coding scheme based on embedded multiple description scalar quantization (MDC2) as described in D.3.1 and D.3.2. In order to simulate the transmission of the descriptions over various channels with different available bandwidths and packet loss parameters, we propose an extractor that truncates the descriptions towards a target rate. The MDC2 architecture is a viable solution for performing robust transmission over channels introducing up to 20% of losses.

This also deliverable describes a new multiple description architecture (MDC-3) with balanced base layer.

Besides, this deliverable also presents experimental results obtained for simulated transmission of scalable video over WiMAX using an Unequal Power Allocation (UPA) scheme. The results show that the UPA scheme discussed in deliverable D3.3 is capable of improving received video quality. In this deliverable, an unequal power allocation simulator is also presented. The simulator can be used to demonstrate the capability of the UPA scheme over WiMAX, or to run simulations to find the effects of various encoding schemes and channel conditions on the received video quality.

Finally, this deliverable presents a rate-control module for the gateway that aims at optimizing the quality of the video sequences received at the terminal. The rate-control strategy relies on NALU retransmissions as the mechanism to protect the SVC transmission. It is based on a discrete wireless channel model that captures the stochastic behaviour of the channel, and a distortion model customized for SVC streams. Results have shown that this rate control strategy results in an increase in the number of decodable frames, thus decreasing the distortion at the decoder.

Keyword list: joint source-channel coding (JSCC), multiple description coding (MDC), scalable video coding (SVC), unequal error protection, H.264/MPEG-4, Error Resilience.

Optimization of the Entire Video Coding System

SUIT_428

30-11-2007

Table of Contents

1 Introduction

1.1 Scope

This document is part of WP3.

In this document the performance of the joint source-channel coding (JSCC) methodology proposed in D3.3 is assessed. This JSCC-methodology was proposed in conjunction with the scalable MDC-scheme described in deliverables D3.1 and D3.2 (Design of scalable MD-SVC) to realise efficient error resilient coding of the video stream. Specifically, the proposed JSCC achieves optimal network resource allocation by jointly optimizing the source rate allocated to encode the source video with a scalable video coder (SVC) and the channel rate allocated to forward error correction coding (FEC). The JSCC-approach represents an alternative (and complementary) solution to the scalable MD-SVC system developed in D3.1 and D3.2.

Additionally, the performance of the MDC-scheme with embedded multiple description scalar quantization (EMDSQ) as proposed in deliverables D3.1 and D3.2 is assessed and a third MDC system, named MDC-3, is discussed and evaluated.

Also, the unequal power allocation (UPA) scheme introduced in deliverable D3.3 is further evaluated by means of a simulator and some experiments.

Finally, a rate control system for the gateway, which will protect the streamed video in the last mile WLAN network, is introduced and discussed.

Contributors to this deliverable are IBBT, UoS and UPM.

1.2 Objectives

The objective of this deliverable is to optimize the entire video coding system of SUIT. The optimization concentrates mainly on adding error resilience functionality to the video sub system. To do so, three different approaches are taken:

- a joint-source channel coding approach;
- an unequal power allocation approach for the wireless links;
- a rate control mechanism approach for the playout

The JSCC algorithm should decide on the best source rate to encode the video taken into account both video source and network source characteristics.

The unequal power allocation algorithm should allow for selecting how much power should be given for each layer to optimize the overall quality for all users.

The rate control mechanism should do two things. Firstly, decide on which SVC packets should be send multiple times to coop with expected channel losses. Secondly, it should decide on how many times the packets should actually be send.

2 Performance Assessment of the Scalable Video Coding Systems

2.1 Optimized Scalable Joint Source-Channel Coding

2.1.1 Experimental Results

For our experiments, we use the Joint Scalable Video Model (JSVM) reference software version 7.10 in scalable coding mode. The rate-distortion points of the embedded sources are obtained once for each sequence by using the quality level assigner [1] and stored as look-up tables. The source rate-distortion points are linearly interpolated to compute the convex hulls.

The first 50 frames of the test sequences of “Bus”, “Football”, “Mobile” and “Foreman” at 30 Hz are used in all our experiments. For our simulations we perform scalable source coding with one resolution layer and three FGS quality enhancement layers for which we choose context-adaptive binary arithmetic coding as entropy coding method. The base layer quantization parameter is set to 40. We set the GOP size to 8, the intra period to 16 and the search range to 32. The source coding of each video sequence is performed only once and the same bitstream is used to extract the target rates for the different experiments.

In all the experiments, we do not consider the (small) amount of rate used by the first SEI (supplemental enhancement information)-message, the sequence parameter sets and the picture parameter sets and assume that this information reaches the decoder intact. For all the considered sequences, this information requires about 400 bytes for the given encoding settings.

2.1.1.1 Lossless transmission

In the following experiments, we apply our algorithm in the situation where no packet losses are incurred. When the transmission channel is lossless (packet loss ε equal to 0%), the extracted bitstream will contain no channel protection. Our proposed method is then reduced to a Lagrangian-based source rate-allocation method which can be used as a bitstream extraction mechanism to achieve a target bit rate R_{tot} for a given resolution and frame-rate.

The JSVM reference software [2] currently provides two approaches to perform such a bitstream extraction. The first approach makes use of a simple truncation of the progressive refinement NAL units. In this approach, the bitstream extractor first parses the encoded stream and determines the size of each NAL-unit. Then it evaluates how many FGS layers can be sent to achieve the required resolution and frame-rate at the given target rate R_{tot}. The last FGS-layer considered is finally truncated to meet the target rate R_{tot} [2]. In the second approach, a quality level assigner (QLA) [2] introduces quality levels at various rate points such that an optimized rate allocation similar to JPEG2000’s EBCOT (Embedded Block Coding with Optimized Truncation) [3] is achieved. This quality level information can be embedded in the NAL-unit header. In this way, no additional rate is spent on the quality level information; we refer to section 2.5.2.2 of [2] for details regarding the calculation of these quality levels.

We compare the performance of our proposed method with both approaches for bitstream extraction present in the JSVM reference software. We use the SVC-encoded sequences of Bus, Football, Foreman and Mobile as described previously. From Table 1, it can be seen that compared to the standard bitstream extractor [2] our allocation using packets of 10 bytes achieves slightly better average PSNR results at 400 and 800 kbps. At 1200 kbps, although we observe a slightly better average PSNR for the standard extractor, we also observe a rate difference, with the actual rates of the proposed JSCC being slightly beneath the required target rate, while the rates of the standard extractor being systematically above the target.

For a codeword size of 256 bytes our rate allocation algorithm performs very close to QLA; by reducing the codeword size to 10 bytes, PSNR results on par with those of QLA can be achieved. One notes that reducing the codeword size results in more truncation points on the convex hulls used in the Lagrangian optimization, and hence, in a slightly improved performance. One concludes that in lossless transmission the proposed allocation yields similar performance compared to the standard H.264 SVC rate-allocation techniques.

Table 1: Performance comparison at different target rates of (a) the standard bitstream extractor [2], (b) the standard bitstream extractor with quality layers computed using the quality level assigner (QLA) [1], and (c) our proposed JSCC-method with codeword sizes of 10 and 256 bytes

(a) MOBILE

Target rate (kbps)	Rate Met (kbps)	Y (dB)	U (dB)	V(dB)	Avg PSNR (dB)
Bitstream Extractor					
400	398.57	27.65	33.69	32.96	**29.54**
800	798.84	30.21	36.22	35.52	**32.09**
1200	1201.11	31.24	37.06	36.28	**33.05**
Bitstream Extractor with QLA					
400	400.45	27.68	33.66	32.96	**29.56**
800	800.84	30.22	36.22	35.51	**32.10**
1200	1201.04	31.67	36.26	35.79	**33.12**
Proposed JSCC (packets of 256 bytes)					
400	400.59	27.62	33.59	32.90	**29.50**
800	799.95	30.17	36.20	35.49	**32.06**
1200	1200.54	31.67	36.27	35.77	**33.12**
Proposed JSCC (packets of 10 bytes)					
400	399.98	27.69	33.62	32.94	**29.55**
800	799.97	30.22	36.21	35.51	**32.10**
1200	1200.00	31.67	36.26	35.78	**33.12**

(b) FOREMAN

Target rate (kbps)	Rate Met (kbps)	Y (dB)	U (dB)	V(dB)	Avg PSNR (dB)
Bitstream Extractor					
400	401.13	34.82	40.96	43.82	**37.34**
800	801.53	37.13	42.71	45.57	**39.47**
1200	1201.53	38.40	43.90	46.67	**40.69**
Bitstream Extractor with QLA					
400	400.95	35.07	40.63	43.36	**37.38**
800	801.24	37.28	42.64	45.49	**39.54**
1200	1201.43	38.76	42.76	45.76	**40.59**
Proposed JSCC (packets of 256 bytes)					
400	400.59	35.04	40.64	43.38	**37.36**
800	799.95	37.25	42.63	45.46	**39.51**
1200	1199.31	38.75	42.77	45.78	**40.59**
Proposed JSCC (packets of 10 bytes)					
400	399.98	35.07	40.62	43.36	**37.38**
800	799.97	37.28	42.64	45.49	**39.54**
1200	1200.00	38.76	42.75	45.75	**40.59**

(c) BUS

Target rate (kbps)	Rate Met (kbps)	Y (dB)	U (dB)	V(dB)	Avg PSNR (dB)
Bitstream Extractor					
400	400.74	28.71	38.18	39.32	**32.05**
800	801.12	31.39	39.67	41.38	**34.43**
1200	1201.13	33.05	41.18	42.59	**36.00**
Bitstream Extractor with QLA					
400	400.55	28.81	37.99	39.10	**32.06**
800	800.96	31.52	39.49	41.07	**34.44**
1200	1201.05	33.19	40.41	41.95	**35.86**
Proposed JSCC (packets of 256 bytes)					
400	399.36	28.77	37.88	38.89	**31.98**
800	799.95	31.49	39.49	41.08	**34.42**
1200	1199.31	33.19	40.38	41.96	**35.85**
Proposed JSCC (packets of 10 bytes)					
400	399.98	28.81	37.99	39.10	**32.06**
800	799.97	31.52	39.49	41.07	**34.44**
1200	1200.00	33.19	40.41	41.96	**35.86**

(d) FOOTBALL

Target rate (kbps)	Rate Met (kbps)	Y (dB)	U (dB)	V(dB)	Avg PSNR (dB)
Bitstream Extractor					
800	800.72	29.72	36.06	38.30	**32.21**
1200	1201.12	31.38	37.42	39.44	**33.73**
1600	1601.10	32.90	38.25	40.23	**35.01**
Bitstream Extractor with QLA					
800	800.75	29.83	36.12	38.01	**32.24**
1200	1200.84	31.51	37.59	39.36	**33.83**
1600	1600.93	33.01	38.15	39.66	**34.97**
Proposed JSCC (packets of 256 bytes)					
800	799.95	29.82	36.10	38.01	**32.23**
1200	1199.31	31.48	37.54	39.34	**33.80**
1600	1599.90	33.02	38.12	39.65	**34.97**
Proposed JSCC (packets of 10 bytes)					
800	799.97	29.84	36.12	38.01	**32.25**
1200	1200.00	31.50	37.59	39.35	**33.82**
1600	1599.98	33.01	38.15	39.66	**34.97**

2.1.1.2 Lossy transmission

In this section, we compare the performance of our developed methodology with our extension towards SVC of one of the latest near-to-optimal JSCC-methodologies for scalable source codes, which was presented in [4] and which provides state-of-the-art results in JSCC of images encoded using the JPEG-2000 standard.

In [4], a Lagrangian-based optimization is proposed which uses backward dynamic programming to determine the source and channel allocation. The algorithm meets the target rate R_{tot} by first choosing a slope λ for the rate-distortion Lagrangian optimization and constructing convex hulls for each codeblock, with this slope as a constraint in a cost-to-go function. The cost function at stage k, state r_k is defined in [4] as:

$$g_k(r_k) = \left[1 - P\left(r_k, \frac{l_k}{r_k}\right)\right](-d_k) + \lambda \frac{l_k}{r_k}, \tag{7}$$

where stage k corresponds to the coding pass and state r_k belongs to the set of available channel code rates at stage k denoted as $\square(k)$. Also, $\frac{l_k}{r_k}$, $P\left(r_k, \frac{l_k}{r_k}\right)$ and d_k denote the length of the coding pass including parity, the probability that there are one or more uncorrected errors after channel decoding for coding pass k and the distortion reduction of including coding pass k respectively. For the given slope λ, the approach then starts from the last possible coding pass that can be included in the codestream and proceeds backwards to the first coding pass. For each coding pass, it computes the optimal cost-to-go $J_k(r_k)$ for each available channel code rate r_k. Let M_l be the total number of codewords. The cost-to-go function is defined as:

$$J_{M_l}(r_{M_l}) = g_{M_l}(r_{M_l}), \text{ when } (k = M_l)$$

$$J_k(r_k) = g_k(r_k) + \min_{r_{k+1}\in\square(k+1)} \left\{ \left[1 - P\left(r_k, \frac{l_k}{r_k}\right)\right] \times \left[J_{k+1}(r_{k+1}) - \lambda\left(\frac{l_{k+1}}{r_{k+1}} + \sum_{p=k+2}^{M_l} \frac{l_p}{r_p^*}\right)\right] + \lambda\left(\frac{l_{k+1}}{r_{k+1}} + \sum_{p=k+2}^{M_l} \frac{l_p}{r_p^*}\right)\right\} \quad (8)$$

where $r_p^* = \begin{cases} \arg\min\limits_{r_{k+2}\in\square(k+2)} J_{k+1}(r_{k+1}), \text{ when } p = k+2 \\ \arg\min\limits_{r_p\in\square(p)} J_{p-1}(r_{p-1}^*), \text{ when } \mathrm{k}+3 \le p \le M_l \end{cases}$.

The minimal cost associated with a state in the first stage then becomes the optimal cost, and the path which leads to this state from the last stage determines the optimal source and channel rate allocation for this codeblock [4]. Finally, a higher level bi-section method is used to identify whether the chosen slope meets the rate or not. If not, the algorithm iterates, adapts the slope and re-computes new convex hulls for the new adapted slope.

We extend this algorithm towards SVC by constructing convex hulls for each frame of the SVC-encoded sequence instead of each codeblock of the JPEG2000-encoded image, proposed originally in [4]. Exactly as in [4], for a certain slope λ we construct for each frame (instead of for each codeblock) the convex hulls backwards by computing the cost-to-go functions for each transmitted codeword. Similar to our developed method, the base layer is protected with the highest protection level.

For channel coding, we employ punctured regular (3,6)-LDPC codes constructed similarly to [5] and produce codewords of exactly 256 bytes. Iterative LDPC-decoding is allowed up to 100 iterations. The performance of these codes was measured off-line (see Table 1).

The LDPC-codewords of 256 bytes are interleaved with a row-column bit interleaver of 2048 by 512 bits. We consider the transmission of the interleaved packets of 512 bits over packet loss channels with 5 and 10% of packet losses and several target bit-rates. As we consider probabilistic packet loss channels, we repeat the transmission of the interleaved codewords 300 times and average the results.

Table 2: Punctured LDPC-codes used for ε = 5% and ε = 10% packet loss channels. K: source bytes, M: parity bytes, P: punctured bytes, $p_f(\varepsilon)$: probability of loosing a codeword when transmitted over a packet loss channel with parameter ε.

$\varepsilon =$ 5 % losses				$\varepsilon =$ 10 % losses			
K (bytes)	**M (bytes)**	**P (bytes)**	**$p_f(\varepsilon)$ (%)**	**K (bytes)**	**M (bytes)**	**P (bytes)**	**$p_f(\varepsilon)$ (%)**
205	51	154	0.00E+00	194	62	132	0.00E+00
209	47	162	1.00E-06	198	58	140	4.84E-05
211	45	166	2.83E-04	200	56	144	1.21E-03
213	43	170	3.24E-02	202	54	148	3.63E-02
215	41	174	1.83E-01	204	52	152	1.49E-01

To measure the performance, we first compute the mean square error over the whole sequence in Y, U and V for each transmission as: $PSNR_X = 10\log_{10}(255^2 / MSE_X)$, with $X = Y,U,V$. These PSNR values are then averaged over all transmissions resulting in $\overline{PSNR_Y}$, $\overline{PSNR_U}$ and $\overline{PSNR_V}$. The global average is finally found as: $\overline{PSNR_{YUV,avg}} = (4 \times \overline{PSNR_Y} + \overline{PSNR_U} + \overline{PSNR_V})/6$.

Table 3(a-d) show the results of our proposed JSCC-method compared to our SVC extension of [4] with equal error protection (EEP) using the highest protection level and unequal error protection (UEP) with 5 codes for transmission over channels with 5 and 10% packet losses. The experiments show that our algorithm and our SVC extension of [4] produce very similar results. The results also show that with the considered protection codes 0.1dB can be gained by use of UEP. This confirms the benefits of UEP, similar to the observations of [4] in case of images.

In Table 4, we compare the average PSNR-results obtained by transmitting unprotected SVC-encoded streams over lossless channels of 1500 kbps with the average PSNR-results obtained when transmitting the protected streams using the proposed JSCC over channels of 1500kbps with 5% and 10% losses. We must observe that unprotected SVC transmission over error-prone channels would have catastrophic effects in the output quality. On the contrary, for the proposed JSCC the PSNR-drop caused by channel errors is limited to less than 1.5 dB. These results demonstrate the robustness of the proposed JSCC-approach against transmission errors even in adverse conditions, such as 10% channel losses.

Table 3 (a-d): Performance comparison of the proposed JSCC-method and the extension towards SVC of the algorithm of [4] for transmission over 5% and 10% packet loss channels.

Table 3(a): Bus

		Total target rate (kbps)	**Total rate met (kbps)**	**Source rate (kbps)**	**Y (dB)**	**U (dB)**	**V (dB)**	**PSNR avg (dB)**
		5 % packet losses						
BUS 50 frames EEP 1 code - strongest	**Proposed JSCC**	***500***	498.9	399.5	28.78	37.98	39.02	**32.02**
		1000	999.0	800.0	31.50	39.49	41.07	**34.42**
		1500	1499.1	1200.5	33.19	40.40	41.99	**35.86**
	Extended JSCC of [4]	***500***	498.9	399.5	28.78	38.00	39.06	**32.03**
		1000	999.0	800.0	31.49	39.48	41.03	**34.41**
		1500	1499.1	1200.5	33.21	40.52	42.06	**35.91**
BUS 50 frames 5 codes	**Proposed JSCC**	***500***	500.1	402.1	28.80	37.96	38.98	**32.02**
		1000	1000.2	811.9	31.54	39.49	41.06	**34.45**
		1500	1500.4	1223.9	33.31	40.42	42.00	**35.95**
	Extended JSCC of [4]	***500***	498.9	400.9	28.79	38.00	39.06	**32.04**
		1000	1000.2	813.0	31.54	39.47	41.02	**34.44**
		1500	1500.4	1225.2	33.29	40.62	42.15	**35.99**
		10 % packet losses						
BUS 50 frames EEP 1 code - strongest	**Proposed JSCC**	***500***	498.9	378.1	28.60	37.79	38.80	**31.83**
		1000	999.0	757.1	31.32	39.48	41.06	**34.30**
		1500	1499.1	1136.1	32.91	40.13	41.65	**35.57**
	Extended JSCC of [4]	***500***	498.9	378.1	28.60	37.83	38.83	**31.84**
		1000	999.0	757.1	31.29	39.47	41.00	**34.27**
		1500	1499.1	1136.1	32.93	40.13	41.66	**35.59**
BUS 50 frames 5 codes	**Proposed JSCC**	***500***	500.1	379.9	28.61	37.81	38.82	**31.85**
		1000	1000.2	767.5	31.36	39.47	41.05	**34.33**
		1500	1499.1	1155.1	32.99	40.18	41.71	**35.64**
	Extended JSCC of [4]	***500***	500.1	379.7	28.62	37.85	38.86	**31.86**
		1000	1000.2	767.8	31.35	39.47	41.02	**34.32**
		1500	1500.4	1158.6	33.04	40.28	41.84	**35.72**

Table 3 (b): Football

		Total target rate (kbps)	Total rate met (kbps)	Source rate (kbps)	Y (dB)	U (dB)	V (dB)	PSNR avg (dB)
		5 % packet losses						
FOOTBALL 50 frames EEP 1 code - strongest	Proposed JSCC	1000	999.0	800.0	29.83	36.11	38.02	**32.24**
		1500	1499.1	1200.5	31.49	37.54	39.34	**33.80**
		2000	1999.3	1601.0	33.02	38.12	39.65	**34.98**
	Extended JSCC of [4]	1000	999.0	800.0	29.83	36.17	38.16	**32.28**
		1500	1499.1	1200.5	31.49	37.53	39.34	**33.81**
		2000	1999.3	1601.0	33.04	38.13	39.67	**34.99**
FOOTBALL 50 frames 5 codes	Proposed JSCC	1000	1000.2	810.0	29.89	36.15	38.06	**32.30**
		1500	1500.4	1220.0	31.54	37.58	39.36	**33.85**
		2000	2000.5	1631.4	33.16	38.19	39.71	**35.09**
	Extended JSCC of [4]	1000	1000.2	809.7	29.88	36.23	38.18	**32.32**
		1500	1500.4	1221.6	31.54	37.56	39.35	**33.84**
		2000	2000.5	1633.8	33.18	38.24	39.78	**35.12**
		10 % packet losses						
FOOTBALL 50 frames EEP 1 code - strongest	Proposed JSCC	1000	1000.2	758.0	29.60	35.88	37.83	**32.02**
		1500	1499.1	1136.1	31.27	37.36	39.22	**33.61**
		2000	1999.3	1515.1	32.65	37.95	39.55	**34.68**
	Extended JSCC of [4]	1000	999.0	757.1	29.57	35.97	37.96	**32.03**
		1500	1499.1	1136.1	31.26	37.37	39.24	**33.61**
		2000	1999.3	1515.1	32.65	37.92	39.54	**34.68**
FOOTBALL 50 frames 5 codes	Proposed JSCC	1000	1000.2	765.2	29.64	35.91	37.85	**32.05**
		1500	1500.4	1151.9	31.32	37.42	39.25	**33.66**
		2000	2000.5	1539.6	31.32	37.42	39.25	**34.76**
	Extended JSCC of [4]	1000	1000.2	763.8	29.63	35.93	37.91	**32.06**
		1500	1500.4	1153.1	31.32	37.44	39.28	**33.66**
		2000	2000.5	1543.8	32.79	37.99	39.59	**34.79**

Table 3 (c): Foreman

		Total target rate (kbps)	Total rate met (kbps)	Source rate (kbps)	Y (dB)	U (dB)	V (dB)	PSNR avg (dB)
		5 % packet losses						
FOREMAN 50 frames EEP 1 code - strongest	Proposed JSCC	500	498.9	399.5	35.03	40.63	43.37	**37.35**
		1000	999.0	800.0	37.25	42.62	45.46	**39.51**
		1500	1499.1	1200.5	38.75	42.78	45.81	**40.60**
	Extended JSCC of [4]	500	498.9	399.5	35.02	40.68	43.42	**37.36**
		1000	999.0	800.0	37.23	42.55	45.38	**39.48**
		1500	1499.1	1200.5	38.75	42.85	45.81	**40.61**
FOREMAN 50 frames 5 codes	Proposed JSCC	500	500.1	406.7	35.08	40.63	43.38	**37.39**
		1000	1000.2	817.4	37.29	42.62	45.45	**39.54**
		1500	1500.4	1226.3	38.84	42.96	45.92	**40.71**
	Extended JSCC of [4]	500	500.1	407.7	35.07	40.77	43.48	**37.42**
		1000	1000.2	819.9	37.27	42.58	45.39	**39.51**
		1500	1500.4	1232.1	38.82	42.96	45.88	**40.69**
		10 % packet losses						
FOREMAN 50 frames EEP 1 code - strongest	Proposed JSCC	500	500.1	379.0	34.89	40.61	43.34	**37.25**
		1000	999.0	757.1	37.06	42.56	45.38	**39.36**
		1500	1500.4	1137.0	38.47	42.67	45.64	**40.36**
	Extended JSCC of [4]	500	500.1	379.0	34.90	40.63	43.37	**37.27**
		1000	999.0	757.1	37.10	42.62	45.45	**39.41**
		1500	1499.1	1136.1	38.49	42.71	45.69	**40.39**
FOREMAN 50 frames 5 codes	Proposed JSCC	500	500.1	384.6	34.94	40.63	43.36	**37.29**
		1000	1000.2	771.7	37.15	42.61	45.44	**39.44**
		1500	1500.4	1159.8	38.58	42.72	45.71	**40.46**
	Extended JSCC of [4]	500	500.1	384.6	34.92	40.62	43.33	**37.27**
		1000	1000.2	775.3	37.12	42.54	45.35	**39.39**
		1500	1500.4	1166.1	38.59	42.66	45.63	**40.44**

Table 3 (d): Mobile

		Total target rate (kbps)	Total rate met (kbps)	Source rate (kbps)	Y (dB)	U (dB)	V (dB)	PSNR avg (dB)
		5 % packet losses						
MOBILE 50 frames EEP 1 code - strongest	**Proposed JSCC**	***500***	498.9	399.5	27.63	33.60	32.91	**29.50**
		1000	1000.2	801.0	30.20	36.21	35.50	**32.08**
		1500	1499.1	1200.5	31.66	36.26	35.78	**33.12**
	Extended JSCC of [4]	***500***	498.9	399.5	27.63	33.62	32.93	**29.51**
		1000	1000.2	801.0	30.18	36.17	35.45	**32.06**
		1500	1500.4	1201.5	31.68	36.23	35.72	**33.11**
MOBILE 50 frames 5 codes	**Proposed JSCC**	***500***	498.9	399.9	27.64	33.62	32.93	**29.52**
		1000	1000.2	809.8	30.23	36.22	35.52	**32.11**
		1500	1500.4	1222.9	31.74	36.26	35.77	**33.17**
	Extended JSCC of [4]	***500***	498.9	399.9	27.63	33.61	32.92	**29.51**
		1000	1000.2	810.7	30.22	36.17	35.45	**32.08**
		1500	1500.4	1222.9	31.74	36.23	35.70	**33.15**
		10 % packet losses						
MOBILE 50 frames EEP 1 code - strongest	**Proposed JSCC**	***700***	699.2	529.9	28.50	34.63	33.89	**30.42**
		1000	999.0	757.1	29.99	35.91	35.21	**31.85**
		1500	1499.1	1136.1	31.44	36.26	35.75	**32.96**
	Extended JSCC of [4]	***700***	699.2	529.9	28.50	34.61	33.86	**30.41**
		1000	999.0	757.1	29.95	36.08	35.31	**31.87**
		1500	1499.1	1136.1	31.44	36.23	35.70	**32.95**
MOBILE 50 frames 5 codes	**Proposed JSCC**	***700***	699.2	533.2	28.52	34.61	33.89	**30.43**
		1000	999.0	765.6	30.02	36.07	35.36	**31.92**
		1500	1500.4	1154.2	31.50	36.26	35.76	**33.01**
	Extended JSCC of [4]	***700***	700.4	533.8	28.53	34.62	33.87	**30.44**
		1000	1000.2	765.7	30.00	36.11	35.32	**31.91**
		1500	1500.4	1156.0	31.51	36.25	35.74	**33.01**

Table 4: Average PSNR figures obtained at 1500 kbps on the four test sequences using SVC over error-free channels and its error-resilient versions over 5% and 10% channel losses.

	Target rate (kbps)	Y (dB)	U (dB)	V (dB)	PSNR avg (dB) 0% losses	PSNR avg (dB) 5% losses	PSNR avg (dB) 10% losses	Delta PSNR (dB) (0% - 5% losses)	Delta PSNR (dB) (0% - 10% losses)
BUS	1500	34.29	41.59	43.14	**36.98**	**35.95**	**35.64**	**1.04**	**1.34**
FOOTBALL	1500	32.58	37.92	39.53	**34.63**	**33.85**	**33.66**	**0.78**	**0.97**
MOBILE	1500	32.73	37.18	36.64	**34.12**	**33.17**	**33.01**	**0.96**	**1.12**
FOREMAN	1500	39.90	44.55	47.09	**41.87**	**40.71**	**40.46**	**1.16**	**1.41**

In Figure 1, the frame-by-frame average PSNR-plots of two individual transmissions of the four test sequences over channels of 1500kbps with an average of 5% packet losses are shown. The plots report the weighted average PSNR for each frame computed as: $PSNR_{YUV,avg} = (4 \times PSNR_Y + PSNR_U + PSNR_V)/6$, where $PSNR_Y$, $PSNR_U$ and $PSNR_V$ are individual frame Y, U and V PSNRs respectively.

These results show that the impact of the losses appears in a bursty manner, which is explained by the use of a block interleaver. The average PSNR-drop between the error-prone and error-free simulations ranges from 1.30 dB for “Mobile” to 1.90 dB for “Football”. Despite of these differences, the visual quality impairment is not significant. This is illustrated in Figure 2, where the frames yielding the largest PSNR-difference between the error-free and error-prone transmissions for two sequences “Football” (frame 34) and “Mobile” (frame 28) are depicted. We observe that the visual impact of errors is limited, despite of the 3.81dB PSNR-difference between the error-free and error-prone version for frame 34 of “Football” and the 2.06 dB PSNR-difference for frame 28 of “Mobile”.

We conclude that the proposed JSCC is robust against packet losses and that the penalty associated with an error-resilient approach for SVC versus the original error-prone version is minimal.

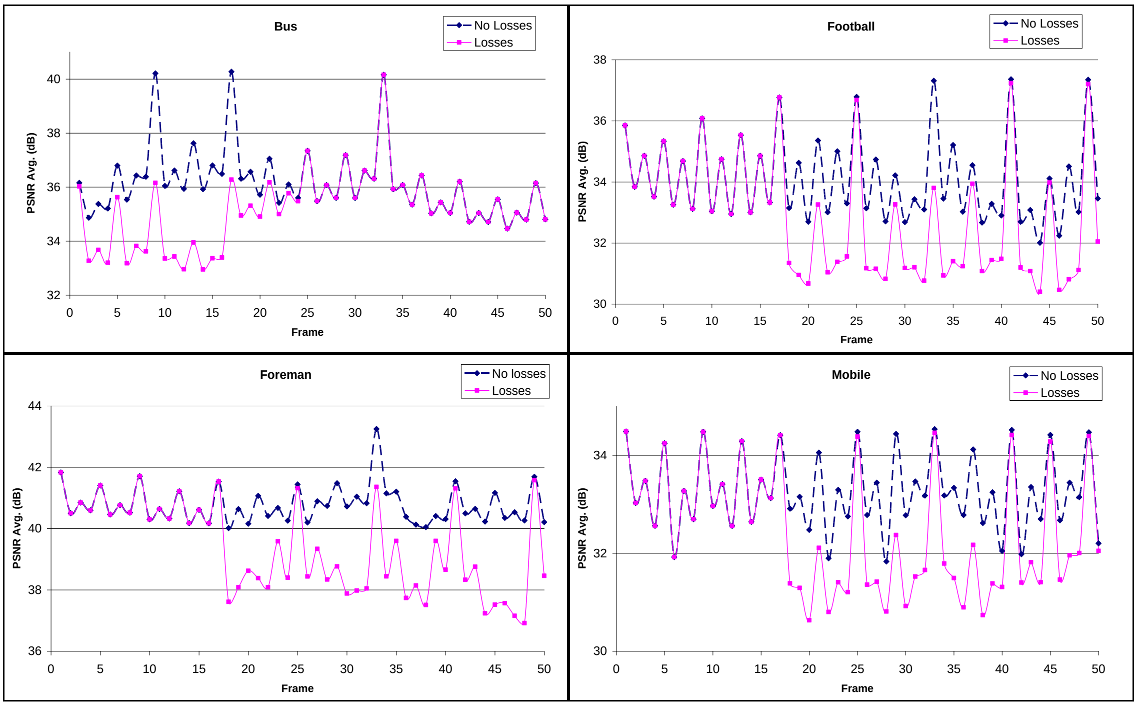
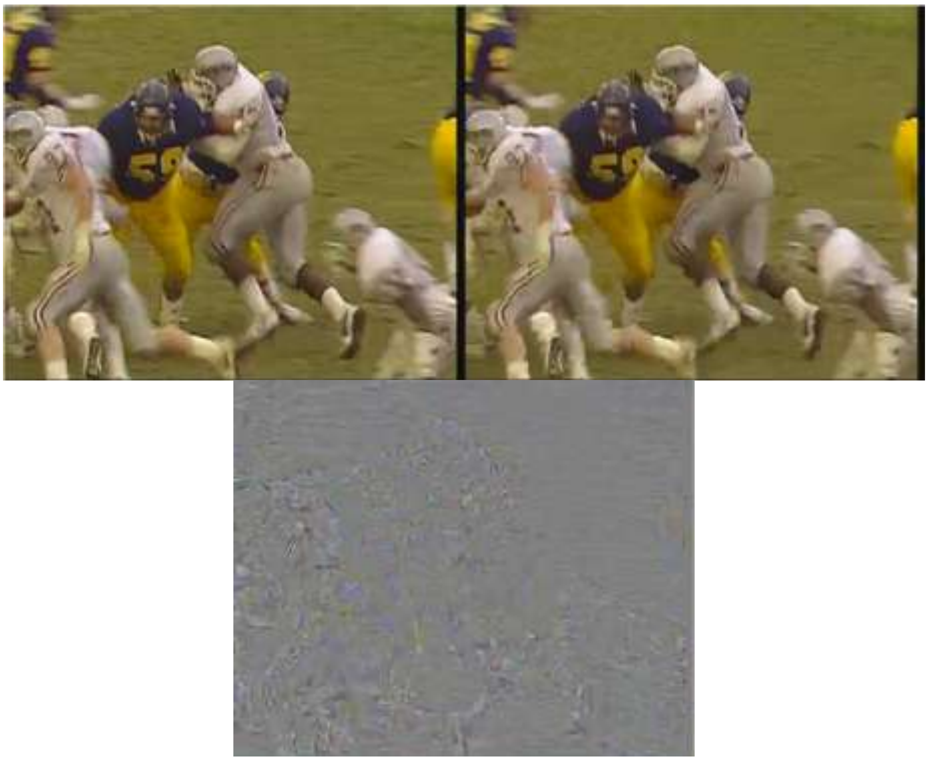


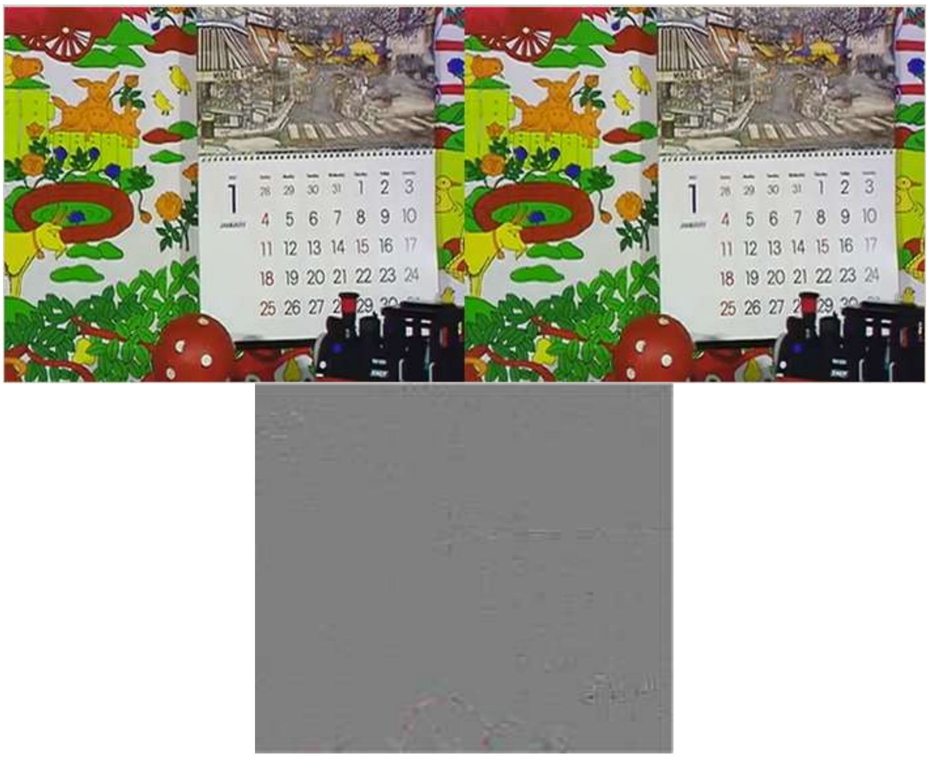
Figure 1: Frame-by-frame PSNR-plots of two individual transmissions of the first 50 frames of the Bus, Football, Foreman and Mobile over channels of 1500kbps introducing 5% of packet losses.

(a) (b)



(c)

(a) (b)



(c)

Figure 2: (above) Frame 34 of the Football test sequence and (under) Frame 28 of the Mobile test sequence transmitted over 5% packet loss channels. (a) Result when no packets are lost. (b) Result when transmitted over a channel where 294 packets of 1221 transmitted packets of the sequence were lost. (c) Difference between (a) and (b).

2.1.2 Complexity Analysis

In the following, we analyze the complexity of our proposed forward programming-based JSCC-algorithm with our extension of the backward programming-based JSCC-algorithm of [4].

For the proposed JSCC-algorithm, the number Y_k of paths that need to be computed for each additional codeword $k, 1 \le k \le M_l$ for d possible protection levels is given by: $Y_1 = d,\quad Y_2 = \left(d(d+1)\right)/2, \ldots\quad,\quad Y_{M_l} = \left(d(d+1)\right)/2$. The total number of paths to compute when transmitting M_l codewords of frame l can thus be written as: $Y = \sum_{k=1}^{M_l} Y_k = d + (M_l - 1)d\frac{(d+1)}{2}$, which is globally of order $O(d^2 M_l)$. For each path $\Pi_k^{[p]}, 1 \le p \le d$ we need to compute the average expected distortion $\overline{D_l}(\Pi_k)$ using our recursive formula (6). The term $\left(D_l(\vec{r}_{l,k}) - D_l(\vec{r}_{l,k-1})\right), 1 \le k \le M_l$ is a source distortion reduction that is interpolated (both in our algorithm as in the algorithm of [4]) from the available source distortion points. As the terms $\left(D_l(\vec{r}_{l,k}) - D_l(\vec{r}_{l,k-1})\right)$ and $(1 - p_f(r_{l,k}, \varepsilon))$ are found using the same complexity in both methodologies, we do not account for them in our computational complexity analysis.

Overall, for the first codeword ($k = 1$), 1 addition and 1 multiplication are needed to compute the average expected distortion for each path $\Pi_k^{[p]}, 1 \le p \le d$. For each additional codeword k, $k \ne 1$, 1 addition and 2 multiplications are required for the computation of each path $\Pi_k^{[q]}, 1 \le q \le d$.

For the backward programming-based algorithm of [4], the number Y_k of paths that need to be computed for each additional step when using d possible protection levels is given by: $Y_{M_l} = d(d+1)/2,\quad \ldots\quad, Y_2 = d(d+1)/2,\quad Y_1 = d$; hence, the total number of paths to be computed when transmitting M_l codewords of frame l is also of order $O(d^2 M_l)$. We observe that for the first cost-to-go computation in the backward algorithm, only the cost as defined by (7) needs to be computed for each protection level. As we are considering codewords of fixed-length, $\frac{l_k}{r_k}$ equals to the fixed codeword length N. For each considered slope, $\lambda\frac{l_k}{r_k}$ can therefore be considered as a constant that is available in memory and that does not require any multiplication. As in the case of the proposed JSCC, the computational complexity of computing the source distortion terms $(-d_k)$ and the probabilities $\left[1 - P\left(r_k, l_k/r_k\right)\right]$ (see equations (7), (8)) is not accounted for. Then, we observe that 1 addition and 1 multiplication are required to compute the first cost-to-go ($k = M_l$) in the fixed-length codeword case. For the following steps in the backward-programming algorithm, the cost-to-go function defined by (8) needs to be computed. We notice that the term $\lambda\left(\frac{l_{k+1}}{r_{k+1}} + \sum_{p=k+2}^{M_l}\frac{l_p}{r_p^*}\right)$ is equivalent to $\lambda\left(\sum_{p=k+1}^{M_l} N\right)$, which, at each subsequent step, can be computed as a simple addition of λN. Equation (8) therefore involves, for each path $\Pi_k, k < M_l$, a total of 5 additions and 2 multiplications. We summarize the resulting number of additions and multiplications performed by the two algorithms in Table 5.

So far, this analysis shows that our proposed JSCC-approach requires 5 times less additions than the method of [4], while no gain is achieved in terms of number of multiplications.

Table 5: Evaluation of the number of additions and multiplications needed for deriving the convex hull in our proposed JSCC-methodology and in the extended algorithm of [4].

Proposed JSCC-methodology	**Number of additions**	**Number of multiplications**
First codeword ($k=1$ **):**	$1\times d$	$1\times d$
Next codewords ($k\neq 1$ **):**	$(M_l-1)\times d\left((d+1)/2\right)$	$(M_l-1)\times d(d+1)$
TOTAL	$d+(M_l-1)\times d\left((d+1)/2\right)$	$d+(M_l-1)\times d(d+1)$
Backward algorithm extended towards SVC of [4]	**Number of additions**	**Number of multiplications**
First step ($k=M_l$ **):**	$1\times d$	$1\times d$
Next steps ($k<M_l$ **):**	$(M_l-1)\times 5\times d\left((d+1)/2\right)$	$(M_l-1)\times 2\times d(d+1)/2$
TOTAL	$d+5\left(M_l-1\right)\times d\left((d+1)/2\right)$	$d+(M_l-1)\times d(d+1)$
Gain: $\frac{\text{TOTAL(extension of [14])}}{\text{TOTAL(proposed)}}$	$O(5)$	$O(1)$

The gains in computational terms are significantly larger than this. Our proposed JSCC-algorithm produces only *once* the virtual envelopes for each frame, and these envelopes do not depend on the target rate or Lagrangian parameter λ. On the contrary, in case of [4] the convex hulls constructed for each frame are λ– and rate–dependent (see equations (7) and (8)), and need to be re-computed for every variation of the Lagrangian parameter and target bit-rate. Hence, considering a channel with a certain bandwidth and packet loss rate, the proposed JSCC will require $O(5\Lambda)$ and $O(\Lambda)$ less additions and multiplications respectively than the approach of [4], where Λ is the number of iterations needed by the bisection method to reach the target bit-rate. In order to give an idea of the order of Λ we measured the number of iterations performed by the bi-section method in both approaches. The results, illustrated in Table 6, show that both algorithms require a comparable amount of iterations to meet the various target bit-rates.

Table 6: Number of iterations required in our proposed JSCC-algorithm and in the algorithm of [4] to achieve different target bit-rates for the first 50 frames of the four test sequences.

Bus				**Foreman**			
	Target Rate (kbps)	**Rate met (kbps)**	**no. iterations**		**Target Rate (kbps)**	**Rate met (kbps)**	**no. iterations**
Viterbi $\varepsilon = 5\%$	**500**	499.98	25	**Viterbi** $\varepsilon = 5\%$	**500**	499.98	22
	1000	999.99	24		**1000**	999.99	28
	1500	1500.00	26		**1500**	1500.00	27
Extension of [14] $\varepsilon = 5\%$	**500**	499.92	33	**Extension of [14]** $\varepsilon = 5\%$	**500**	499.92	31
	1000	999.84	32		**1000**	499.92	31
	1500	1500.00	33		**1500**	1500.00	35
Viterbi $\varepsilon = 10\%$	**500**	499.98	22	**Viterbi** $\varepsilon = 10\%$	**500**	499.98	24
	1000	999.99	21		**1000**	999.99	25
	1500	1500.00	27		**1500**	1500.00	27
Extension of [14] $\varepsilon = 10\%$	**500**	499.98	22	**Extension of [14]** $\varepsilon = 10\%$	**500**	499.92	28
	1000	999.99	21		**1000**	999.84	32
	1500	1500.00	27		**1500**	1500.00	33
Football				**Mobile**			
	Target Rate (kbps)	**Rate met (kbps)**	**no. iterations**		**Target Rate (kbps)**	**Rate met (kbps)**	**no. iterations**
Viterbi $\varepsilon = 5\%$	**1000**	999.99	22	**Viterbi** $\varepsilon = 5\%$	**500**	499.98	22
	1500	1500.00	24		**1000**	999.99	19
	2000	1999.98	24		**1500**	1500.00	24
Extension of [14] $\varepsilon = 5\%$	**1000**	999.84	25	**Extension of [14]** $\varepsilon = 5\%$	**500**	499.92	29
	1500	1500.00	38		**1000**	999.84	32
	2000	1999.92	30		**1500**	1500.00	34
Viterbi $\varepsilon = 10\%$	**1000**	999.99	21	**Viterbi** $\varepsilon = 10\%$	**700**	699.99	34
	1500	1500.00	20		**1000**	999.99	30
	2000	1999.98	20		**1500**	1500.00	33
Extension of [14] $\varepsilon = 10\%$	**1000**	999.84	34	**Extension of [14]** $\varepsilon = 10\%$	**700**	699.84	32
	1500	1500.00	30		**1000**	999.84	35
	2000	1999.92	33		**1500**	1500.00	36

From the above, we can conclude that the major advantage of the proposed JSCC-approach is that the construction of the convex virtual envelopes used in the Lagrangian optimization does not depend on the target rate or Lagrangian parameter λ. Compared to the approach of [4], this brings significant reductions in computational complexity terms, while providing a similar compression performance. Moreover, the proposed JSCC-approach retains all the scalability functionalities of SVC and enables optimized error-resilience in error-prone transmission conditions.

2.2 EMBEDDED MULTIPLE DESCRIPTION SCALAR QUANTIZATION (EMDSQ)-BASED MULTIPLE DESCRIPTION CODING (MDC-2)

2.2.1 Bitstream extractor

In order to transmit the multiple descriptions generated by the SVC MDC-scheme based on EMDSQ, as discussed in deliverable D.3.2, over channels with different available bandwidths, we have developed a bitstream extractor that extracts, from the original descriptions encoded at the highest quality, frame rate and resolution, the NAL units such that an optimized quality is perceived at the receiver. The bitstream extractor extracts a bitstream of lower bit rate and lower quality, and possibly of lower spatial and temporal resolution.

The extractor operates as follows. Consider a video sequence of *F* frames. After SVC-encoding, each frame consists of a set of NAL units. The extractor operates on a fixed number of encoded frames *N* at a time and truncates from the according set of NAL units the necessary ones to achieve the required resolution *res* and bitrate R_{tot}.

Specifically, the extraction algorithm starts by determining the applicable spatial layer and the applicable temporal level. From this information the according NAL units from the *N* first frames are loaded. An example for the extraction of the lowest resolution is given in Figure 3 where all NAL units of one resolution are loaded. The total number of bytes *TotalBytes* required to achieve the target bitrate for the *N* frames encoded at frame rate *f* can be computed as *TotalBytes*= R_{tot}*f/N .

The available bandwidth is then met by processing the NAL units of the subsequent quality layers and subsequently adding the length of each processed NAL unit until the total number of bytes required is met. The extracted NAL units are finally transmitted in their original order; i.e. on a frame-by-frame basis. We note that this algorithm introduces an initial delay of *N/f* seconds and that by varying the value of *N* the delay can be reduced or increased.

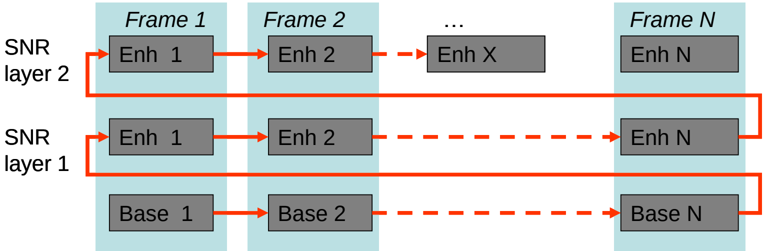


Figure 3: Example of the MDC2 extraction process for extraction of one resolution of one encoded description.

2.2.2 Experimental Results

In the following experiments we use the proposed MDC2-based SVC codec for encoding the CIF test sequence "Bus" (150 frames) with 3 FGS layers and 1 resolution layer. The following encoding parameters are used. The quantization parameter is set to 40 and the minimum and maximum GOP sizes are set on 16. Also, the no-fast-P-skip parameter is set. The two descriptions generated by the MDC2 codec are transmitted over various bitrates by extracting the streams with our proposed extractor.

2.2.2.1 Lossless transmission

In Table 7 we show the results of using our proposed extractor. From the results it can be seen that the extractor meets the target rates as close as possible. In the extractor, we did not include truncation of the FGS layers, such that the rate is never closely met. This was done to optimize the received quality of the video sequences. Also, the results in Table 7, demonstrate that the use of 2 descriptions does not impair the PSNR-results in the lossless case.

Table 7: Results of using the MDC2 extractor

Target R_{md1} (kbps)	Target R_{md2} (kbps)	Achieved rate R_{md1} (kbps)	Achieved rate R_{md2} (kbps)	Total rate (kbps)	PSNR $_Y$	PSNR $_U$	PSNR $_V$	PSNR $_{YUV}$
1024	*0*	*1005*	*0*	1005	29.57	37.81	39.15	32.54
0	*1024*	*0*	*996*	996	28.02	37.81	39.15	31.51
512	*512*	*500*	*476*	976	28.46	37.81	39.15	31.80
2048	*0*	*1963*	*0*	1963	30.92	37.81	39.15	33.44
0	*2048*	*0*	*2002*	2002	28.72	37.81	39.15	31.98
1024	*1024*	*1005*	*996*	2002	31.02	37.81	39.15	33.51

2.2.2.2 Lossy transmission

2.2.2.2.1 Balanced Situation

In Table 8, we present the results of the transmission of the MDC2 descriptions over packet loss channels introducing 5, 10 and 20% of packet losses. The introduced losses are randomly generated and the experiments are averaged over 50 transmissions. The results show clearly the benefits of using multiple descriptions to transmit the video sequences and demonstrate that gains of up to 5dB in PSNR can be achieved by using two descriptions to transmit a sequence.

Table 8: Transmisision of MDC2 descriptions over channels introducing 5, 10 and 20% of packet losses.

	Target R_{md1} (kbps)	Target R_{md2} (kbps)	Total rate (kbps)	PSNR $_Y$	PSNR $_U$	PSNR $_V$	PSNR $_{YUV}$
5% losses	*1024*	*0*	*1024*	25.92	37.61	38.83	*30.02*
	0	*1024*	*1024*	24.17	37.54	38.69	*28.82*
	512	*512*	*1024*	28.16	37.80	39.13	*31.60*
	1536	*0*	*1536*	26.34	37.62	38.82	*30.30*
	0	*1536*	*1536*	24.18	37.52	38.65	*28.81*
	768	*768*	*1536*	29.32	37.80	39.14	*32.37*
	2048	*0*	*2048*	26.72	37.61	38.83	*30.55*
	0	*2048*	*2048*	24.62	37.55	38.70	*29.12*
	1024	*1024*	*2048*	30.49	37.81	39.14	*33.15*
10% losses	*1024*	*0*	*1024*	23.33	37.39	38.48	*28.19*
	0	*1024*	*1024*	22.01	37.31	38.34	*27.28*
	512	*512*	*1024*	27.53	37.78	39.08	*31.16*
	1536	*0*	*1536*	23.53	37.38	38.48	*28.33*
	0	*1536*	*1536*	22.09	37.29	38.32	*27.33*
	768	*768*	*1536*	28.47	37.77	39.08	*31.79*
	2048	*0*	*2048*	23.76	37.34	38.44	*28.47*
	0	*2048*	*2048*	22.43	37.31	38.36	*27.57*
	1024	*1024*	*2048*	29.39	37.77	39.07	*32.40*
20% losses	*1024*	*0*	*1024*	20.02	36.88	37.77	*25.79*
	0	*1024*	*1024*	19.15	36.76	37.62	*25.17*
	512	*512*	*1024*	25.45	37.64	38.85	*29.72*
	1536	*0*	*1536*	19.86	36.86	37.77	*25.68*
	0	*1536*	*1536*	19.19	36.68	37.53	*25.16*
	768	*768*	*1536*	26.03	37.65	38.87	*30.11*
	2048	*0*	*2048*	19.96	36.81	37.72	*25.73*
	0	*2048*	*2048*	19.45	36.77	37.66	*25.37*
	1024	*1024*	*2048*	26.54	37.65	38.86	*30.45*

2.2.3 Conclusions

We have propsed a novel JSCC-methodology for transmitting scalable video sequences over packet loss channels. Our experiments demonstrate that this JSCC-methodology delivers competitive results against state-of-the-art solutions at a much lower complexity.

Also, we have assessed the performance of our proposed MDC2 system based on EMDSQs. The results clearly show that gains up to 5 dB in PSNR can be achieved by using a multiple description coding system when transmitting over packet loss channels.

2.3 Splitting Unbalanced MDC-3

The MDC-3 is an Unbalanced Multiple Description (UMD) approach: in a two-channel scenario (Figure 4), an UMD coder generates a coded video stream at almost full quality, along with a second version at reduced quality. These versions make up the descriptions and are each transmitted over a different network.

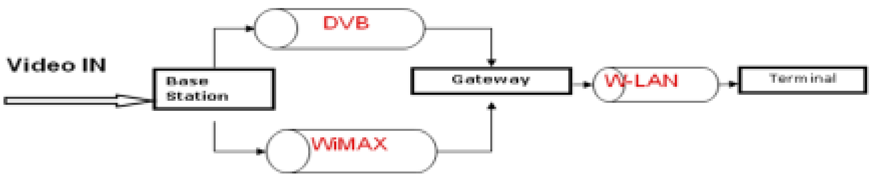


Figure 5: SUIT Transmitter system.

The low quality version constitutes a ‘base layer splitted’ version and is essentially redundant. The descriptions are unbalanced, since the low quality version will typically be of much lower rate than the almost full quality description. UMD enables improved utilization of available bandwidth in the underlying networks. As shown in Figure below the upper layer can use data coming from the lower layer for prediction.

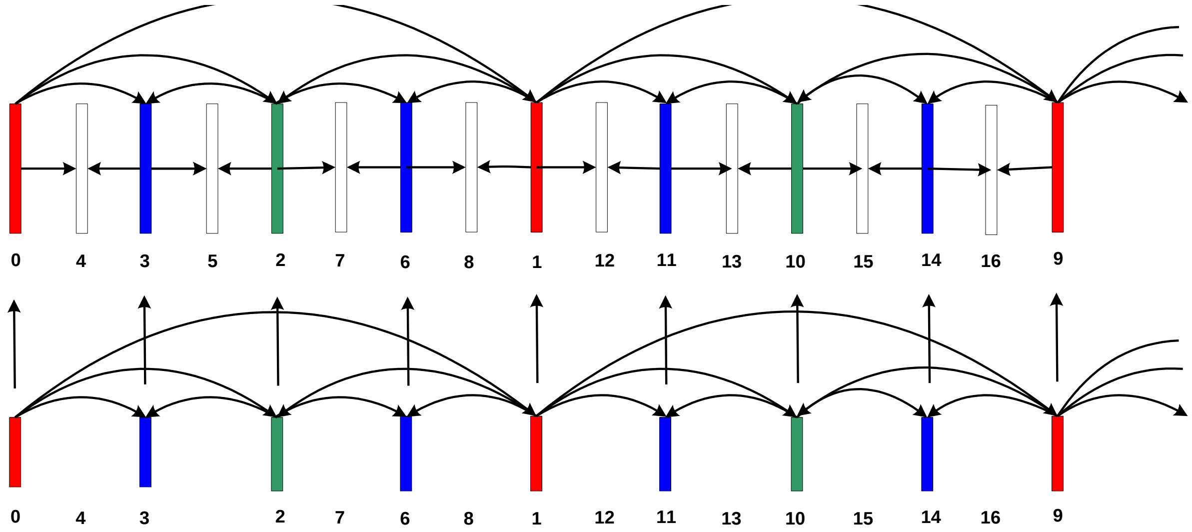


Figure 6: Base layer and enhanced layers.

The encoding algorithm is based on the splitting unit that is embedded into the existing H.264 SVC encoder. It determines the splitting break point (SBP), until all coefficients are copied into both description and from it on, coefficients are copied alternatively to each description. As shown in the following Figure below the splitting unit operates on the quantized DCT coefficients. Two descriptions are created by duplicating header information, motion vectors and quantized DC coefficients of Intra coded blocks. Each description also includes nonzero AC coefficients whose quantization levels equal or exceed the threshold value (corresponding to a SBP), it means they are duplicated into the two descriptions and then the remaining coefficients, after SBP, are split in accordance with our partitioning algorithm to meet the balanced condition for the base layer. The splitting procedure is done in such a way that when a quantized coefficient value is copied into one of the descriptions, a corresponding zero is sent to the other description.

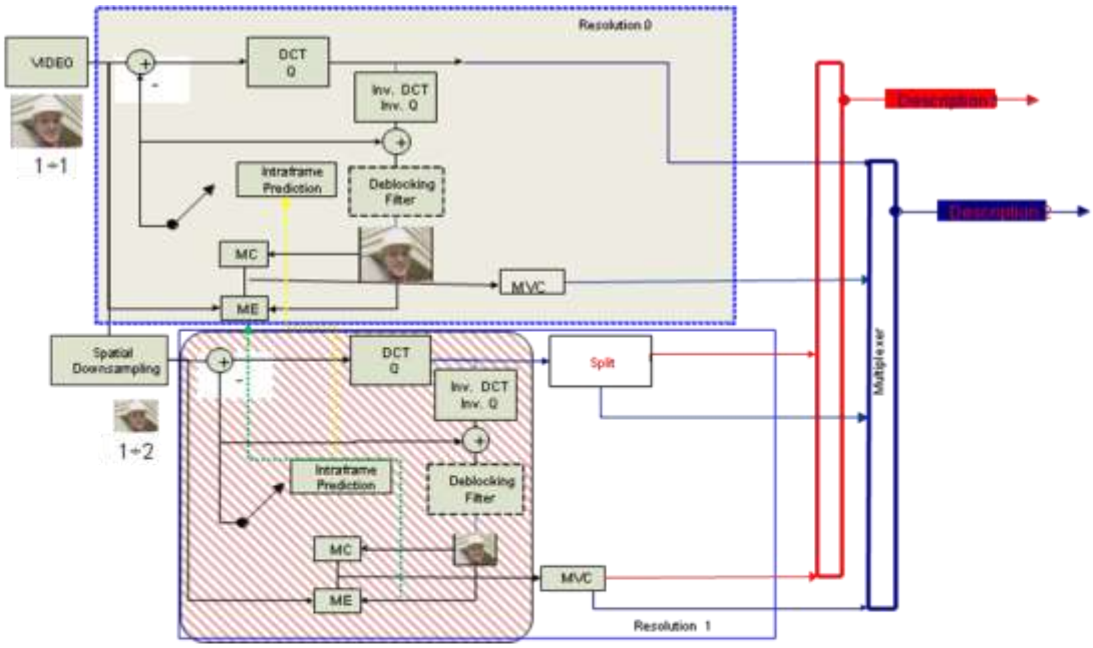


Figure 7: MDC-3 encoding system.

2.3.1 Experimental Results

Both description, generated by MDC-3, are balanced for base layer. ρ is the distortion measure given by,

$$\rho = \frac{\sum_{i=1}^{n}(R_{WiMAX} + R_{DVB-T} - R_{SD})}{\sum_{i=1}^{n} R_{SD}} \quad ,$$

where,
R_{WiMAX} is the bitrate for description of WiMAX
R_{DVB-T} is the bitrate for description of DVB-T base layer
R_{SD} is the bitrate of a single description
n is the number of blocks

For instance, assuming ρ=50%, the description 1 in DVB-T base layer and the description 2 provided a PSNR=38.5. We have used CIF test sequence "Bus" (150 frames).

The following figure shows the PSNR. We used the proposed MDC-3 SVC codec for encoding the CIF test sequence "Bus" (150 frames) with 2 FGS layers and no spatial layers. The following encoding parameters are used, bitrate parameter minimum and maximum are set 50 Kbps to 500 Kbps. Also, the no-fast-P-skip parameter is set. The two descriptions generated by the MDC-3 codec are transmitted over a different network at various bitrates.

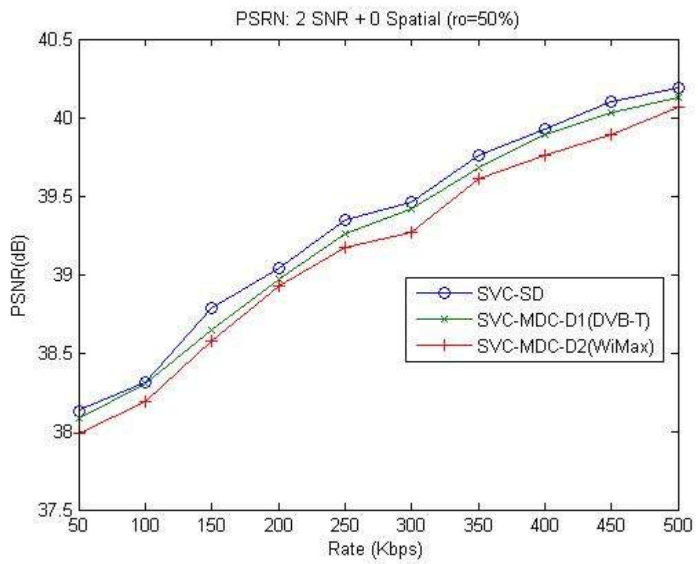


Figure 8: Comparison of description distortions at different bitrates using the "Bus" sequence and ρ = 0.50.

The following figure shows the PSNR with the same conditions but the test sequence has 2 FGS layers and 2 Spatial layers.

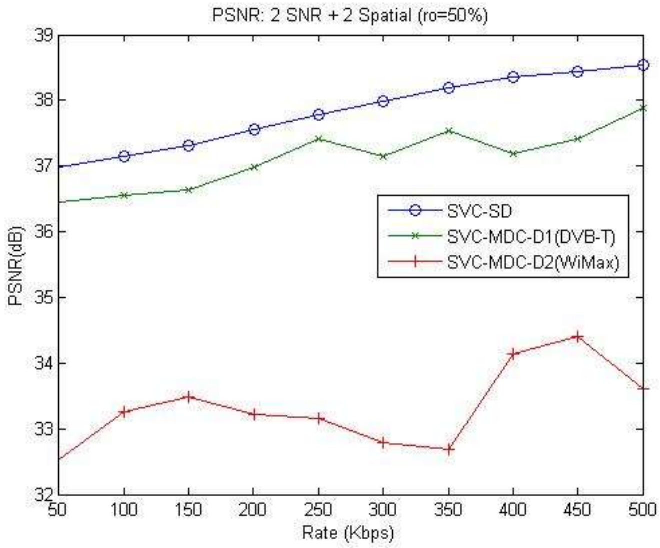


Figure 9: Comparison of description distortions at different bit rates using the "Bus" sequence ρ = 0.50.

As we can observe in the two figures above, the low quality layer or base layer split into the description associated to WiMAX has the worst PSNR and the other description (DVB-T) has almost full quality, very close to the original single description, as we expected.

2.4 Conclusions

We have proposed a novel MDC solution. It provides balanced description for the base layer. The proposed scheme can provide broadcasting TV services to WiMAX terminals as well as HD DVB-T terminals. The DVB-T description shows very small quality degradation in comparison to the single description around 0.5 dB.

3 Overview of UPA Scheme

The Unequal Power Allocation (UPA) scheme for transmission of scalable video coded packets over a WiMAX channel has been explained in D3.3. In WiMAX transmission, the bandwidth is divided into a number of sub-channels. Data is transmitted to a user through a subcarrier or using multiple sub-carriers. Therefore, the first step for a power allocation scheme is to distribute the subcarriers among the users under a total power constraint.

In scalable video transmission, more than one layer of video may be transmitted to a user through multiple subchannels. So the second step for a power allocation scheme is to allocate subcarriers to the layers of video for a user, subject to the power constraint of that particular user. The power is allocated based upon the importance of the layer and channel state information. Since the base layer is more important than the enhancement layer, more power and a greater number of subchannels are allocated to the base layer. If the channel conditions indicate a high packet loss rate, then all the power available to a user may be allocated to the base layer. An efficient unequal power allocation algorithm increases the decoded video quality at the receiver by protecting the base layer from transmission errors. Further details of the proposed UPA scheme can be found in D3.3.

In the following sections, a simulator for demonstrating the UPA scheme is presented. Then using the UPA simulator, experimental results are discussed.

3.1 UPA Simulator

The UPA simulator provides a platform to run experiments using the UPA scheme, and obtain objective and subjective results. The software is written using Microsoft Visual C++ and Microsoft Foundation Class Library. The UPA simulator has been tested on the Windows XP operating system. A user interface dialogue accepts all the parameters required to simulate and generate the results. All the files should be copied in the executable file directory. The files required by the simulator are:

- Suit encoder (x264.exe)
- Suit parameter file
- YUV sequences
- WiMAX trace simulator program (packet_err_trace_generator.exe)
- WiMAX error pattern files
- Mean program (mean.exe)
- SUIT decoder (suitdecoder.exe)
- Modified YUV viewer (YUVviewer.exe)
- Psnr program (psnr.exe)

The UPA simulator consists of three parts: encoder, Wimax simulator, and UPA algorithm as shown in Figure 13.

3.1.1 Encoder

The video sequence is encoded using the SUIT encoder. Its parameters include the number of frames to be encoded, the quantization parameter, the frame rate, and the input and output file names. A command line example is as follows:

```
x264.exe --frames 100 --no-fast-pskip -v -q 40 -o output.264 input.yuv 325x288
```

The "--no-fast-pskip" option is mandatory. The other parameters can be changed according to the requirement. Futher parameters and information about the video layers are controlled through the suitencoder.cfg file.

In the UPA simulator, the values of QP, encoding rate, and name of the video sequence are passed through the user interface.

3.1.2 WiMAX Simulator

Simulation of WiMAX baseband channels requires appropriate error trace files. Therefore, the UPA simulator requires error traces generated using a WiMAX baseband simulator for a range of SNR values. The details of the simulation procedures are described in D2.3 The parameters used for generation of the error traces are described in SUIT deliverable D3.3. Two MCS modes have been used for testing: MCS5 and MCS7 as shown in Table 9. The name of the trace file indicates the MCS mode and the channel SNR value. For example, trace_mcs5_14.40, indicates that it was generated using MCS5, with a channel SNR of 14.40dB. The values of parameters (such as modulation scheme, trace start random point, video packet length, and number of slots) are passed through the user interface window to the WiMAX simulator program. An example command for the trace simulator with parameters is as follows:

```
packet_err_trace_generator.exe 4 c:\UPASim\pusc_vehA_ctc_60kmph/trace_mcs5_6.00
c:\UPASim\pusc_vehA_ctc_60kmph\trace_mcs5_12.00 135 500 out.264 corrupetd.264 300
```

Note that the WiMAX simulator accepts two trace files, one for the base layer corruption, and the other for the enhancement layers. The selection of the trace files is carried out by the UPA algorithms, whose parameters are explained in next subsection. The results of the simulator are stored in a text file for packet loss rate analysis.

Table 9 Properties of MCS modes used

Parameter	MCS 5	MCS7
Modulation	QPSK	16 QAM
Repetition	3/4	3/4
Number of bits per data slot	72	6
Maximum number of concatenated slots	144	3

3.1.3 UPA Simulator

The user interface window for UPA simulator is shown in Figure 10. The channel condition is selected using a slide bar. ‘1’ indicates the maximum packet loss rate for the modulation scheme selected in WiMAX parameter box. ‘12’ represents an error free channel. The channel condition selection represents feedback in the WiMAX simulator. The UPA algorithm uses this indicator to allocate power for the base layer and enhancement layer, as described in D3.3. After selection of appropriate parameters for the SUIT encoder, and the WiMAX and UPA simulators, the simulation is carried out in the following order:

1- The SUIT encoder generates the compressed encoded video sequence using the parameter file settings.

2- The UPA algorithm selects the corresponding trace files for corruption of the video packets

3- The SUIT decoder decodes and conceals the missing frames. The output is a decoded video sequence.

4- Video sequences with and without the UPA scheme are displayed simultaneously, as shown in Figure 11.

5- The objective results in the form of average PSNR are displayed in the output box of the simulator dialogue window, as shown in Figure 12.

The above procedure can be repeated to find appropriate settings for the simulator.

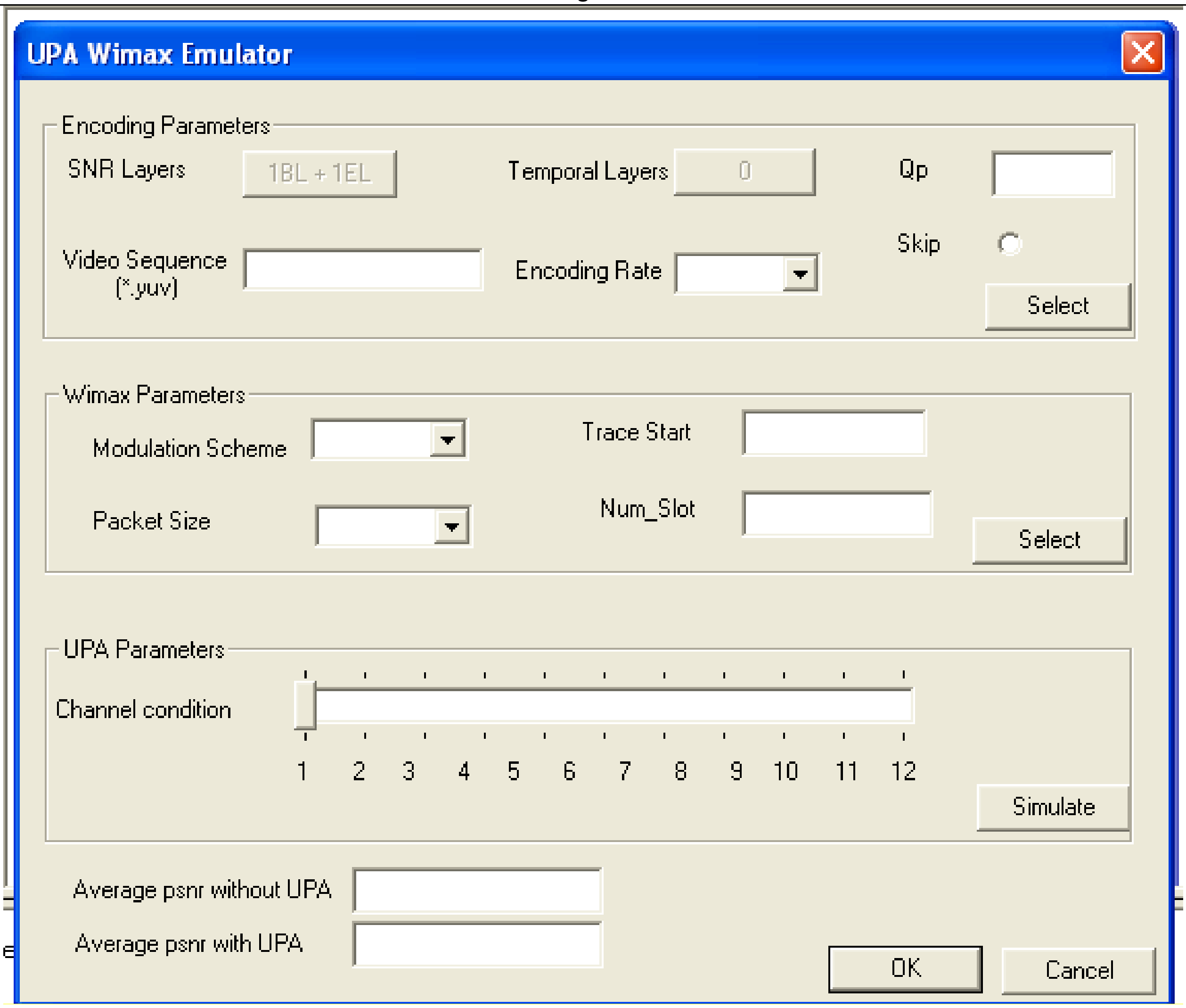


Figure 10: User interface dialogue window for UPA simulator

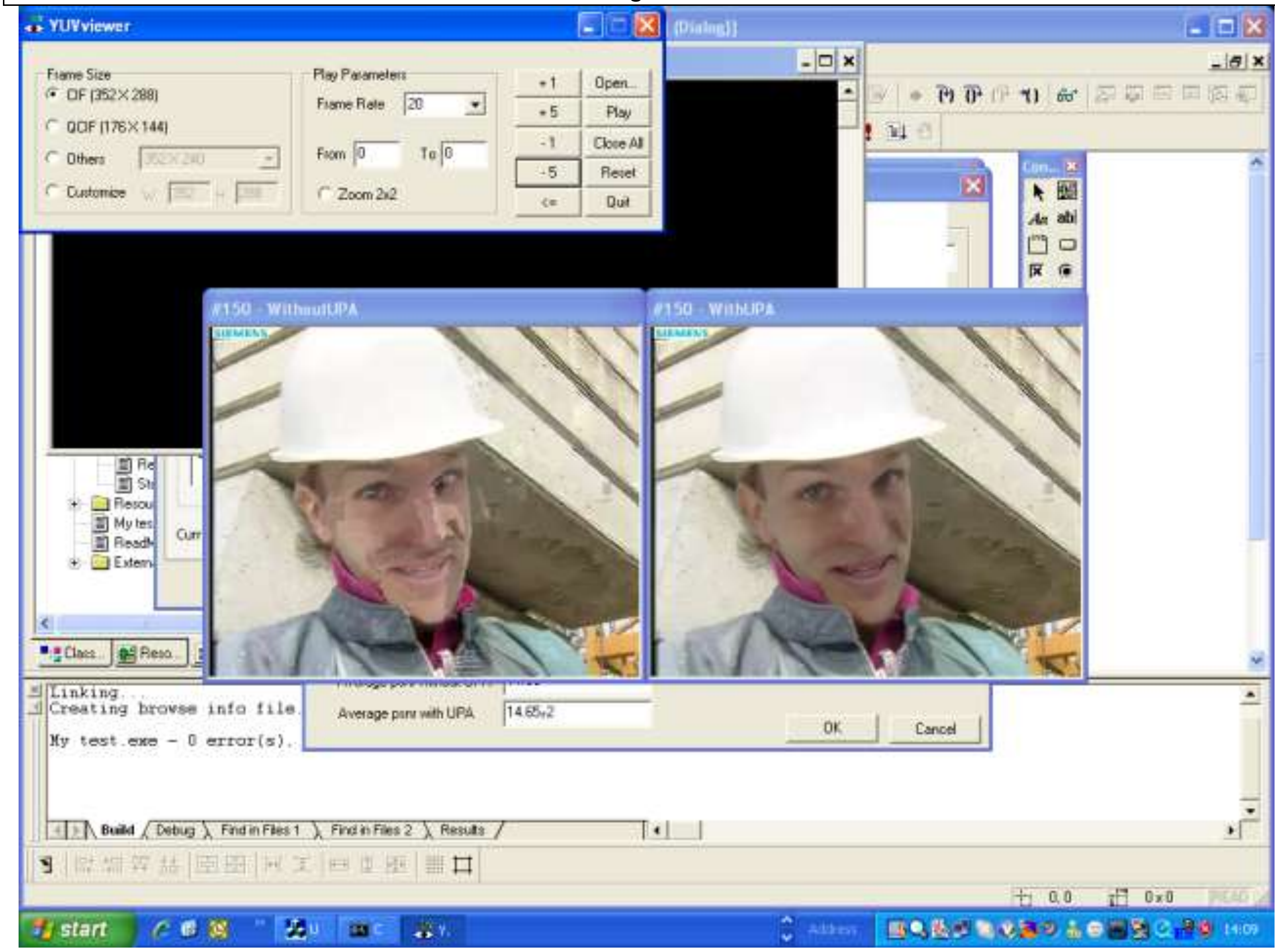


Figure 11: Screen shot of subjective comparison for foreman sequence

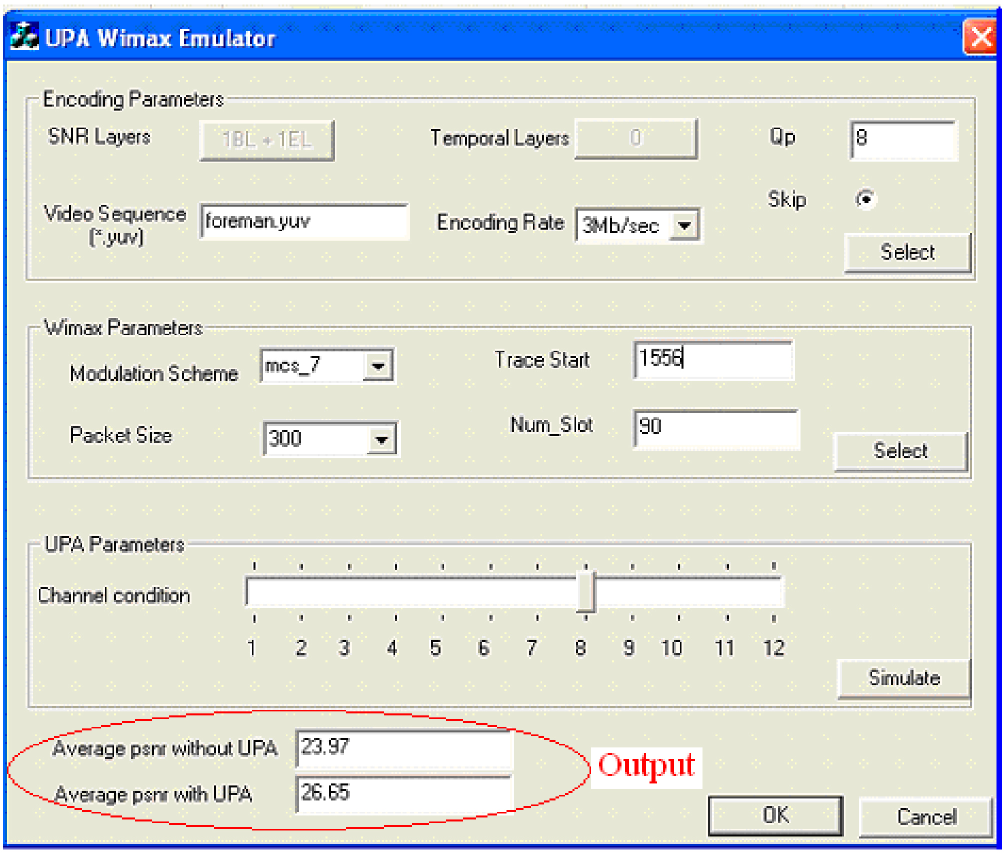


Figure 12: Output screen for the UPA simulator showing an objective comparison of the results.

3.2 UPA Experiments

Using the UPA simulator described in section 2.1, a number of experiments have been carried out. Two CIF video sequences, Foreman and City, have been tested.

3.2.1 Parameter Settings

Figure 13 shows the block diagram of the system for the simulation of the unequal power allocation scheme. The test sequences Foreman and City have been compressed offline using the SUIT H.264 SVC codec with the parameters listed in Table 10 (a). The bitstream extractor is used to extract the layers and analyse the transmission bandwidth requirement.

For the purposes of simulation, we have adopted a two stage process. The first stage is the generation of an error trace using a WiMAX baseband simulator for a range of SNR values, for which the simulation procedures are described in SUIT Deliverable Document 208. The parameters used for generation of the error trace are given in Table 10 (b).

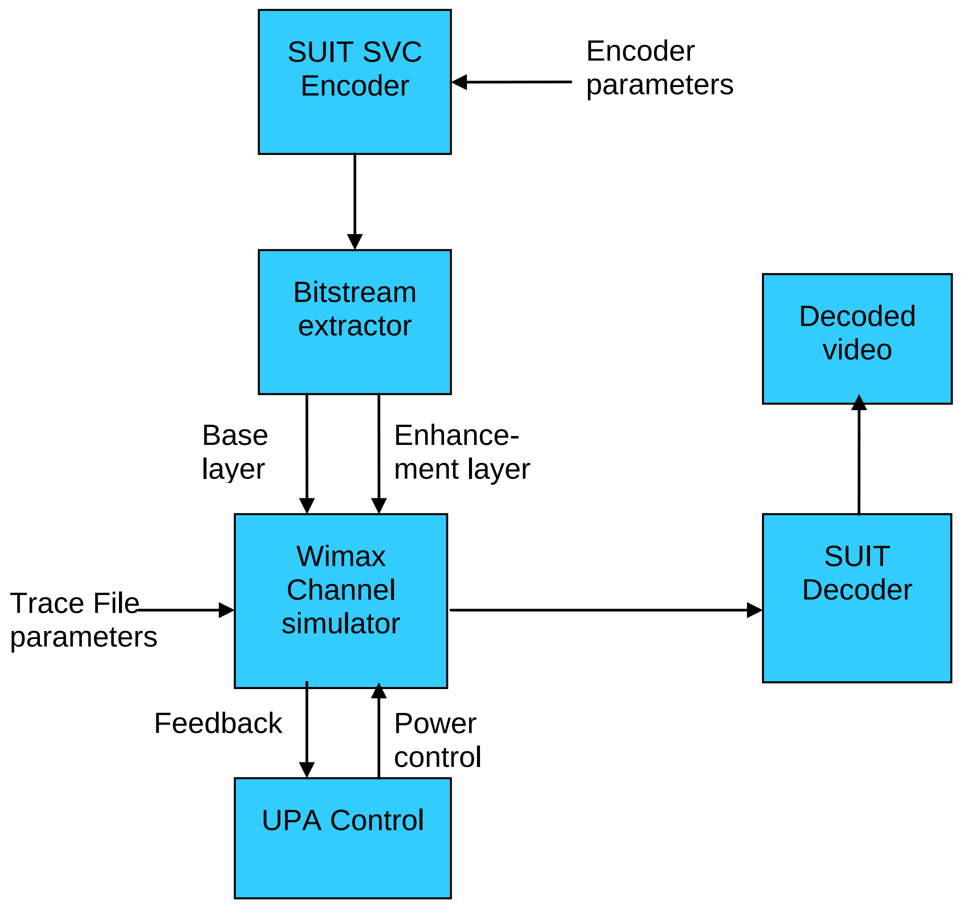


Figure 13: Simulation platform for testing of proposed UPA scheme.

In the second stage of the process, we have performed the system level simulations including the effects of unequal power allocation based upon the priority of video data using the parameters given in Table 10 (b) and Table 10 (c). Therefore, this model simulates a time varying channel in which the effects of multipath is already simulated during trace generation process using the ITU Vehicular A model. The total received power can be mapped to an SNR value as mentioned in D3.3. For the given SNR value, the pre-simulated error trace that has the closest SNR is used to corrupt the video stream transmitted through the simulator. In the simulations, we have used 2 video layers, for which the simulator selects the corresponding two error traces.

Table 10 Wimax simulation parameters for (a)SUIT encoder (b) error trace(c) UPA simulator

(a)

Parameter	Values
Sequence	Foreman, City
Frames	300
Frame sequence	IPPP…
Resolution	CIF
Encoding rate	Approximately 600 kbs/sec
Decoder concealment	Frame copy
Frame rate	25 frames per second
Number of layers	1 BL + 1 EL

(b)

Parameter	Values
Length of Trace (s)	15
Bandwidth	10 MHz
Permutation	PUSC
Channel Coding	CTC
Terminal Speed (km/h)	60
Test Environment	ITU Vehicular A
MCS Mode	MCS5, MCS7

(c)

Parameter	Values
SNR range MCS5	5 to 13 dBs
SNR range MCS7	10 to 17 dBs
Power increment step ΔP	0.5 to 0.75
Power distribution layers	2

3.2.2 Lookup Tables for UPA Algorithm

The proposed UPA algorithm does not use the complex optimisation solutions to solve the power allocation distribution among the users or distribution of power among the video layers for a particular user. As described in D3.3, the UPA scheme calculates power for the subchannels with the help of lookup tables, which translate the estimated distortion into SNR ratios. The percentage packet loss rate and estimated distortion in dBs are the parameters obtained by the UPA algorithm through the feedback channel. Depending upon the MCS scheme being used, and the channel conditions, the UPA algorithm distributes the power budget among the layers of video packets. The objective is to minimize the received video distortion such that:

$$\min_{p} Distortion\,(p_{m,s}) \quad subject\ to \quad \sum_{i=1}^{L} P_i \leq P_{m,S} \qquad (1)$$

where $P_{m,S}$ is the total power budget for user m and subchannel s. We assume that all the subcarriers assigned to a user have equal power. This means allocation of two subcarriers to a particular layer doubles the power, and so forth. Hence, power increment is available in fixed step increments of ΔP_s. The UPA algorithm is provided with the information about the number of layers being transmitted to the user, the condition of the subcarrier channels $N_{m,s}$, and the number of subcarriers available.

Initially, the available power for a user is distributed equally among the subchannels $N_{m,s}$ Then depending upon the feedback about the received video distortion, power distribution is adjusted according to lookup tables similar to Table 11 and Table 12. These tables were obtained experimentally, after a number of simulations describing the best distribution of power among base layer and enhancement layers. If the channel condition indicates high packet loss rate corresponding to the low decoded video quality, then all the power available for a user is used to transmit the base layer.

Table 11 Power distribution lookup table for MCS5 with Delta P = 0.55, and average base layer to enhancement layer data ratio of 0.6

SNR Level	PLR	BER	MCS 5	
			$D_{estimated}$	BL:EL
5.00	0.2296032	0.9932137	8.50	2.75
6.65	0.1422165	0.7630214	8.50	2.54
8.30	0.0453449	0.2806538	10.46	2.13
8.85	0.0273837	0.1768291	11.17	2.02
9.40	0.0153232	0.1038376	16.87	1.97
9.95	0.0077446	0.0553034	18.09	1.76
10.50	0.0034710	0.0260556	21.36	1.59
11.05	0.0014022	0.0109829	23.80	1.51
11.60	0.0005203	0.0042650	30.66	1.47
12.15	0.0001747	0.0014701	33.11	1.25
12.70	0.0000615	0.0005000	33.17	1.07
13.25	0.0000243	0.0002094	34.02	1.00

Table 12 Power distribution lookup table for MCS7 with Delta P = 0.60, and average base layer to enhancement layer data ratio of 0.75

SNR Level	PLR	BER	MCS 7	
			$D_{estimated}$	BL:EL
10.00	0.2296032	0.9932137	8.86	3.11
11.80	0.1422165	0.7630214	9.32	3.11
13.60	0.0206413	0.1441821	12.93	3.03
14.20	0.0105427	0.0759231	18.33	2.65
14.80	0.0049684	0.0366872	20.83	2.07
15.40	0.0022480	0.0167462	26.14	1.96
16.00	0.0009883	0.0074538	26.96	1.69
16.60	0.0004131	0.0031282	27.32	1.21
17.20	0.0001762	0.0013462	29.22	1.17
17.80	0.0000689	0.0005410	31.85	1.05
18.40	0.0000302	0.0002231	32.54	1.00
19.00	0.0000134	0.0001000	34.02	1.00

BL:EL = base layer : enhancement layer = ratio of distribution of power between base layer and enhancement layer

3.2.3 Objective Results

This subsection provides objective performance of the proposed unequal power allocation scheme for video transmission over WiMAX simulated channels. Two CIF sequences 'foreman' and 'city' are encoded at approximately 600 kbits/sec with 2 SNR scalable layers; including one base layer and one enhancement layer. If a baselayer frame is lost during transmission, the decoder copies the missing frame from the previously decoded frame. Other test parameters are described in section 3.2.1. The WiMAX error patterns, developed in SUIT, have been used to simulate wireless links. To simulate the UPA scheme, a packet error trace program has been used, that accepts two error trace files, and drops the NAL units in case of any error in the packet. The simulations were repeated 20 times to obtain stable results using the UPA simulator described in section 3.1.3.

Table 13 and Table 14 show the test results for the 'foreman' and 'city' sequences, with and without the proposed UPA scheme for a WiMAX simulated channel with modulation schemes of MCS5 and MCS7. As can be seen from Table 13 and Table 14, performance with the UPA scheme is better than equal power allocation communication at the same transmission rate. The main reason for this is due to the fact that the proposed UPA scheme protects the base layer effectively, which results in more graceful degradation of decoded video performance in error prone channel conditions. It should also be noted that the range of SNR for which acceptable video quality can be achieved, significantly increases with the UPA scheme.

Table 13 Objective results with and without the UPA scheme for Wimax simulated channel. The sequence is Foreman CIF at 30fps and encoded at 600 kb/sec.

MCS scheme	Channel SNR	Decoded PSNR	
		With UPA	Without UPA
MCS 5	5.00	8.5	8.5
	6.65	8.5	8.5
	8.30	11.18	9.74
	8.85	15.93	10.42
	9.40	21.34	12.2
	9.95	24.59	14.19
	10.50	27.44	17.28
	11.05	29.67	21.54
	11.60	32.32	27.21
	12.15	33.72	31.21
	12.70	33.92	33.32
	13.25	34.02	34.02
MCS 7	10.00	9.23	8.5
	11.80	9.99	8.66
	13.60	14.84	11.02
	14.20	23.06	12.91
	14.80	26.62	16.65
	15.40	30.24	21.44
	16.00	31.23	24.9
	16.60	32.02	27.62
	17.20	32.92	29.43

	17.80	33.52	31.28
	18.40	34.02	32.68
	19.00	34.02	34.02

Table 14 Experimental results for CITY CIF sequence with and without the UPA scheme. The sequence is encoded at approximately 600 kb/sec and transmitted over a WiMAX simulated channel, and then decoded using SUIT decode with simple frame concealment for lost frames during transmission.

MCS scheme	Channel SNR	Decoded PSNR	
		With UPA	Without UPA
MCS 5	5.00	9.34	9.34
	6.65	9.34	9.34
	8.30	10.83	9.62
	8.85	13.13	11.13
	9.40	17.03	12.34
	9.95	21.27	15.49
	10.50	25.72	18.87
	11.05	30.27	23.28
	11.60	32.44	27.78
	12.15	33.17	32.44
	12.70	33.17	33.17
	13.25	33.17	33.17
MCS 7	10.00	9.45	9.34
	11.80	10.14	9.24
	13.60	13.63	10.82
	14.20	18.92	12.54
	14.80	22.97	15.96
	15.40	26.93	19.08
	16.00	30.17	22.36
	16.60	32.17	25.86
	17.20	33.17	29.57
	17.80	33.17	32.39
	18.40	33.17	33.17
	19.00	33.17	33.17

3.2.4 Subjective Results

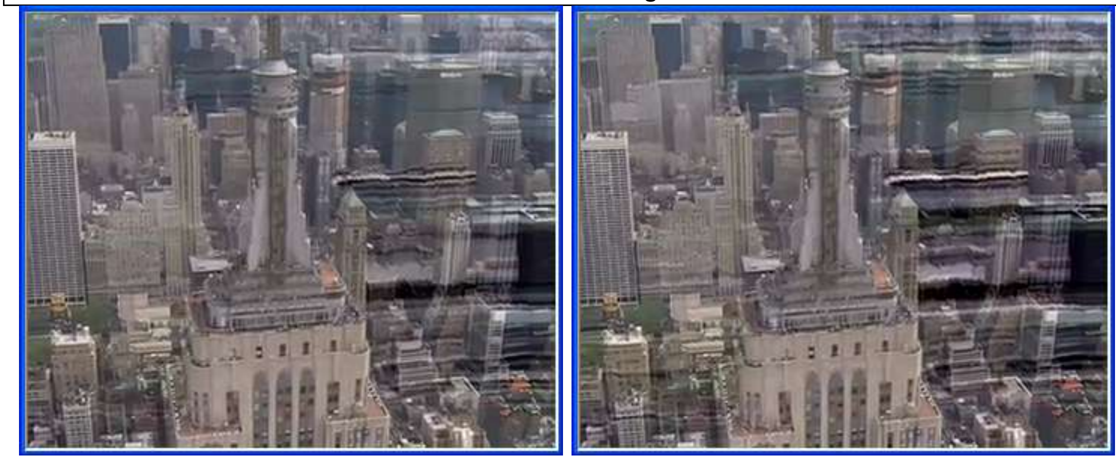
In this section the effects of the unequal power allocation scheme on the decoded video quality received through the WiMAX channel are presented. Figure 14 (a) and Figure 14 (c) are the decoded video frame numbers 80 and 160, respectively, with unequal error protection scheme. Figure 14 (b) and Figure 14 (d) are the frames numbers 80 and 160, respectively, without unequal error protection scheme. The WiMAX modulation scheme of MCS 5 is used. Similar results for the CITY sequence can be observed in Figure 15.

(a) (b)

(c) (d)

Figure 14: Selected frames from foreman sequence encoded at 600 kbits/sec, and transmitted over a WiMAX channel with MCS 5 modulation and SNR = 11.05.(a) Frame 80 with UPA scheme (b) Frame 80 without UPA (c) Frame 160 with UPA (d) Frame 160 without UPA scheme.

(a) (b)



(c) (d)

Figure 15: Performance comparison of selected frames from the 'City' sequence encoded at 600 kbits/sec, and transmitted over WiMAX channel with MCS 5 modulation with SNR = 11.60.(a) Frame 80 with UPA scheme (b) Frame 80 without UPA (c) Frame 160 with UPA (d) Frame 160 without UPA sccheme.

3.3 Conclusion

An unequal error protection scheme for scalable video coded packets was proposed in SUIT deliverable D3.3. In this document, detailed experiments were carried out to show the performance improvement of decoded video quality over simulated WiMAX channels due to the UPA scheme. A number of simulations were carried out to generate look up tables for various channel conditions and modulation schemes, which are used by the UPA algorithm. The lookup tables translate the estimated video distortion into power levels. The objective results show that the UPA scheme extends the range of SNR transmission for which the acceptable video quality can be achieved. The subjective results also show improvements in video quality due the fact that UPA scheme protects the base layer more than the enhancement layer in a range of SNR values. For the experimental and demonstration purposes, an unequal power allocation simulator was also built and described in this deliverable. The simulator is written in C++ and has been tested on the Windows XP platform. It combines the SUIT encoder, WiMAX simulator, and UPA scheme to provide the objective and subjective results using a user interface window. Future work may be extended to perform tests for more than two SNR layers including the temporal scalability.

4 Rate-Control Module

WLAN transmission has to deal with two main difficulties: limited bandwidth and high error probability typical in wireless channels. In such conditions, ensuring a certain level of quality of the decoded sequence may require a prohibitive cost in terms of rate, especially in scenarios where rate constraints are imposed due to the existence of limited bandwidth. In these cases, the objective is to determine the transmission policy which maximizes quality of the delivered content, meeting the rate constraints, leading to a rate-distortion trade-off. The field of the techniques oriented to the search and find of these optimal policies is commonly known as “rate-control”.

SUIT Gateway tackles this issue through the introduction of a binary rate control strategy, by means of a rate control module (Figure 16) specifically developed to optimize the transmission of scalable video within an IEEE 802.11g wireless local area network. It particular implementation will be described along this chapter.

4.1 Module architecture

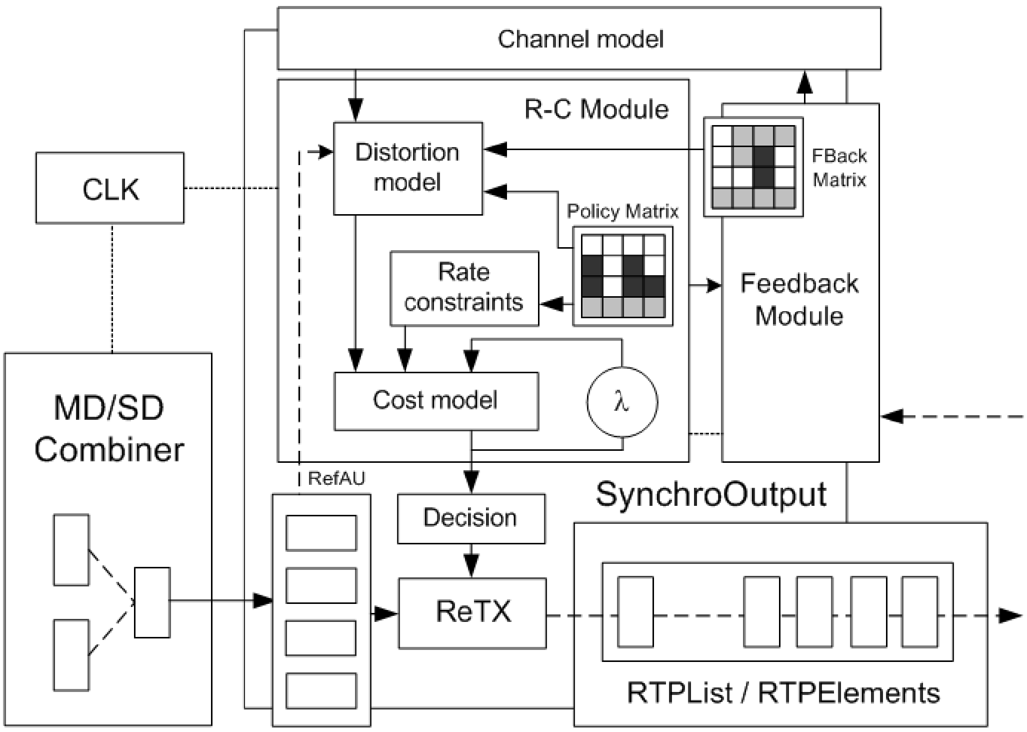


Figure 16: Rate-control module scheme

Implementation of SUIT Gateway’s rate-control strategy is based on a discrete wireless channel model, a distortion model and tools for the rate management. Values of rate $R(\pi)$ and distortion $D(\pi)$ corresponding to certain transmission policy π are obtained, together with the value for the cost $C(\pi)$ associated to that policy. Among all possible policies evaluated, the optimal decision π^* will be the one that minimizes the cost function under some maximum rate constraints.

As will be seen, the cost function is based on the rate and distortion values, and it also considers the stochastic behaviour of the channel by means of the discrete channel model. Its parameters can be computed ‘offline’ previously to the operation of the Gateway. In case feedback is available, these parameters can be updated on-line in order to capture the dynamic behaviour of the wireless channel.

The proposed rate-control strategy can be applied both, to the case of a single transmission chance, and to a scenario with *T* possible transmission opportunities. In addition, the availability or not of feedback from the Terminal is also considered. Concrete aspects regarding these approaches and their implementation will be depicted in the next sections.

4.1.1 PERCP channel model

SUIT Gateway makes use of a discrete channel model that encloses two levels of granularity: (i) a fine grain one which models the error probability for a single transmission unit (NAL unit in this case) and (ii) a coarse one that is based on the Packet Error Rate (PER class) obtained during the transmission of a cluster of *L* NAL units. This model is based on the one proposed in D2.4 – "802.11g WLAN network modelling".

4.1.1.1 Model parameters

Fundamentals of this Packet Error Rate Class Partitioned (PERCP) model are shown in Figure 17. PERCP model is divided into *C* different error classes (*Ci*). Each class aims to represent the behaviour of the channel for a given range of PER values. Boundaries between these classes can be defined 'a priori', or being obtained as a result of a previous training process under certain criteria. After the transmission of *L* NALUs, the channel moves to another class according to the transition probabilities between classes.

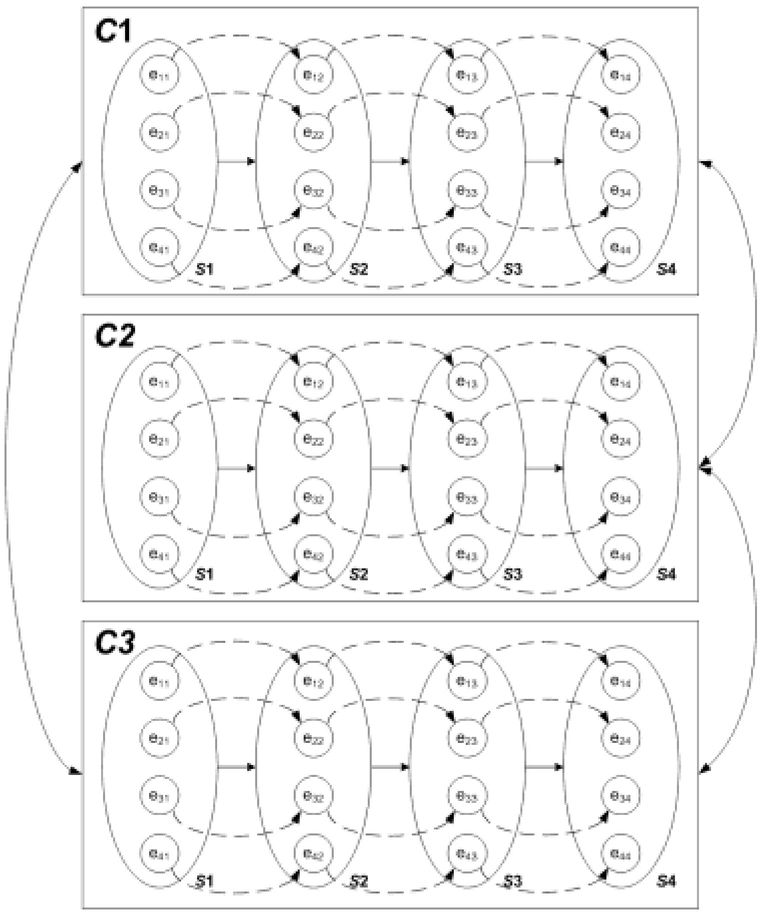


Figure 17: PERCP channel model

Model parameters for the classes can be computed from statistics measured during the transmission of all the NALUs included in a cluster of *L* frames. Regarding the transmission probabilities between classes, they can also be determined by using an 'off-line' training step. However, if feedback is available, the Gateway could know the current class the system is in by processing transmission statistics coming in feedback packets from the terminals. In this case, the parameter *L* defines a trade-off between model's speed to adaptation to variations in the behaviour of the channel and resources devoted to the delivery of feedback statistics

Within each class, the characterization of the error probability for each NALU differentiates at two levels:

1. Type of NALU within an Access Unit. For example, given an Access Unit with four NALUs, (see **Error! Reference source not found.**) the model would consider four different error probabilities, one for each NALU.
2. Transmission opportunity or state. In the example shown in **Error! Reference source not found.**) four transmission opportunities are allowed, then, the model would consider for a given Access Unit four sets of error probabilities, one for each NALU.

Thus, the probability of having a NALU l_i received up to certain state t_j can be modelled as a Markov chain.

In case there is no feedback available, within a class *c*, for a NALU l_i in a state t_j, the error probability e_{ij} represents the probability that the NALU has not arrived to the Terminal before the next transmission opportunity ('Forward Trip Time'). It is defined as:

$$e_{ij} = P(FTT_{ij} > t_{j+1} - t_j) = P(FTT_{ij} > \Delta t)$$

In the case the packet is lost, $FTT_{ij} \to \infty$ is assumed.

In a scenario where acknowledgement packets from the Terminal are used, e_{ij} represents the probability that the 'Round Trip Time' ('Forward Trip Time' plus the time that the ACK packet spends on its trip to the Gateway, 'Back Trip Time', BTT) is longer than the time till the next transmission opportunity. It is defined as:

$$e_{ij} = P(RTT_{ij} = FTT_{ij} + BTT > t_{j+1} - t_j) = P(RTT_{ij} > \Delta t)$$

It can be observed that both definitions are equivalent if Back Trip Time is assumed to be very small ($BTT_{ij} \to 0$).

Therefore, assuming that at each transmission opportunity a given NALU l_i is sent, the probability that it has been received at a certain state t_j, $P_{tx}(l_i,t_j)$, can be computed as:

$$P_{tx}(l_i, t_j) = 1 - \prod_{k=0}^{k<j} e_{ik}$$

Let $\pi_i = \{\pi_{i0}, \pi_{i1}, \ldots \pi_{ij-1}\}$ be the decisions taken for a given NALU l_i up to state t_j, where $\pi_{ij} \in \{0,1\}$ indicates whether the NALU has been sent (1) or not (0). The probability that the terminal has received it, $P_{tx}(\pi_i)$, can be computed as:

$$P_{tx}(\pi_i) = 1 - \prod_{k=0}^{k<j} e_{ik}^{\pi_{ik}}$$

4.1.1.1.1 Off-line parameters estimation

For the discrete channel model optimization, statistics obtained from Activity 2.4 (WLAN channel modelling) are taken as input. In addition, new measurements have been performed that validate those models.

Real RTP level packet loss traces are obtained from a single laptop, for a given set of transmission conditions (channel speed, SNR…). Specific values were selected according to the transmission scenarios proposed in the SUIT project.

Traces are captured by a wireless network analyser (Wireshark/Ethereal), and then processed in order to define the concrete model parameters.

4.1.1.1.2 On-line parameters adaptation

In a more sophisticated approach, feedback statistics obtained during the transmission can be used to adapt channel model, by means of the “dynamic channel adaptation module”. This module has been developed in order to permit dynamic adaptation of channel probabilities to the actual behaviour of the transmission, by processing packet level statistics retrieved from feedback from the Terminal.

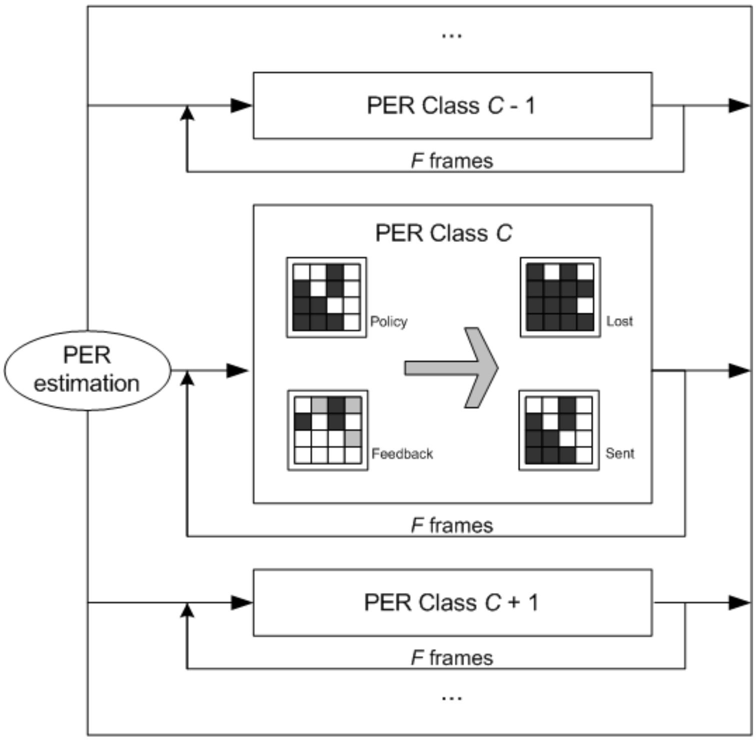


Figure 18: Channel model dynamic adaptation

Dynamic adaptation module (Figure 18) proceeds in the following manner. When the last retransmission state is over, feedback and policy matrixes are processed to update the new error probabilities. From these matrixes, also statistics about losses depending on NALU type, or losses within current cluster of *F* Access Units, can be obtained.

The speed of the update process is controlled by a parameter α that defines the weight of the on-line computed statistics with respect to the off-line ones. The greater this value is, the stronger the initial conditions will be, so, it will be less sensitive to adaptation.

4.1.2 Distortion model

The rate control module utilizes a simple additive distortion model. This model is based on the assumption that there are dependency relationships between the different NALUs within an Access Unit *f*. Thus, the distortion of the decoded Access Unit decreases as NALUs arrives to the decoder, provided that the NALUs from which they depend have already arrived.

Let D_{f0} be the distortion incurred when a entire Access Unit *f* is lost, and Δd_{fi}, the decrease in distortion when NALU l_i arrives to the decoder and their precedent NALUs have also arrived successfully. Thus the arrival of all *L* NALUs which form part of that Access Unit would drive to a minimal distortion whose value is:

$$D_{f\,\min} = D_{f0} - \sum_{i=0}^{L} \Delta d_{fi}$$

Values of this model have been defined by analysing distortion characteristics of some available SVC coded training sequences. For each scalability level, an average distortion value is obtained by comparing sequence frames to the original non-coded frames in terms of MSE. From these values, it is possible to determine additive distortion Δd_{fi},

Considering an SVC-coded stream, consisting of a base layer and L-1 enhancement layers, we can define D_{fi} as the distortion when i consecutive layers are received. Therefore, D_{f0} stands for distortion when no layers are received at all; D_{f1} is the distortion for the base layer, D_{f2} for the base plus one enhancement layer, and so on. Then, the incremental distortion values can be found as:

$$\Delta d_{fi} = D_{fi} - D_{fi+1}$$

where distortion values are assumed to be expressed in natural units (MSE).

Taking into account that, at a state t_j the probability that NALU l_i has arrived to its destination is $P_{tx}(\pi_i)$, and that for achieving a reduction in distortion Δd_i all NALUs l' from which l_i depend must have also arrived, then the general expression of expected distortion for a set of policies $\pi=\{\pi_0, \pi_1 \ldots \pi_L\}$ can be defined as:

$$D(\pi) = D_{f0} - \sum_{i=0}^{L} \Delta d_{fi} \prod_{l' \le l_i} P_{tx}(\pi_{l'})$$

Implemented distortion model stores and gives access to these additive values of Δd and permits the calculation of expected distortion for a concrete transmission policy π.

4.1.3 Rate calculation

Let B_i be the length in bytes of NALU l_i. Then, the rate $R(\pi)$ associated to a concrete policy π will be:

$$R(\pi) = \sum_{i=0}^{L} B_i \rho(\pi_i)$$

Where $\rho(\pi_i)$ is the number of times NALU l_i is retransmitted along policy π_i.

4.2 Rate-optimization process

Cost function $C(\pi)$ consists of a lagrange formulation that includes a trade-off between rate $R(\pi)$ and distortion $D(\pi)$:

$$C(\pi)=D(\pi)+\lambda R(\pi)$$

where λ is a Lagrange multiplier.

The aim of rate-control module calculations is to minimize this function, in order to find the optimal transmission policy $\pi*$ under certain maximum rate constraints.

These maximum rate constraints are integrated into the searching algorithm of this optimal policy by means of an iterative search process with the objective of finding the more accurate value for multiplier λ that meets the rate constraint . In (Figure 19), the searching process flow chart is portrayed.

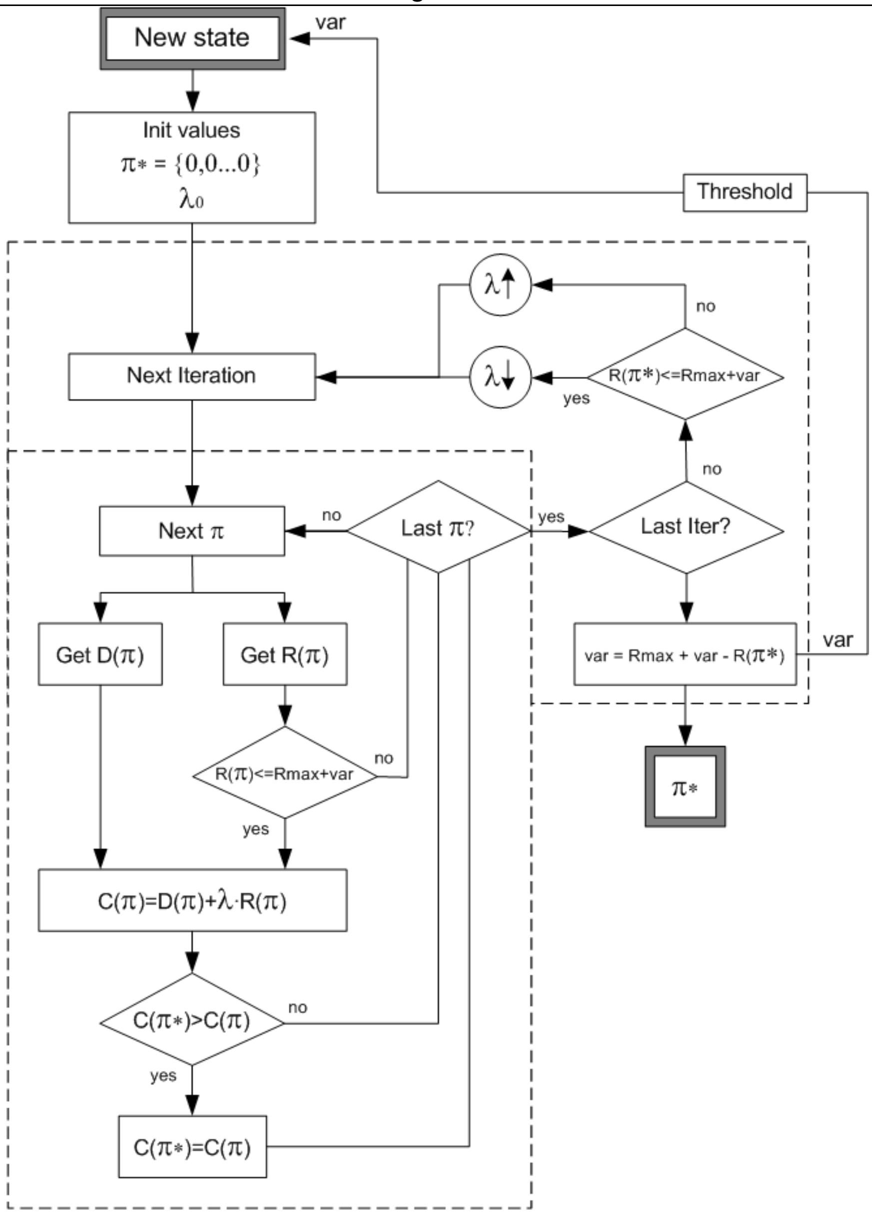


Figure 19:Flow chart for the optimal policy searching process.

Process starts with a new state, taking as parameter a value *var* which specifies the allowed variation over R_{max}, the rate constraint.

- In the first state for each frame, *var* equals to 0.
- In following states, *var* is updated as the increment (either positive or negative) between the maximum allowed rate R_{max} and the rate for the selected optimal policy $R(\pi^*)$. A threshold is applied to *var* in case it exceeds a maximum deviation value specified as a percentage of R_{max}. Therefore, the smaller this percentage is, the more strict the application of the maximum rate constraint is.

The proposed rate control algorithm takes a binary decision for each NALU at each transmission opportunity. Therefore, the set of possible policies contains 2L elements as an Access Unit is assumed to be comprised of L NALUs. As a first approach, and due to the fact that optimization takes place for small NALU sets, searching is performed in an exhaustive manner, by selecting the policy with a least cost among all possible transmission policies.

In order to ensure a real time operation, the transmission policy is determined in a maximum of *Q* iterations. In each iteration, the algorithm proceeds as follows:

1. For each policy π, values of rate $R(\pi)$ and distortion $D(\pi)$ are calculated, taking into account the feedback information and the prior states policies matrixes.
2. If $R(\pi)$ exceeds the rate constraint (determined by $R_{max}+var$)), policy π is discarded, and the next policy is then evaluated.

3. If not, the value of cost associated, with the current value of λ, is calculated.
4. This cost value is compared with the minimum one stored up to that moment: if the new one is smaller, currently evaluated policy becomes to be optimal policy.

As soon as the 2^L vectors have been processed, value of $R(\pi□)$is compared to R_{max}. Then, the value of λ is modified according to the sign of the comparison. The process stops with the last iteration, and optimal policy that has been determined is sent as a decision to the transmission scheduling module.

4.3 Results

According to IEEE 802.11 standard, multicast data transmission is less robust than unicast, due to the lack of acknowledgement packets and retransmissions at link level.

Table 15: NALU / Packet error rate for unicast / multicast transmission

	Base layer	First SNR enhancement layer	Second SNR enhancement layer
Unicast	0.57%	0.57%	0.59%
Multicast	12.64%	17.18%	18.47%

Table 15 shows mean values for packet error rate in transmission tests conducted in our test scenario. These tests have been performed under different SNR conditions. As expected, the use of multicast can drive to a remarkable degradation in packet transmission efficiency. It is interesting to note that, due to their size, error rate for enhancement layer NALUs is higher than for base layer NALUs. Therefore, a separate modelling of these error probabilities, as the one proposed in section 4.1.1, can drive to an improvement in the whole system performance.

Video transmission is especially sensitive to losses in this environment. Existent dependencies among frames, or components of the same frame, imply that the loss of a single packet can lead to a severe decrease in final quality of the decoded sequence. In SVC, these dependences are found among the base layer and successively enhanced layers.

For this multicast scenario, the use rate-control strategies is twofold: (i) ensure that rate of transmitted stream does not exceed rate constraints imposed for dynamic wireless channel conditions; and (ii) optimize this control scheme in terms of the rate-distortion trade-off, being aware of intrinsic scalability characteristics of the transmitted stream. As a result, decision of delivering or not certain NAL unit will be taken in an intelligent manner, for instance by boosting the retransmission of base layer NALUs (mandatory for the decoding of a frame), or avoiding the delivery of enhancement layer NALUs if the base or previous enhancement layers' NALUs have been lost. Table 16 shows the results for some of the conducted tests:

Table 16: Distribution of received NALUs in a rate-controlled multicast scenario.

SNR (dB)	Rate constraint (Kbps)	Base layer	First SNR enhancement layer	Second SNR enhancement layer	Resultant Bitrate (Kbps)
40	No RC	87.73%	84.00%	82.48%	1206
25		86.99%	81.64%	80.58%	
40	500	87.73%	70.36%	8.98%	481
25		87.11%	71.04%	9.07%	
40	1000	95.91%	91.45%	41.53%	897
25		94.68%	92.05%	40.96%	
40	1250	99.26%	97.38%	60.05%	1086
25		93.18%	84.13%	43.28%	

40	2000	97.89%	89.73%	83.44%	1531
25		94.84%	87.59%	80.86%	
40	No restriction	93.08%	84.25%	74.00%	1760
25		93.23%	88.86%	81.27%	

For the multicast tests depicted in **Error! Reference source not found.**, SVC sequences (12000 frames) with a base layer and two SNR enhancement layer were transmitted to two different laptops: one near to the WLAN Access Point operating at 36Kbps (SNR=40 dB), the other placed two rooms away (SNR=25 dB).

As can be observed, the use of rate-control strategies tailors the rate to imposed constraints, with an especial protection of the most important layers (especially the base) that drives to a significant decrease of lost frames in the decoding process. It is interesting to notice that, even for those cases with a lower resultant bit rate, the percentage of received NALUs for the base layer (and thus, decodable frames) is higher than for the non controlled test.

Enhancement layer NALUs are smartly added as rate constraints are less demanding. For instance, in the scenario limited to 1000 Kbps, more than 90% of the frames can be decoded up to the first enhancement, outperforming the non-controlled transmission approach with the 75% of its rate.

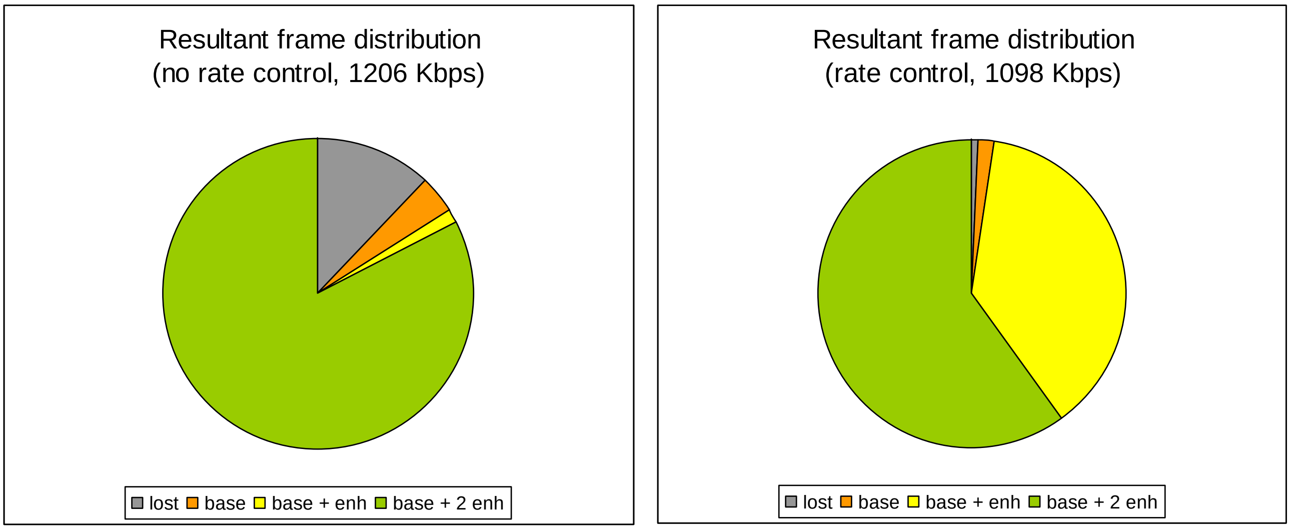


Figure 20: Resultant frame distributions: non-controlled vs. rate-controlled transmission.

Effect of rate-control on the resultant frame distributions at the decoder for two streams of a similar rate can be seen in Figure 20. Although the number of frames that can be decoded with the maximum quality (base + 2 enhancement layers) diminishes, the use of rate-control boosts the delivery of the most important NALUs, so number of lost frames is dramatically reduced.

As a general conclusion, it is proved that the introduction of rate control strategies improves the number of decodable frames, and also the distortion at the decoder, by intelligently retransmitting the most significant NAL units of the stream. This optimization is performed under certain maximum rate constraints and, in most of the cases, rate controlled streams outperform non controlled transmission.

5 Conclusions

This deliverable focused on the optimization of the video system in SUIT. Next to an extension of the technologies developed in the previous deliverables, this deliverable also contained some new approaches to optimize the video coding system in terms of error resilience.

Firstly, both the MDC-2 (D3.1, D3.2) and the proposed joint source-channel coding (D3.3) technologies were evaluated. Experimental results for MDC-2 show that the use of multiple description coding over single description coding can result in up to 5dB gain in PSNR when transmitting over packet loss channels.

Experiments for the JSCC method show that it performs similar to the current state-of-art. The advantage of the proposed method is twofold. First, the computation is less complex. And second, the results of the computation can be used multiple times.

Secondly, a novel MDC approach was introduced and tested. The method, called MDC-3, allows for a balanced base layer in both WiMAX and DVB-T.

Thirdly, a simulator for the UPA algorithm described in D3.3 was introduced. The simulator is used to test the algorithm in a WiMAX context. The objective results of these tests show that the algorithm extends the range of SNR transmission for which the acceptable video quality can be achieved by better protecting parts of the base layer.

Finally, a rate-control module for the Gateway was described and tested. The module works based on both a discrete wireless channel model that captures the stochastic behaviour of the channel and can be updated in case feedback is available, and a distortion model customized for SVC streams. This module is in charge of optimizing the quality of the decoded sequence by intelligently retransmitting the most significant NAL units of the stream. This optimization is performed under certain maximum rate constraints. Results have shown that in most of the cases, rate controlled streams outperform non controlled transmissions.

6 Acronyms

AVC: Advanced Video Coding

BEC: Binary Erasure Channel

BL: Base Layer

BLER: Block Error Rate

BS : Base Station

CGS: Coarse-Grain Scalable

EBCOT : Embedded Block Coding with Optimized Truncation

EEP: Equal Error Protection

EL: Enhancement Layer

EMDSQ: Embedded Multiple Desription Scalar Quantization

FGS: Fine-Grain Scalable

GOP: Group Of Pictures

JPEG : Joint Photographic Experts Group

JSCC : joint source-channel coding

JSVM: Joint Scalable Video Model

JVT : Joint Video Team

LDPC: Low-Density Parity-Check

MGS: Medium-Grain Scalable (MGS)

MPEG : Moving Picture Experts Group

MSE : Mean Square Error

NAL: Network Abstraction Layer

OFDM: Orthogonal Frequency-Division Multiplexing

PEG : Progressive Edge Growth

PSNR : Peak Signal to Noise Ratio

QAM: Quadrature Amplitude Modulation

QLA : Quality Level Assigner

QP: Quantization parameter

RD: Rate Distortion

SEI : Supplemental Enhancement Information

SVC : scalable video coding

UEP : unequal error protection

UPA : Unequal Power Allocation

VCEG : Video Coding Experts Group

VCL : Video Coding Layer

7 References

[1] I. Amonou, N. Cammas, S. Kervadec, and S. Pateux, Enhanced SNR scalability for layered CGS coding using quality layers, Input Document to JVT JVT-S044, Geneva, Switzerland, 1-7 April 2006.

[2] J. Reichel, H. Schwarz, and M. Wien, Joint Scalable Video Model JSVM-6, Draft Output Document from JVT, JVT JVT-S202, Geneva, Switzerland, April 2006.

[3] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamental, Standards and Practice* Dordrecht, Kluwer Academic Publishers, 2001.

[4] Z. Wu, B. Ali, and M. W. Marcellin, "Joint Source/Channel Coding for Image Transmission with JPEG2000 over Memoryless Channels," *IEEE Transactions on Image Processing*, vol. 14, pp. 1020-1032, August 2005.

[5] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Transactions on Information Theory*, vol. 51, pp. 386-398, January 2005.