Source: **TSG SA WG4 Codec**

Title: **Report of MBMS FEC Status in SA4**

Document for: **Information and action**

Agenda Item: **7.4.3**

1. Introduction

During SA#27 (Tokyo), the following decision has been made related to SA4's MBMS FEC (Forward Error Correction) selection work.

Extract from the official draft SA#27 meeting report:

> **"It was agreed that there will be only a single FEC Code for Rel-6 and will be specified as mandatory.**
>
> ….
>
> **SA WG4 were asked to provide a clear status report to TSG SA#28 to help with any discussions**."

This document reports the progress since SA#27 in this matter.

2. Motivation for FEC and historical background pre SA#27

Tdoc SP-050088[1] ("Report of FEC selection for MBMS" from SA4,) summarised the status of SA4's FEC discussions before SA Plenary#27. At that time there were two remaining candidate FEC codes, Raptor codes and a proposal based on Reed-Solomon codes.

3. FEC Proposals and Selection

SA4's PSM ad hoc meeting in Sophia Antipolis (6-8 April 2005) was almost exclusively devoted to identify appropriate testing conditions. This was done with the appreciated help of RAN4 during a joint session. Liaisons were also exchanged with GERAN on the simulation assumptions for that case.

The resulting simulation assumptions, as presented in Annex 1 and 2 were agreed by the ad hoc meeting and endorsed later by SA4#35. Proponents were encouraged to provide results according to these conditions.

Immediately before SA4#35, one proponent withdrew the Reed-Solomon proposal: *"For the sole purpose of facilitating the progress and finalization of MBMS in SA4 (and at 3GPP level), Nokia decided to simplify the FEC code selection process to the extent that at this meeting, a final decision will be taken on the selected FEC code for MBMS. To realize this goal, Nokia decided to withdraw the FEC code candidate (Reed-Solomon) so far supported. This includes the withdrawal of the document S4-050265 (Reed-Solomon code specification for MBMS download and streaming services), and includes no submission for additional simulation results on Reed-Solomon FEC. "*. (Further details are given in Tdoc S4-050321[2].)

[1] ftp://ftp.3gpp.org/tsg_sa/TSG_SA/TSGS_27/Docs/ZIP/SP-050088.zip

[2] ftp://ftp.3gpp.org/tsg_sa/WG4_CODEC/TSGS4_35/Docs/S4-050321.zip

SA4 has considered the detailed simulation results according to the agreed assumptions, for both candidate codes. Results for the selected Raptor codes are summarised in Annexes 3 to 6. Results from different companies have been cross-verified and in particular confirm the suitability of Raptor codes for MBMS across the full range of conditions tested. Object code verifying the performance of the encoding and decoding algorithms have been made available to SA4. Other contributions on computational complexity and FEC selection criteria have also been considered by SA4.

Radio layer simulation results were provided by RAN4 for various operating points for MBMS bearers at various bit-rates and for different user positions within the cell. These verify the relevance of the operating points considered by the SA4 simulations.

Based on the above, SA4 has agreed to select Raptor codes as the FEC code for MBMS file delivery and streaming in Release 6. A CR to TS 26.346 (Tdoc S4-050378) was agreed and, in SA4's view, contains all changes necessary to support Raptor codes.

4. Raptor Codes Characteristics

Raptor Forward Error correction codes are systematic erasure codes. The same code is proposed for both download and streaming cases and the specific code proposed has been available and stable since May 2004.

The key aspects of Raptor codes are summarised as follows:

Flexibility:

> Raptor codes can efficiently support all presently identified MBMS requirements (see e.g. TS.22.246 and S4-040348[3]) in terms of bearer rates, file sizes, loss rates, packet sizes, packet size variability and protection period for streaming without negative tradeoffs in terms of other parameters or computational complexity. This is true for all loss rates. Code performance in all cases is close to the ideal code.

Resource usages:

> *Download:* For the simulation assumptions agreed by SA4 (See Annex 1 and 2)**,** Raptor codes required transmission resources within 1% of the theoretical minimum[4]. (See Annex 3 and 5)
>
> *Streaming:* For the simulation assumptions agreed by SA4 (See Annex 1 and 2), Raptor codes support media rates at the agreed target reliability level which are typically within 1% and at worst within 2.5% of the theoretical maximum rate. (See Annex 4 and 6.)

Computational complexity:

> Raptor codes have very low complexity, allowing large blocks of data to be decoded over a short time period whilst maintaining a low CPU load. Raptor maintains the same complexity independent of loss. (See Annex 7)

Memory consumption for file download:

> The working memory requirement for Raptor codes is at most 512KB for all file sizes.

Latency:

> Raptor codes maintain the order of sending of source packets for streaming so that no additional encoding latency is introduced and so that no additional delay is introduced at decoders when a stream is joined within a protection period (under no losses or low losses).
>
> The decoding delay for Raptor codes can efficiently be set as a small fraction (<10%) of

---

[3] ftp://ftp.3gpp.org/tsg_sa/WG4_CODEC/TSGS4_31/Docs/S4-040348.zip

[4] In one extreme case of a 3MB file transmitted with small packets on a high-loss GERAN channel the difference was 1.8%.

the protection period with low CPU load during decoding even in worst case loss conditions. Hence Raptor is able to efficiently support any desired overall tune-in delay (=protection period +decoding delay).

The duration of the protection period is operator selectable. The suitability of the Raptor codes has been verified by SA4 over a protection period range from 5s to 20s, but values outside this range may also be applicable.

5. Action requested from TSG-SA

SA4 recommends the adoption of Raptor as the only mandatory FEC for MBMS.

SA4 asks TSG-SA to approve the corresponding CR in Tdoc **SP-050250** (S4-050378).

Annex 1: Simulation Conditions and assumptions (UTRAN)

The full simulation conditions and the format for the report of the simulation results are available in Tdoc S4-AHP247[5]. The following table is copied verbatim from said document and summarizes the simulation conditions.

UTRAN Download		
	Bearer rates	64kbit/s, 128kbit/s, 256kbit/s
	RLC PDU size	640 bytes, 1280 bytes, 1280 bytes respectively
	RLC BLER	1%, 5%, 10% (required), 15%, 20%, 30% (optional)
	RLC block loss pattern	Independent random loss
	Number of trials	At least 10,000 for files <= 512KB, 3,000 for 3072KB
	File sizes	50KB, 512KB, 3072KB
	FLUTE payload size	456 bytes
	ROHC	No
	IPv4/UDP header	28 bytes
	FLUTE header	16 bytes
	FEC overhead	Varied in steps of X packets, where X=ceil(0.005N) and N is the number of packets containing source data
UTRAN Streaming		
	Bearer rates	64kbit/s, 128kbit/s and 256kbit/s
	RLC PDU size	640 bytes (for 64kbit/s bearer) 1280 bytes (for 128kbit/s bearer) 1280 bytes (for 256kbit/s bearer)
	RLC BLER	1%, 5%, 10% (required), 15%, 20%, 30% (optional)
	RLC block loss pattern	Independent random loss
	Simulation duration	24 hours
	Media rates	Varied by steps of 1% of bearer rate, assuming only a single media stream[1]
	FEC overhead	Varied to sum FEC and Media to equal bearer rate
	Source packet RTP payload size	64 kbit/s: 456 bytes 128 kbit/s: 456 bytes 256 kbit/s: 768 bytes

[5] ftp://ftp.3gpp.org/tsg_sa/WG4_CODEC/Ad-hoc_PSM/Docs/S4-AHP247.zip

	Repair packet RTP payload size	Minimum value supported by the FEC code which is not less than 470 (for 64kbit/s and 128kbit/s) and 782 (for 256kbit/s)[2]
	Protection period	5s, 20s
	ROHC	No
	IPv4/UDP/RTP header	40
	Source packet FEC payload ID	Specified by FEC code
	Repair packet FEC payload ID	Specified by FEC code
	FEC Symbol size	Specified by FEC code

Note:

1 In practice, multiple media streams may be carried within a single MBMS bearer. However, only a single media stream is considered for FEC simulation purposes for simplicity.

2 The last repair packet of a block may be shorter if supported by the FEC code in order to fit within the protection period

Annex 2: Simulation Conditions and Assumptions (GERAN)

The full simulation conditions and the format for the report of the simulation results are available in Tdoc S4-AHP252[6]. The following table is copied verbatim from said document and summarizes the simulation conditions.

GERAN Download		
	Bearer rates	28.8kbit/s, 59.2kbit/s, 118.4kbit/s
	RLC PDU size	36 bytes, 74 bytes, 74 bytes respectively
	RLC BLER	for 28.8 kbit/s 0.1% for 59.2kbit/s : 0.5% for 118 kbit/s: 1%, 10%
	RLC block loss pattern	Independent random loss
	Number of trials	At least 10,000 for files <= 512KB, 3,000 for 3072KB
	File sizes	50KB, 512KB, 3072KB
	FLUTE payload size	456 bytes, (for 10% BLER at 118.4 kbps also simulate a case with 146 bytes payload)
	ROHC	No
	SNDCP/LLC/IPv4/UDP header	38 bytes
	FLUTE header	16 bytes
	FEC overhead	Varied in steps of X packets, where X=ceil(0.005N) and N is the number of packets containing source data
GERAN Streaming		
	Bearer rates	28.8kbit/s, 59.2kbit/s and 118.2 kbit/s
	RLC PDU size	36 bytes, 74 bytes, 74 bytes, respectively
	RLC BLER	for 28.8 kbit/s 0.1% for 59.2kbit/s : 0.5% for 118 kbit/s: 1%, 10%
	RLC block loss pattern	Independent random loss
	Simulation duration	24 hours
	Media rates	Varied by steps of 1% of bearer rate, assuming only a single media stream[1]
	FEC overhead	Varied to sum FEC and Media to equal bearer rate
	Source packet RTP payload size	456 bytes (for 10% BLER at 118.4 kbps also simulate a case with 146 bytes payload)
	Repair packet RTP payload size	Minimum value supported by the FEC code which is not less than 470 bytes (for 10% BLER at 118.4 kbps also simulate a case with 160 bytes payload) [2]
	Protection period	5s, 20s

[6] ftp://ftp.3gpp.org/tsg_sa/WG4_CODEC/Ad-hoc_PSM/Docs/S4-AHP252.zip

	ROHC	No
	SNDCP/LLC/IPv4/UDP/RTP header	50
	Source packet FEC payload ID	Specified by FEC code
	Repair packet FEC payload ID	Specified by FEC code
	FEC Symbol size	Specified by FEC code

Note:

1. In practice, multiple media streams may be carried within a single MBMS bearer. However, only a single media stream is considered for FEC simulation purposes for simplicity.
2. The last repair packet of a block may be shorter if supported by the FEC code in order to fit within the protection period

Annex 3: Simulation Results: UTRAN Download

FEC Overhead required for 99% probability of recovery at specific BLER points

64kbit/s

Error rates	Power required (G=-3dB[1])	Power required (G=-6dB[2])	File size	Ideal (%)	Raptor (%)
Low (1% BLER)	2.0%	4.5%	Small (50KB)	7.0	8.0
			Medium (512KB)	3.3	3.6
			Large (3072KB)	2.4	2.6
Medium (5% BLER)	1.8%	3.9%	Small (50KB)	21.8	22
			Medium (512KB)	13.0	13.4
			Large (3072KB)	11.0	11.2
High (10% BLER)	1.7%	3.7%	Small (50KB)	39.0	39.0
			Medium (512KB)	25.8	26.0
			Large (3072KB)	22.6	22.8
15% BLER	n/a	n/a	Small (50KB)	56.0	56.0
			Medium (512KB)	40.5	41.0
			Large (3072KB)	36.0	37.0
20% BLER	n/a	n/a	Small (50KB)	76.0	76.0
			Medium (512KB)	57.0	57.0
			Large (3072KB)	52.0	52.0
30% BLER	n/a	n/a	Small (50KB)	130.0	130.0
			Medium (512KB)	100.0	100.0
			Large (3072KB)	92.0	92.0

Notes:

1 This corresponds to ~90% of users assuming uniform distribution of users

2 This corresponds to ~99% of users assuming uniform distribution of users

FEC Overhead required for 99% probability of recovery at specific BLER points 128kbit/s and 256kbit/s

Error rates	Power required (G=-3dB[1]) 128 kbit/s 256 kbit/s	Power required (G=-6dB[2]) 128 kbit/s 256 kbit/s	File size	Ideal (%)	Raptor (%)
Low (1% BLER)	4.0% 7.9%	8.9% 19.3%	Small (50KB)	7.5	8.0
			Medium (512KB)	3.1	3.4
			Large (3072KB)	2.1	2.2
Medium (5% BLER)	3.7% 7.1%	7.8% 16.0%	Small (50KB)	20.0	21.0
			Medium (512KB)	11.2	11.4
			Large (3072KB)	8.8	9.0
High (10% BLER)	3.4% 6.8%	7.2% 14.8%	Small (50KB)	35.0	35.0
			Medium (512KB)	21.5	21.5
			Large (3072KB)	17.8	18.1
15% BLER	n/a	n/a	Small (50KB)	50.0	50.0
			Medium (512KB)	32.0	32.1
			Large (3072KB)	28.0	28.1
20% BLER	n/a	n/a	Small (50KB)	66.0	66.0
			Medium (512KB)	44.6	45.0
			Large (3072KB)	38.0	38.2
30% BLER	n/a	n/a	Small (50KB)	106.0	106.0
			Medium (512KB)	72.0	72.0
			Large (3072KB)	66.8	67.0

Annex 4: Simulation Results UTRAN Streaming

Maximum supported Media Rate for Mean Time Between FEC Block Loss of 1 hour

Error rates	Bearer rate	Ideal	Raptor
Low (1% BLER)	Low (64kbit/s)	5s: 56.8 20s: 60.6	5s: 55.8 20s: 60.4
	Medium (128kbit/s)	5s: 116.3 20s: 122.6	5s: 115.5 20s: 122.4
	High (256kbit/s)	5s: 237.4 20s: 246.4	5s: 236.2 20s: 245.7
Medium (5% BLER)	Low (64kbit/s)	5s: 47.4 20s: 54.2	5s: 46.6 20s: 53.6
	Medium (128kbit/s)	5s: 102.2 20s: 112.5	5s: 100.8 20s: 111.8
	High (256kbit/s)	5s: 228.0 20s: 224.5	5s: 227.0 20s: 224.0
High (10% BLER)	Low (64kbit/s)	5s: 39.5 20s: 47.5	5s: 38.5 20s: 47.2
	Medium (128kbit/s)	5s: 88.5 20s: 101.8	5s: 87.5 20s: 101.2
	High (256kbit/s)	5s: 182.0 20s: 201.5	5s: 179.5 20s: 200.5

Annex 5: Simulation Results GERAN Download

FEC Overhead required for 99% probability of recovery at specific GERAN operation points

Operation Points	File size	Ideal (%)	Raptor (%)
Low Bitrate (CS3)	Small (50KB)	4.7	5.3
0.1% BLER	Medium (512KB)	2.4	2.7
28.8 kbit/s	Large (3072KB)	1.8	2.1
Medium Bitrate (MCS-6)	Small (50KB)	9.4	9.7
0.5% BLER	Medium (512KB)	5.6	5.9
59.2 kbit/s	Large (3072KB)	4.6	4.7
High Bitrate (MCS-9)	Small (50KB)	16.0	16.0
1% BLER	Medium (512KB)	10.6	10.8
118.4 kbit/s	Large (3072KB)	9.2	9.3
High Bitrate and High Error Rate (MCS-6)	Small (50KB)	60	60
10% BLER, 146 byte packet payload	Medium (512KB)	51.5	51.5
118.4 kbit/s	Large (3072KB)	49.2	52.0

* 440 byte payloads, according to Raptor specification recommendations.

Annex 6: Simulation Results GERAN streaming

Maximum supported Media Rate for Mean Time Between FEC Block Loss of 1 hour

Operation Points	**Ideal**	**Raptor**
Low Bitrate (CS3) (0.1% BLER) 28.8 kbit/s	5s: 24.9 kbit/s 20s: 26.6 kbit/s	5s: 24.4 kbit/s 20s:26.2 kbit/s
Medium Bitrate (MCS-6) (0.5% BLER) 59.2 kbit/s	5s: 51.1kbit/s 20s: 54.6 kbit/s	5s: 50.4 kbit/s 20s: 54.2 kbit/s
High Bitrate (MCS-9) (1% BLER) 118.4 kbit/s	5s: 99.5 kbit/s 20s: 104.5 kbit/s	5s: 98.2 kbit/s 20s: 104.2 kbit/s
High Bitrate and High Error Rate (MCS-6) 10% BLER, 146 byte packet payloads 118.4 kbit/s	5s: 66.5 kbit/s 20s: 72.5kbit/s	5s: 66.4 kbit/s 20s: 72.3 kbit/s

Annex 7: Complexity Analysis

1. Analysis approach

The traditional approach for analysis of computational complexity, for example of audio codecs, has been wMOPS or ‘weighted Millions of Operations Per Second’. This is based on determining the processor operations required to perform the encoding or decoding, assigning each operation a weight based on an agreed measure of the relative time taken to perform each type of operation (e.g. addition, multiplications etc.).

This approach focussed on the basic operations to be performed by the processor. In practice, these operations require data to operate on. Fetching such data from memory is in general somewhat slower than performing a single arithmetic operation itself. Therefore the wMOPSs approach assumes that cost of memory operations is accounted for within the weighted operation cost.

In the case of FEC codes, the amount of data generally being processed precludes that it all be held in processor cache – at least on constrained systems such as mobile devices. Thus it could be expected that the principle determining factor of FEC code speed is the number of memory operations required (reads/writes) and the speed of the memory.

Thus, a wMOPS analysis of FEC codes needs to ensure that the cost of memory operations properly accounted for.

Simplistically, each basic operation (for example an XOR) requires three memory accesses – two to fetch the operands and one to store the result. However, depending on the construction of the FEC code, one or both operands may already be available within the processor as a result of an earlier operation. In general we can assume that an arithmetic operation and a memory access (associated with a different operation) can be performed concurrently.

In this analysis we assume that the majority of the computational complexity arises from the operations performed upon the received source and repair symbols in order to generate the lost source symbols. A certain amount of additional processing is required to determine the operations that need to be performed and their sequencing (‘scheduling’). In fact, scheduling is equivalent to calculating the inverse of a matrix. For Raptor codes it is a sparse matrix over GF(2). It is important to note that the construction of the Raptor code matrix is such that the complexity of scheduling is close to linear as the code size increases. However, we will not consider scheduling costs further here and we will also assume that the scheduling information is readily available to the processor (i.e. does not have to be fetched from memory).

2. Weighted operations

The operations required by Raptor FEC codes are bitwise exclusive OR operations. We assume that memory access and standard arithmetic and logical operations (including XOR operations) can be performed 32 bits at a time.

We assign the following weights to these operations and associated memory accesses:

Operation	Weight
32-bit memory read or write	1
32-bit memory read or write + standard arithmetic/logical operation (+,-,AND, OR, XOR)	1

3. Raptor codes Complexity

The Raptor code decoding process uses only XOR operations. The number of XOR operations is in general proportional to the number of symbols, k, however it does vary slightly depending on the number of symbols received. This is because if more symbols are received then these excess symbols can be used in preference to the LDPC and Half symbols for decoding. This requires fewer XOR operations, because the matrix rows corresponding to LDPC and Half symbols are relatively dense.

For streaming applications, however, it is necessary to consider the 'worst case' computational load. This will occur when the received overhead is very low – i.e. the number of received symbols is close to the number of source symbols, k.

With very low overhead, decoding a Raptor code requires approximately 18.9 XOR operations per symbol. Each of these requires at least one of its operands to be read from memory. Approximately 8.2 of these 18.9 require both operands to be read from memory. Finally, approximately 9.6 require the result to be written back to memory. The cost in weighted operations of these is:

- XOR operations (including one memory read): 18.9k
- Additional memory reads: 8.2k
- Memory writes: 9.6k

The total weighted operations per symbol word is therefore: **36.7 k**

Thus the total number of operations required is 36.7k x (T/4), where T is the symbol size in bytes. The file size, or streaming block size, F, is calculated as k·T and so the total operations is **9.175F.**

For example, the number of operations required to decode a 3MB file is just under 29 million. This can easily be achieved on current mobile platforms in a few seconds using a small fraction of the available computational resources.

The number of operations required to decode 5 seconds of a 256kbit/s stream is just under 1.5 million operations. This can easily be achieved on current mobile phone platforms in less than a second using a small fraction of the available computational resources.

It is important to note that the computational load is independent of the packet loss rate.

It is also worth noting that there are many factors which impact the decoding speed for FEC codes, not least the availability of cache and deliberate localisation of memory access in the code implementation (to increase cache hits). Additionally, some modern processors have registers wider than 32-bits, which allows further optimisation of XOR operations.