

Source: BenQ mobile¹

Title: Components for TR on Video Minimum Performance Requirements

Document for: Decision

Agenda Item: 9, 13.1

1 Introduction

At SA4#38 a first draft on the TR for Video Codec Performance Requirements has been compiled. The draft still has many technical and procedural white spots despite the process on the generation of the performance requirements has been agreed during SA4#38.

In this document we provide a list of necessary components to be integrated in the TR. This include information on

- the generation of the minimum performance requirements,
- definition of parameters,
- definition of metrics,
- definition of test cases,
- software components for the generation of the MPR
- files to be included in the TR
- initial values on MPRs for different service scenarios

Finally, section 7 contains a list of proposals which could be agreed on individually can be used. BenQ mobile offers to generate and contribute immediate input to all agreed topics even during SA4#39.

For convenience, the document is attached in pdf form as MathType Equations might not display appropriately in Word with a simple equation editor.

Also attached is a quality evaluation tool in ANSI C-code.

¹ Contact: Thomas Stockhammer (stockhammer@nomor.de). BenQ mobile. Munich.

2 File-Based MPR Generation Approach

2.1 Overview

The testing environment is depicted in Figure 1. Importantly, the testing environment to generate the minimum performance figures is setup in a file-based manner from the video source to the video sink. A file is generated between each of the processing units and indicated with flags and numbers. For an individual test the sequences are for example generated in the order 1, 2, 3, 4, 5, and 6. The performance metrics are in general generated by applying files 2, 4, and 6 to a quality assessment tool.

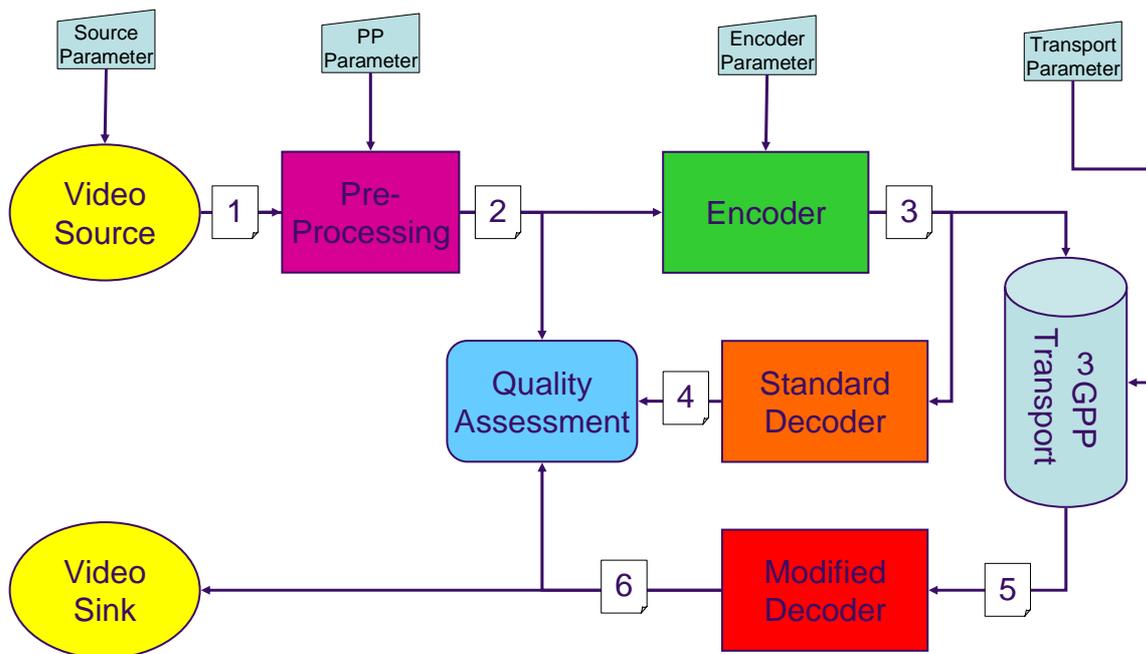


Figure 1 Overview Testing Environment

For the generation of the performance metrics the following steps are carried out:

- An appropriate video sequence 1 is generated according to the selected source parameters, i.e. one of the test video sequences is selected.
- The video sequence 1 is used to generate a video sequence 2 according to the selected preprocessing parameters, i.e. the frame skip is selected.
- An encoded packet stream file 3 is generated by applying video sequence 2 and the appropriate encoding parameters, e.g. service and bitrate.
- The encoded packet stream file 3 is decoded by a standard decoder to generate a reconstructed video sequence 4. No external parameters are applied.
- The encoded packet stream file 3 is also applied to the transport software which generates a corrupted packet stream file 5 based on the 3GPP transport parameter selection, e.g. service, bearer, random seed, etc..
- The corrupted packet stream file 5 is decoded by a modified decoder which handles corrupted packet stream files. No external parameters are applied.
- Video sequence files 2, 4, and 6 are applied to the quality assessment tool to generate performance metrics.

2.2 Parameters

The following provides a list of input parameters available for the each of the components

Source Parameters

As video source, one of the following video sources can be used:

S1. Test Sequence

- 3GPP SA4 Test Sequence 1: stunt_walk_friends_QCIF
- 3GPP SA4 Test Sequence 2: bugs_QCIF
- 3GPP SA4 Test Sequence 3: news_car_QCIF
- 3GPP SA4 Test Sequence 4: news_car_QVGA

For detailed descriptions we refer to S4-050789 (Video Adhoc Group Database – Permanent Document), Section 4.

Pre-processing Parameters

In the pre-processing, the original video sequences are modified by simple signal modifications, usually subsampling is applied. In our case, the only modification is the use of simple temporal subsampling by modifying the frame rate to a frame rate divided by any integer 1, 2, ...

P1. Frameskip = 0, 1, 2, 3, ...

Encoding Parameters

In the video encoding process, the encoder is made aware of the service it is operating in and the expected bitrate.

E1. Service: Service A (MMS-like), Service B (PSC-like), Service C (MBMS-like)

Note: The service defines many encoding parameters taking into account the expected computational resources, buffer considerations, error resilience.

E2. Bitrate: Bitrate in kbit/s

Note: needs to be clarified if needed at all as bearer specifies bitrate. If yes, it is necessary to understand which headers are included.

Transport Parameters

T1. Bearer: The bearer is an integer number 1, 2, 3...and defines a bearer setting in the file bearers.txt

Note: The bearer parameters itself as well as the selected bearers need to be defined.

T2. Random Seed: Integer number 1,2,3,... to allow results with different statistical properties.

Note: the amount of statistical runs need to be specified.

2.3 General Description of Quality Metrics for Individual Run

The following section provides a general description of the quality metrics. The detailed algorithm is provided in the attached c-program. The input to the program are three sequences, an original sequence (file 2), a reconstructed sequence (file 4), and a decoded sequence (file 6). In special cases, the reconstructed and the decoded sequence might be identical. Each of the following metrics provides a single value.

Average Encoding PSNR (AEP)

The average PSNR between the original preprocessed sequence (file 2) and the reconstructed sequence (file 4).

Average Decoding PSNR (ADP)

The average PSNR between the original preprocessed sequence (file 2) and the decoded sequence (file 6).

Standard Deviation of Encoding PSNR (SDEP)

The standard deviation for PSNR between the original preprocessed sequence (file 2) and the reconstructed sequence (file 4).

Standard Deviation of Decoding PSNR (SDDP)

The standard deviation for PSNR between the original preprocessed sequence (file 2) and the decoded sequence (file 6).

Percentage Degraded Video Duration (PDVD)

The percentage of degraded video frames compared between the reconstructed sequence (file 4) and the decoded sequence (file 6).

Standard Deviation of Percentage Degraded Video Duration (SDPDVD)

The standard deviation of the percentage of degraded video frames compared between the reconstructed sequence (file 4) and the decoded sequence (file 6).

PSNR of Encoding Average Normalized Square Difference (PEANSD)

The PSNR of the average squared difference between the original preprocessed sequence (file 2) and the reconstructed sequence (file 4).

PSNR of Decoding Average Normalized Square Difference (PDANSD)

The PSNR of the average squared difference between the original preprocessed sequence (file 2) and the decoded sequence (file 6).

Average Frame Rate (AFR)

The average frame rate of the decoded sequence (file 6).

Standard Deviation of Instantaneous Frame Rate (STIDR)

The standard deviation of the instantaneous frame rate of the decoded sequence (file 6).

2.4 Derivation of Metric for Multiple Runs

For statistical significance, in general multiple runs need to be carried out. This is accomplished by applying different Random seeds in the channel simulator software from 1, ..., N . If the individual metrics as discussed above, which involve file 6, are available for each run $i=1, \dots, N$, then to obtain a single metric, the following equations need to be applied.

Average Decoding PSNR (ADP)

The average decoding PSNR for N runs is defined as:

$$ADP = \frac{1}{N} \sum_{i=1}^N ADP_i$$

Standard Deviation of Decoding PSNR (SDDP)

The standard deviation of the decoding PSNR for N runs is defined as:

$$SDDP = \sqrt{\frac{1}{N} \sum_{i=1}^N (SDDP_i^2 + ADP_i^2) - \left(\frac{1}{N} \sum_{i=1}^N ADP_i \right)^2}$$

Percentage Degraded Video Duration (PDVD)

The percentage of degraded video frames for N runs is defined as:

$$PDVD = \frac{1}{N} \sum_{i=1}^N PDVD_i$$

Standard Deviation of Percentage Degraded Video Duration (SDPDVD)

The standard deviation of the percentage of degraded video frames for N runs is defined as:

$$SDPDVD = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(SDPDVD_i^2 + \frac{PDVD_i^2}{100^2} \right) - \left(\frac{1}{N} \sum_{i=1}^N \frac{PDVD_i}{100} \right)^2}$$

PSNR of Decoding Average Normalized Square Difference (PDANSD)

The PSNR of the average squared difference for N runs is defined as:

$$PANSD = 10 \log_{10} \left(\frac{N}{\sum_{i=1}^N 10^{-0.1PANSD_i}} \right)$$

Average Frame Rate (AFR)

$$AFR = \frac{1}{N} \sum_{i=1}^N AFR_i$$

Standard Deviation of Instantaneous Frame Rate (SDIFR)

The standard deviation of the instantaneous frame rate of the decoded sequence (file 6).

$$SDIFR = \sqrt{\frac{1}{N} \sum_{i=1}^N (SDIFR_i^2 + AFR_i^2) - \left(\frac{1}{N} \sum_{i=1}^N AFR_i \right)^2}$$

3 Tests

In the following, test conditions for different service scenarios are proposed. The scenarios are not necessarily typical 3GPP transmission conditions, but are situations which might occur and a video codec might have to operate in this environment.

3.1 Service Scenario A

	S1	P1	E1	E2
A1	1	1	MMS	64
A2	1	0	MMS	64
A3	1	0	MMS	128
A4	2	1	MMS	128
A5	4	1	MMS	128
A6	4	1	MMS	256

3.2 Service Scenario B

	S1	P1	E1	E2	T1	T2
B1	1	1	PSC	64-x	1 (0.5%)	1...32
B2	1	1	PSC	64-x	2 (1.0%)	1...32
B3	1	0	PSC	128-x	3 (0.5%)	1...32
B4	1	0	PSC	128-x	4 (1.0%)	1...32
B5	2	1	PSC	128-x	3 (0.5%)	1...32
B6	2	1	PSC	128-x	4 (1.0%)	1...32

3.3 Service Scenario C

	S1	P1	E1	E2	T1	T2
C1	3	1	MBMS	64-x	11 (PP 2sec, OH 10% 1.5%)	1...32
C2	3	1	MBMS	64-x	12 (PP 5sec, OH 20% 10%)	1...32
C3	3	1	MBMS	128-x	13 (PP 2sec, OH 10% 1.5%)	1...32
C4	3	1	MBMS	128-x	14 (PP 5sec, OH 20% 10%)	1...32
C5	4	1	MBMS	256-x	15 (PP 2sec, OH 10% 1.5%)	1...32
C6	4	1	MBMS	256-x	16 (PP 5sec, OH 20% 10%)	1...32

4 MPR Generation Components

4.1 File Formats

Video Sequence File Format

For video sequence format, it is proposed that a simple internal file format is used. The file format contains some header information, the yuv data, and some timing information for each frame. Only YUV 4:2:0 are included.

The following header in ASCII format is proposed:

```
width    = <integer> # width of y component
height   = <integer> # height of luminance component
filesize = <integer> # total file size
vfr      = <0,1>     # Variable frame rate flag
tr       = <double>  # frame rate, if 0 90 kHz timestamp assumed
length   = <integer> # the number of uncoded frames
```

In case of vfr=0, immediately after the line break of the header file all yuv data data follows. The timing is interpreted according to the specified frame rate. In case of vfr=1, each video frame in the uncoded sequence is preceded by a time index coded as an integer being a

multiple of the inverse frame rate. The file format is abbreviated by vff (video file format) in the following.

Note that file 2, 4, and 6 are assumed to be in the video sequence file format.

Note: Modifications to the format are easily done to address copyright statements, etc. The basic functionality should be maintained. A commonly defined file format for uncompressed video might also be applied.

Packet Stream Interim File Format

The video encoder is assumed to produce RTP packets which are encapsulated into an interim packet stream interim file format (PSIFF). The resulting packet stream is stored in the PSIFF and can be processed by the 3GPP transport software. The PSIFF is such that each RTP packet is preceded by two additional fields as shown in Figure 2. The fields include the size of the RTP packet including payload and header in bytes and a time stamp (time in ms). This time stamp indicates the availability of the packet to the next processing unit. This file format allows the encoder to signal to the 3GPP transport software when the packet is available and also the 3GPP transport software when the packet is available to the decoder. The PSIFF might also be modified in a sense that entire RTP packets including their PSIFF header format are missing.

It is also assumed that sequence and picture parameter sets are included in the first two encapsulated fields and that these two fields are always present and not dropped.

File 3 and 5 are assumed to be in the PSIFF.

packet size	time stamp	RTP Header	RTP Payload
-------------	------------	------------	-------------

Figure 2. Packet Stream Interim File Format. The RTP packets are encapsulated in an PSIFF using two additional header fields. Packet size: size of the RTP packet including payload and header. Time stamp: time (in ms) at which the packet is transmitted or received.

4.2 Quality Assessment Tool

General Overview

Quality assessment is carried out by applying file 2, 4, and 6 to the quality assessment tool.

Software

The quality assessment tool is attached to this contribution in the file QualAssessment.c. It is proposed that this tool should be used for the assessment. A detailed mathematical description of the metrics seems to be quite difficult, therefore a program is provided.

Subjective Viewing Tool

The authors offer to provide a subjective viewing tool for SA4 internal viewing processes. Based on the experience with such a tool, it might be considered to include this tool or something similar into the TR. The tool plays sequences according to the defined Video Sequence File format. Please issue a request to the author of this document if you are interested in the tool.

4.3 Encoder

General Overview

tbd.

Software

For the encoder the publicly available Nokia software is used.

tbd

Note: It needs to be defined whether software is integrated or not. Otherwise only a reference to the encoder software is sufficient to explain how the anchors have been generated.

Parameter Settings for Service Scenario A

The settings as those proposed in S4-050722 are to be used.

Parameter Settings for Service Scenario B

The settings as those proposed in S4-050722 and S4-060080 are to be used.

Parameter Settings for Service Scenario C

The settings as those proposed in S4-050722 and S4-060093 are to be used.

4.4 Standard Decoder

A standard-compliant decoder being supplied with a PSIFF stream according to the above conventions has been used. The output of the decoder was such that a video sequence format file format with appropriate header and timing information is generated.

4.5 Modified Decoder

A standard-compliant decoder being supplied with a PSIFF stream according to the above conventions has been used. In addition the decoder was able to handle lost packets, i.e. PSIFF streams with missing encapsulated RTP packets. The output of the decoder was such that appropriate timing information is assigned to the video frames. The following error concealment was used:

- Option 1: freeze frame
- Option 2: previous frame concealment

A modified version of the JM reference software could be included into the TR.

4.6 3GPP Transport Tool

General Overview

For a general description of the software tool see S4-050560.

Parameters

The parameters to the transport tool are:

- input file in PSIFF format as indicated by file 3 in Figure 1.
- output file in PSIFF format as indicated by file 5 in Figure 1.
- bearer

- Random Seed

Bearers

```
# This file contains some bearer configuration. The bearers can be indexed by the number.
# The specific columns are explained in the following
# Number:      Number of the bearer used as index (integer)
# File:        File name of the error masks, can be bit errors or packet errors
# Format:      Gives the format of the file (binary for bit errors, ascii for packet errors)
# TTI:        Transmission Time Interval in ms
# RFS:        Radio Frame Size in bytes describes the RLC-PDU size
# note that 8*RFS/TTI results in the bit rate in kbit/s
# Mode:        Transmission Mode: UACK is unacknowledged bearer, ACKP is acknowledged bearer with persistent mode, ACKN non-
persistent, FECK is unacknowledged and FEC
# System:     CDMA2000, UMTS, GPRS, EGPRS, main difference is in sizes of fields added for headers
# CRUIH:     Compressed RTP/IP/UDP header size assuming header compression
# RDel:      (only for ACK mode) The retransmission delay before it is available at the encoder in multiples of the TTI
# NoRet:     (only non-persistent ACK mode ACKN) Number of Retransmission for ACK mode
# N:         (only for FECK mode) expresses the maximum number of encoding symbols
# del:       (only for FECK mode) expresses the maximum delay in ms
# K:         (only for FECK mode) expresses the maximum number of source symbols
# overhead:  (only for FECK mode) expresses the overhead in %
# T:         (only for FEC mode) expresses the symbol length of the code
# G:         (only for FEC mode) expresses the number of symbols per packet
# MFS:      (only for FEC mode) expresses the maximum fragment size
#
#
# The following bearers are defined
# Number File          Format   TTI      RFS      Mode System CRUIH  RDel/N/del      Amod/K/Overhead T G MFS
# PSC Bearers
# 64 kbit/s
1   PSC_64kbps_20ms_BLER_0_5.txt      ascii  20      160      UACK  UMTS   4
2   PSC_64kbps_20ms_BLER_1_0.txt      ascii  20      160      UACK  UMTS   4
# 128 kbit/s
3   PSC_128kbps_20ms_BLER_0_5.txt     ascii  20      320      UACK  UMTS   4
4   PSC_128kbps_20ms_BLER_1_0.txt     ascii  20      320      UACK  UMTS   4
# MBMS UTRAN BEARERS
# 64 kbit/s
11  MBMS_64kbps_80ms_BLER_1_5.txt     ascii  80      640      FECD  UMTS   8      2000  10  32  16  600
12  MBMS_64kbps_80ms_BLER_10_1.txt    ascii  80      640      FECD  UMTS   8      5000  20  32  16  600
# 128 kbit/s
13  MBMS_128kbps_80ms_BLER_1_5.txt     ascii  80      1280     FECD  UMTS   8      2000  10  64  16  1200
14  10                                  iid    80      1280     FECD  UMTS   8      5000  20  64  16  1200
# 256 kbit/s
15  1.5                                  iid    40      1280     FECD  UMTS   8      2000  10  64  16  1200
16  10                                  iid    40      1280     FECD  UMTS   8      5000  10  64  16  1200
```

Service Scenario B Configuration

The following configuration for the channel simulator shall be used:

```
RTPinfile      = user defined
RTPoutfile     = user defined
LogFile        = user defined
StatFile       = user defined
Bearer         = see below
RandomSeed     = 1-64
ErrorFreeRTP   = 4
TSMoDeSender   = 0 # 0 use TS
MaxSendingDelay = 0 # 0 ignore TS
MaxE2EDelay    = 500 # 0 ignore TS, > 0 drop packet at receiver if delayed
```

With this configuration all packets arriving later than 500 ms than they were generated dropped before the decoder. The first four RTP packets were handed to the decoder directly. This includes the SPS, PPS, as well as the first intra slice. These settings basically correspond to the settings as proposed in S4-050603.

Usage of simulator:

```
sa4sim -f psc.cfg -p RTPinfile=<user defined> -p Bearer=<1-10> -p RandomSeed=<1-64>
```

Service Scenario C Configuration

For service scenario C the following configuration for the channel simulator shall be used:

```
RTPinfile      = user defined
RTPoutfile     = user defined
LogFile        = user defined
StatFile       = user defined
Bearer         = see below
RandomSeed     = 1-64
ErrorFreeRTP   = 4
TSMoDeSender   = 0 # 0 use TS
MaxSendingDelay = 5000 # 0 ignore TS
MaxE2EDelay    = 10000 # 0 ignore TS, > 0 drop packet at receiver if delayed
```

With this configuration all packets arriving later than 5000ms than they were generated are dropped at the transmitter and all packets arriving later than 10000ms before the decoder. The first four RTP packets were handed to the decoder directly. This includes the SPS, PPS, as well as the first intra slice. These settings basically correspond to the settings as proposed in S4-050662.

Usage of simulator:

```
sa4sim -f mbms.cfg -p RTPinfile=<user defined> -p Bearer=<11-16> -p RandomSeed=<1-64>
```

Software

The software tool provided in document S4-050560 has been used for the tests.

4.7 Preprocessing Tool

General Overview

The software allows generating a temporally subsampled file.

Parameters

- input file: original file in vff format
- output file: preprocessed file in vff format
- frame skip applied

Software

not yet available.

5 Files and Anchors included in TR

5.1 Overview

To avoid the inclusion of encoder reference software and 3GPP transport software it is proposed that the TR contains different files in the above format to verify the assessment procedure.

5.2 Original Video Sequence File Format 1

Format

It is proposed that a video sequences are provided in a video sequence file format similar to the file format as defined in section 4.1.

Naming Convention

The following naming convention for original files is proposed: <test sequence file>.vff

Files

- stunt_walk_friends_QCIF.vff
- bugs_QCIF.vff
- news_car_QCIF.vff
- news_car_QVGA.vff

Note: Files are not yet available in this format.

5.3 Pre-processed Video Sequence File 2

not necessary – done with pre-processing tool.

5.4 Encoded Packet Stream File 3

Format

For the format the packet stream intermediate file format (PSIFF) is used.

Naming Convention

The following naming convention is proposed:

sa4_vag_mpr_tc<Test Case>.psiff

The only parameter is the test case specified sections 3.1 to 3.3.

Files

- sa4_vag_mpr_tcB1.psiff
- sa4_vag_mpr_tcB2.psiff
- sa4_vag_mpr_tcB3.psiff
- sa4_vag_mpr_tcB4.psiff
- sa4_vag_mpr_tcB5.psiff
- sa4_vag_mpr_tcB6.psiff
- sa4_vag_mpr_tcC1.psiff
- sa4_vag_mpr_tcC2.psiff
- sa4_vag_mpr_tcC3.psiff

- sa4_vag_mpr_tcC4.psiff
- sa4_vag_mpr_tcC5.psiff
- sa4_vag_mpr_tcC6.psiff

5.5 Reconstructed Video Sequence File 4

The reconstructed video file is not necessary as it is generated by the application of the decoder to one of the appropriate encoded packet stream file 3.

5.6 Corrupted Packet Stream File 5

Note: It needs to be clarified if this section is necessary.

Format

For the format the packet stream intermediate file format (PSIFF) is used.

Naming Convention

The following naming convention is proposed:

sa4_vag_mpr_tc<Test Case>_r<Random Seed>.psiff

The parameter are the test case specified sections 3.1 to 3.3 and the random seed applied in the software.

Files

- sa4_vag_mpr_tcB1_r1.psiff
- etc.

5.7 Decoded Video Sequence File 6

The decoded video files are not necessary as they are generated by the application of the decoder to one of the appropriate encoded packet stream file 3.

6 Minimum Performance Requirements

The following tables provide the preliminary minimum performance requirements for the

6.1 Service Scenario A

	AEP	SDEP	PEANSD
A1	30.22	3.08	29.34
A2	28.34	3.15	27.44
A3	31.55	3.29	30.57
A4	32.20	7.87	27.81
A5	31.37	3.70	29.82
A6	34.09	3.60	32.57

6.2 Service Scenario B

Details on the generation of the results can be found in contribution S4-060263.

	AEP	ADP	SDEP	SDDP	PDVD	PEANSD	PDANSD	AFR	STIFR
B1	29.79	24.58	3.33	5.85	47.71	28.76	21.12	7.22	0.73
B2	29.79	21.61	3.33	5.52	60.62	28.76	19.03	6.94	1.02
B3	30.91	27.57	3.27	5.22	32.99	29.92	24.19	14.71	1.02
B4	30.91	24.91	3.27	5.49	55.22	29.92	21.80	14.43	1.43
B5	32.06	29.62	7.70	6.98	31.08	27.72	25.34	12.23	0.92
B6	32.06	27.78	7.70	6.48	40.22	27.72	23.73	11.95	1.30

6.3 Service Scenario C

Details on the generation of the results can be found in contribution S4-060264.

	ADP	SDDP	PDANS	PDVD	AFR	STIFR
C1	30.97	5.00	24.17	8.78	13.89	1.60
C2	28.06	7.25	20.21	29.66	12.52	2.42
C3	34.57	3.90	31.38	3.69	14.84	0.73
C4	31.64	6.40	23.60	26.33	13.59	2.11
C5	33.53	3.98	31.05	1.845	14.96	0.45
C6	28.64	7.54	20.49	36.89	12.93	2.71

7 Proposals

For the related WI, it is proposed to

1. adopt the concept of the file-based testing procedure to generate the minimum performance figures as explained in section 2.1,
2. include figure 1 as well as the text in 2.1 in the draft TR in section 7,
3. adopt the parameters according to section 2.2,
4. specify the bitrate of the video sequence including RTP/UDP/IPv4 headers,
5. adopt the computation of metrics for a single run by using the attached software,
6. adopt the computation a single value averaged over all runs by using the procedure proposed in section 2.4,
7. select a reasonable amount of metrics which express the subjective quality appropriately, namely AEP, ADP, PEANS, PDANS, PDVD, AFR
8. use the test cases as defined in section 3,
9. include Scenario C in the TR,
10. use $N=32$ different random seeds for each lossy transmission case,
11. adopt a common internal video sequence file format which addresses timing similar or identical to the format proposed in section 4.1,
12. adopt a common packet switched interim file format to encapsulate RTP streams in files identical or similar to the one proposed in section 4.1,
13. apply the quality assessment tool for the generation of the performance metrics for each individual run,
14. make section 4.3 informative in a sense that only pointers to the code are provided, but the code is not a component in the TR,
15. include a modified JM reference software decoder into the TR which fulfils the software description in sections 4.4 and 4.5,
16. adopt previous frame concealment as concealment technique,
17. include 3GPP transport software according to S4-050560 with the updates on the bearers into the TR,
18. adopt the 3GPP transport software configurations for service scenario B as proposed in section 4.6 to the TR,
19. adopt the 3GPP transport software configurations for service scenario C as proposed in section 4.6 to the TR,

20. adopt the integration of a simple preprocessing tool to the TR to allow temporal subsampling of original files
21. include the original video sequences with the format and naming convention as proposed in section 5.2,
22. include the encoded packet streams with the format and naming convention as proposed in section 5.4,
23. not include corrupted video sequences as they can be generated by the application of the 3GPP transport software tool and the encoded video streams,
24. use precision of two digits after decimal separator,
25. use the results in section 6 as preliminary numbers for the Video Codec Performance Requirements in the TR.

BenQ mobile offers to generate and contribute immediate input to all agreed topics even during SA4#39.