TSG-RAN Working Group 2 (Radio layer 2 and Radio layer 3) ***TSGR2#7(99)a56***
Malmö, Sweden, 20 - 24 September 1999

TSG-RAN Working Group 1 meeting #7 ***TSGR1#7(99)e36***
Hannover, Germany
August 30 – September 3, 1999

To: TSG RAN WG2

Source: TSG RAN WG1

Title: Liaison statement on TFCI mapping

TSG RAN WG1 would like to inform TSG RAN WG2 about the status of TFCI mapping in layer 1 specifications. WG1 has specified the mapping of TFCI values to code words in [1]. Previously, the mapping of TFC to TFCI was also specified in [1].

For the mapping of TFC to TFCI, all combinations of the different TrCH's TFs were derived. These TFCs were then mapped to TFCI values. Each TFCI value was then assigned a code word. WG1 has identified this solution as inappropriate from a coding perspective since code words are used for all TFCs even if some TFCs are not allowed. This results in worse detection probability than if only the allowed TFCs are assigned code words. WG1 has therefore removed this text from [1].

No new text was inserted since WG1, after discussion, came to the conclusion that this is very closely related to higher layer signalling and therefore better described in WG2 documentation. A proposal for TFCI mapping [2] has been presented and discussed in WG1. This proposal was viewed as interesting but since TFCI mapping is seen as a WG2 issue, WG1 would like to redirect the proposal to WG2. The proposal is enclosed with this liaison statement.

To summarise, there is now no description of mapping of TFC to TFCI in WG1 documentation and WG1 asks WG2 to specify this in their documentation.

[1] TSG RAN WG1, "TS 25.212 Multiplexing and channel coding (FDD)".

[2] Ericsson, "TSGR1#7(99)b33 TFCI mapping".

Attachment: TSGR1#7(99)b33, "TFCI mapping", source Ericsson

TSG-RAN Working Group 1 meeting #7 **TSGR1#7(99)b33**
Hannover, Germany
August 30 – September 3, 1999

Agenda item:
Source: Ericsson
Title: TFCI mapping
Document for: Decision

---

1 Introduction

Up to now, the discussion on TFCI has been around the channel coding of the TFCI and method of transmission. However, an equally important problem is how the different transport format combinations (TFCs) are assigned different TFCI, in an efficient manner. This contribution addresses this problem, proposes a solution and contains a text proposal.

2 Problem

Assume that I transport channels $TrCH_i$, $i = 1, 2, \ldots, I$, are multiplexed, i.e. the transport format combination indicates the transport format of I transport channels. Each $TrCH_i$ has L_i transport formats, i.e. the transport format indication TFI_i can take L_i values, $TFI_i \in \{0,1,2,\ldots,L_i-1\}$.

If all combinations of transport formats are allowed, the number of transport format combinations will be $C = L_1 \times L_2 \times \ldots \times L_I$. This can become a rather significant number, even with only a few transport channels. In reality, only a subset of all the C TFCs are used. For example, assume a UEP AMR speech service with three transport channels for the three protection classes. AMR has 9 different rates (including DTX), so only 9 TFCs are used. However, if we compute C, then we get 9×8×3 = 216 combinations! Similar problems can arise when considering other service combinations.

Allocating Layer 1 signalling (TFCI) for a large number of transport format combinations, many of which are not use, leads to two problems:

- There may not be enough available TFCI words (64 or 1024).

- The performance of the TFCI detection is depending on how many TFCI code words that are in use. There is a significant difference detecting 8 code words out of 64 possible or detecting 64 code words out of 64 possible. Moreover, using the 2×(15, 5) code to handle up to 1024 TFCI code words has much worse performance than the 1×(30, 6) code that handles up to 64 TFCI code words. Hence, from a performance point of view, one should not allocate TFCI for combinations that are not used.

The current TFCI mapping rule defined in TS 25.212 does not take into account that not all transport format combinations are possible. Hence, the allocation used in that algorithm will suffer from wasting the TFCI code words.

From the discussion above, it is obvious that we need to limit ourselves to only signal the used combinations. A proposal that allows this is described in the next section.

3 Proposal

It is proposed that the higher layers signal the used TFCs to Layer 1, so that each allowed TFC can be unambiguously mapped to a certain TFCI. To signal this mapping in an efficient way, it is proposed that the higher layers computes a value we will call CTFC (Calculated TFC). The CTFC is calculated by the following algorithm:

Define $P_i = \prod_{j=0}^{i-1} L_j$, where $i = 1, 2, \ldots, I$, and $L_0 = 1$.

Let $TFC(TFI_1, TFI_2, \ldots, TFI_I)$ be the transport format combination for which $TrCH_1$ has transport format TFI_1, $TrCH_2$ has transport format TFI_2, etc. Now, for any combination of transport formats $TFC(TFI_1,$

TFI$_2$, …, TFI$_I$), we can compute the corresponding CTFC(TFI$_1$, TFI$_2$, …, TFI$_I$).

$$CTFC(TFI_1, TFI_2, \ldots, TFI_I) = \sum_{i=1}^{I} TFI_i \cdot P_i$$

The algorithm delivers a unique CTFC value for all possible combinations of TFI$_1$, TFI$_2$, …, TFI$_I$.
After computing the CTFC value for all allowed transport format combinations, the CTFCs are signalled in order. The TFCIs are assigned in the same order, i.e. the first TFC signalled by its CTFC will correspond to TFCI = 0, the next corresponds to TFCI = 1, etc. In this contribution we do not go into the details of this signalling, since that is a WG2 issue to define.
It is straightforward to calculate the TFIs of a certain TFC's CTFC using the following algorithm (C language):

```
m = CTFC;
i = I;
while (i > 0) {
   TFIi = floor(m / Pi);
   m = m % Pi;
   i = i – 1;
}
```

Another straightforward way of signalling only allowed TFCs, is to signal the TFI for each transport channel for each TFC. However, it can be shown that the number of bits required with the proposed scheme is always less than or equal to the straightforward signalling of the TFIs in the allowed combinations.
The number of bits required to signal one TFC for the straightforward case is A:

$$A = \sum_{i=1}^{I} \lceil \log_2 L_i \rceil$$

For the proposed scheme, the number of bits required to signal a TFC is directly related to the largest possible CTFC value, CTFC$_{max}$.

$$CTFC_{\max} = \sum_{i=1}^{I} (L_i - 1) P_i = \sum_{i=1}^{I} (L_i - 1) \prod_{j=0}^{i-1} L_j = \sum_{i=1}^{I} \left(\prod_{j=0}^{i} L_j - \prod_{j=0}^{i-1} L_j \right) = \prod_{j=0}^{I} L_j - \prod_{j=0}^{0} L_j = \prod_{j=1}^{I} L_j - 1$$

.
Now, the number of bits for the proposed scheme is B:

$$B = \lceil \log_2 CTFC_{\max} \rceil = \left\lceil \log_2 \left(\prod_{j=1}^{I} L_j - 1 \right) \right\rceil \le \left\lceil \log_2 \prod_{j=1}^{I} L_j \right\rceil = \left\lceil \sum_{j=1}^{I} \log_2 L_j \right\rceil \le \sum_{j=1}^{I} \lceil \log_2 L_j \rceil = A$$

.
Hence, we have shown that the number of bits required for the proposed scheme is always less that or equal to what is needed with the straightforward scheme.
It is proposed that the procedure above is adopted for UTRA. It is further propose that the definition of how the CFTC values are calculated is described in TS 25.212, while the details of the signalling is described in RAN WG2 specifications.

3 Example

Assume 3 transport channels, with TFI$_1 \in \{0, 1, 2\}$, TFI$_2 \in \{0, 1, 2\}$, TFI$_3 \in \{0, 1\}$. Further, assume that when TFI$_1$ = 0, any combination of TFI$_2$ and TFI$_3$ is allowed, while when TFI$_1 \neq 0$ then TFI$_2$ and TFI$_3$ must both be 0.
That gives the following:
$P_1 = L_0 = 1$
$P_2 = L_0 \times L_1 = 1 \times 3 = 3$
$P_3 = L_0 \times L_1 \times L_2 = 1 \times 3 \times 3 = 9$
In the table below we list the valid combinations, and compute the CTFC.

TFI_1	TFI_2	TFI_3	CTFC	TFCI
0	0	0	0×1 + 0×3 + 0×9 = 0	0
0	1	0	0×1 + 1×3 + 0×9 = 3	1
0	2	0	0×1 + 2×3 + 0×9 = 6	2
0	0	1	0×1 + 0×3 + 1×9 = 9	3
0	1	1	0×1 + 1×3 + 1×9 = 12	4
0	2	1	0×1 + 2×3 + 1×9 = 15	5
1	0	0	1×1 + 0×3 + 0×9 = 1	6
2	0	0	2×1 + 0×3 + 0×9 = 2	7

As can be seen, each valid combination results in different CTFC. To indicate the allowed combinations, the sequence of CTFCs (0, 3, 6, 9, 12, 15, 1, 2) is signalled to Node B and UE, where each CTFC in order is allocated a TFCI value. From the CTFC both Node B and UE can determine the exact transport format combinations the TFCI values (used to communicate between Node B and UE) represent. In this example, signalling each CTFC requires 4 bits, i.e. the total required signalling is 8×4 = 32 bits. Simply signalling the TFIs of all combinations would require 8×(2+2+1) = 40 bits.

4 Text proposal for TS 25.212 V2.0.1

4.2.12.1 4.2.13.2 Explicit transport format detection based on TFCI

4.2.13.1.1 4.2.13.2.1 Transport format combination indicator (TFCI)

The Transport Format Combination Indicator (TFCI) informs the receiver of the transport format combination of the CCTrCHs.~~number of bits in each frame of each of the services currently in use.~~ As soon as the TFCI is detected, the transport format combination, and hence the individual transport channels' transport formats are known, and decoding of the transport channels can be performed.

The TFCI to use to indicate a certain transport format combination is signalled from higher layers using the Calculated Transport Format Combination (CFTC). The signalled CTFC are unambiguously associated with a certain TFCI value. How the signalling is performed is described in higher layer specifications. How the CTFC is composed is described in the next section.

4.2.13.1.2 4.2.13.2.1 Calculated transport format combination (CTFC)

The Calculated Transport Format Combination (CTFC) is a tool for efficient signalling of transport format combinations to be assigned TFCI values. The CTFC is calculated by higher layers by the algorithm below:

Let I be the number of transport channels that are included in the transport format combination. Each transport channel $TrCH_i$, $i = 1, 2, \ldots, I$, has L_i transport formats, i.e. the transport format indicator TFI_i can take L_i values, $TFI_i \in \{0,1,2,\ldots,L_i-1\}$.

Define $P_i = \prod_{j=0}^{i-1} L_j$, where $i = 1, 2, \ldots, I$, and $L_0 = 1$.

Let $TFC(TFI_1, TFI_2, \ldots, TFI_I)$ be the transport format combination for which $TrCH_1$ has transport format TFI_1, $TrCH_2$ has transport format TFI_2, etc. The corresponding $CTFC(TFI_1, TFI_2, \ldots, TFI_I)$ is then computed as:

$$CTFC(TFI_1, TFI_2, \ldots, TFI_I) = \sum_{i=1}^{I} TFI_i \cdot P_i.$$

~~a certain bit-rate is known, the number of code channels, the spreading factor and the puncturing/repetition rate is immediately known from the rules described in section 4.2.6.~~

~~This document therefore only explains the mapping from TFCI bits to TFCI service rate combinations.~~

~~A connection may in general include the variable rate services~~ ~~$S_1, S_2, \ldots, S_K$. Each service S_i has a set of possible transport format combination indicators $TF_{i,1}, TF_{i,2}, \ldots TF_{i,Li}$:~~

~~S_1: $TF_{1,1}, \ldots, TF_{1,L1}$~~

~~S_2: $TF_{2,1}, \ldots, TF_{2,L2}$~~

~~…~~

~~S_K: $TF_{K,1}$, … , $TF_{K,LK}$~~

~~This gives L=L1xL2x…xLK service rate combinations, and thus it is required that L is less than or equal to 64 with the default TFCI word or 1024 with the extended TFCI word.~~

~~These service rate combinations shall be mapped to a certain service rate combination number, *m*, in the following way:~~

~~For j=K: 1:1,~~

~~SRC[j]= m MOD L[j];~~

~~m = m DIV L[j];~~

~~End;~~

~~From this pseudo-code, given a service rate combination number, i.e. a certain combination of TFCI bits, *m*, SRC contains the rates of each of the K services. The integer values used for *m* shall be consecutive, starting from 0.~~

~~Note that this code gives the mapping rule from *m* to SRC, i.e. the rule used in the receiving side. The mapping rule from SRC to *m*, i.e. the transmitting side rule, is [TBD].~~