

ARQ Considerations For The New GSM/EDGE Flexible Layer One

Kent Pedersen¹, Benoist Sébire², Guillaume Sébire³, Glenn Platt¹

¹Nokia Mobile Phones, Frederikskaj, DK-1790, Copenhagen V, Denmark

²Nokia Research Center, P.O Box 407, 00045 NOKIA GROUP, Finland

Email: kent.pedersen@nokia.com, benoist.sebire@nokia.com, guillaume.sebire@nokia.com,
glenn.platt@nokia.com,

Abstract- A major enhancement of the GSM/EDGE Radio Access Network (GERAN) is the Flexible Layer One (FLO) concept, due to be introduced in Release 6 of the 3GPP standards. FLO greatly simplifies the introduction of new services to the GERAN system, allowing for “tailor-made” provision of any given Quality of Service (QoS), significantly enhancing GERAN’s flexibility, and ensuring it will evolve on par with other 3G systems. In the GERAN Iu mode, FLO supports data transfer in transparent, unacknowledged and acknowledged RLC (Radio Link Control) modes. In the latter mode, the RLC applies backward error correction (BEC) through an ARQ (Automatic Repeat reQuest) mechanism. Considering ARQ mechanisms, Hybrid Type II ARQ (a.k.a. incremental redundancy) is a very efficient technique that requires a close interaction between the retransmission protocol at the RLC layer and the channel encoder/decoder at the physical layer. This paper first introduces the concept of FLO, and then describes how hybrid type II ARQ can be implemented in the FLO physical layer. Finally, we present the performance gains possible by introducing incremental redundancy to FLO, especially when compared to the performance of Chase combining.

Index Terms—FLO, GERAN, hybrid type II ARQ, incremental redundancy

I. INTRODUCTION

Until recently the radio bearers of GERAN (GSM/EDGE Radio Access Network) have been dedicated bearers designed specifically for a given service, such as circuit switched speech services, or generic data bearers having fixed payload sizes, such as the (E)GPRS (General Packet Radio Service) coding schemes. The introduction of the IP Multimedia Subsystem (IMS) in Release 5 of the 3GPP (3rd Generation Partnership Project) standards has seen new requirements placed on the radio bearers of GERAN [1]. The aim of the IMS is to enable the provision of IP multimedia services (conversational, streaming, interactive and background services) to mobile users. The fact that the nature of future IP multimedia services is to a large extent unknown means that the GERAN physical layer needs to evolve towards enabling the fast introduction of new services. To increase the flexibility of the GERAN physical layer, a major new feature is being standardized for Release 6 of the 3GPP standards, to be finalized during 2003: the Flexible Layer One (FLO).

One of the major changes between FLO and GERAN prior to FLO is that with FLO, the particular configuration of the physical layer is negotiated at call setup to suit the Quality of Service (QoS) requirements of the traffic to be carried. Following this, other protocol layers are also configured in order to provide appropriate QoS. For example, and of particular interest to this paper, is the RLC protocol, which can operate in RLC acknowledged mode. RLC acknowledged mode allows for backward error correction of the transmitted data through selective retransmissions of erroneously received data. Hybrid type II ARQ (otherwise known as incremental redundancy - IR) allows for a significant increase in throughput, by combining the retransmission of a given block with its previous transmissions. Hybrid type II ARQ relies on the RLC protocol to control the retransmissions, and on the physical layer to perform the channel encoding and decoding as a function of the retransmissions to be made.

This paper studies the implementation of incremental redundancy to GERAN with FLO. First, we present a general introduction to FLO and IR, followed by a description of how FLO can support IR, and an analysis of the performance gains possible using IR with FLO.

II. INTRODUCING THE FLEXIBLE LAYER ONE

Currently, the GERAN physical layer offers the MAC layer logical channels for carrying data. A fixed set of logical channels is available, each of which has its transport configuration predefined and optimized for the type of traffic to be carried. Rather than logical channels with a fixed transport configuration, with FLO the GERAN physical layer offers *transport channels* to the MAC layer. Exchange of data between the MAC layer and the physical layer on a transport channel is done by means of *transport blocks*. The transport configuration of the transport channels is negotiated at call setup and can be varied during a connection, and thus the transport configurations available at any particular time can be tailored to the traffic to be carried.

The particular configuration (coding rate, CRC size, input block size, etc.) of a transport channel is denoted as the *transport format* of that transport channel. A transport channel can use any of a number of transport formats that make up a *Transport Format Set* (TFS). The network

configures this TFS. To reduce complexity, a limited number of transport formats are allowed per transport channel. The set of valid transport format combinations across all transport channels is referred to as the *Transport Format Combination Set* (TFCS). The network can change the TFCS available at any particular time, thus providing new transport formats and allowing optimal QoS for a particular traffic type.

The FLO physical layer architecture is shown in Fig. 1. Following the data sequence in Fig. 1, it can be observed that after having a CRC sequence attached (for error detection), transport blocks pass through the channel coding function, where they are channel encoded with a rate 1/3 convolutional code for forward error correction [2]. The encoded blocks are then passed to the rate matching function. This function “adjusts” the size of an encoded block on a transport channel, so that when the block from this transport channel is multiplexed with the blocks from all the other concurrent transport channels, the resulting data will fit in one radio packet. Effectively, the rate matching operation adjusts the protection given to data on a particular transport channel by bit puncturing (the removal of bits in the coded sequence) or repetition (the repetition of bits in the coded sequence), thus altering the size of the encoded block, and changing the level of protection given by the channel coding. The level of protection given to data on a particular transport channel is determined by the rate matching attribute, which is assigned by layers above the physical layer. The rate matching attribute

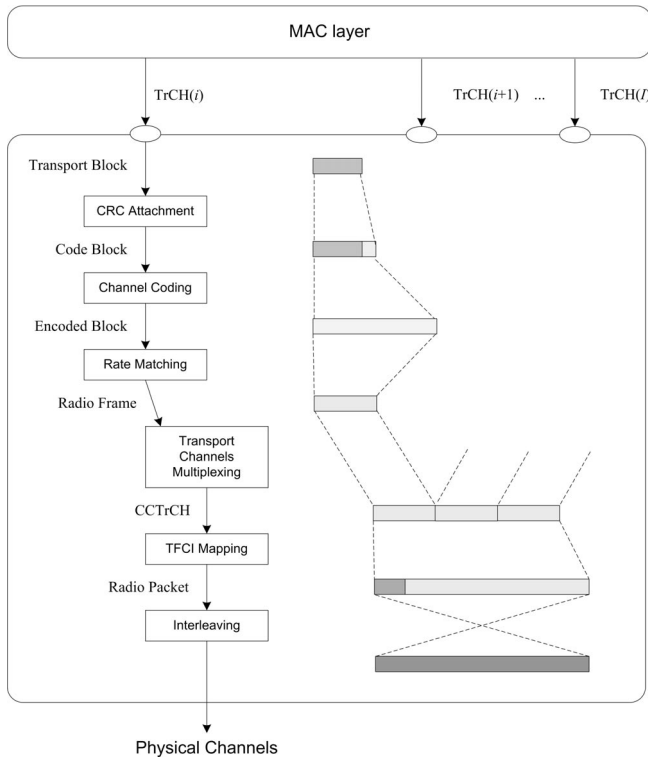


Fig. 1: Architecture of the physical layer of FLO.

controls how much data on a particular transport channel will be punctured or repeated, affecting the level of channel coding protection given to this data, and thus prioritizing between data on different transport channels. Its operation is shown in Fig. 2.

After rate matching, the radio frames (rate-matched data) of all the transport channels are multiplexed by the transport channel multiplexing function, yielding a *Coded Composite Transport Channel* (CCTrCH). This multiplexing operation is a simple serial operation, where data from all the transport channels is combined into one block. Next, a coded *Transport Format Combination Indicator* (TFCI) is appended to the block, in order to indicate to the receiver the active transport channels and their transport formats. The resulting *radio packet* is then interleaved and transmitted. A more detailed description of the physical layer operation and design considerations is provided in [4].

III. PRINCIPLES OF HYBRID TYPE II ARQ

The basic difference between Hybrid Type II ARQ and more traditional ARQ techniques (e.g. Type I ARQ) is that when retransmitting a coded block with Hybrid Type II ARQ, each retransmission is punctured using a different puncturing pattern than the original coded block. Thus, the level of redundancy is increased for each retransmission. Fig. 3 shows the basic principle of incremental redundancy. In the example in the figure, the first transmission is punctured using the pattern (P1). The receiver is unable to decode the block and therefore a 2nd transmission is made using pattern (P2). Upon

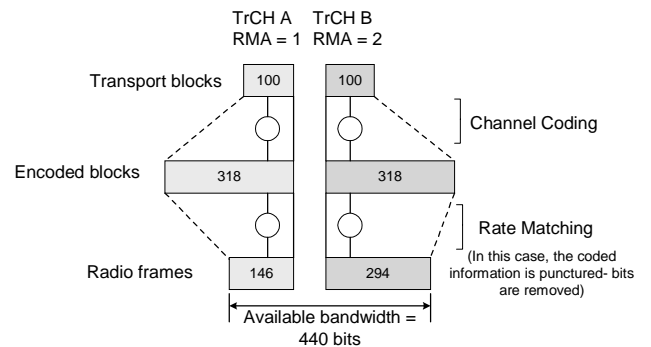


Fig. 2: Rate matching operation in FLO. The diagram shows two transport channels currently active, TrCH A, and TrCH B. The maximum radio packet size is 440 bits, and TrCH A and TrCH B are both to be multiplexed into one radio packet. Both transport channels have the same amount of data arriving at the physical layer- 100 bits. This data is then channel coded, resulting in encoded blocks of the same size (318 bits). TrCH A and TrCH B have different rate matching attributes (RMA's)- effectively prioritizing between them. With a higher RMA than TrCH A, TrCH B is punctured less (has fewer bits removed) than TrCH A, and thus after rate matching has 294 bits in one block, whereas TrCH A is punctured more, and has 146 bits in one block (meaning it has less channel protection than TrCH B). These two blocks are then combined into one radio block and transmitted.

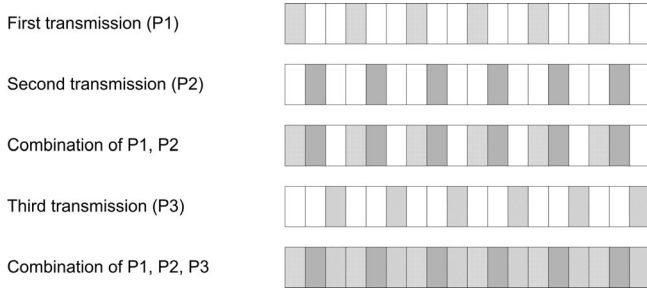


Fig. 3: Example of received bit patterns for 3 transmissions using incremental redundancy.

reception of the 2nd transmission, the receiver combines the two received blocks and attempts to decode this combined block. If the attempt is unsuccessful the receiver requests a 3rd transmission, using pattern (P3). After receiving the 3rd block, the receiver combines this block with the two previous and attempts to decode the resulting block. As shown in the figure, after three retransmissions the information at the receiver is now the equivalent of no puncturing being performed on the coded block. That is, the combination of blocks punctured with (P1), (P2) and (P3) corresponds to no puncturing, since different bits are punctured in each (re)transmission.

Incremental redundancy requires the channel encoder on the transmitter side be able to generate different puncturing patterns achieving the same code rate M/N , based on the same mother code of rate $1/k$, with:

$$1 \geq \frac{M}{N_i} \geq \frac{1}{k} \text{ and } k \in \mathbb{N} + \quad (1)$$

where M is the length in bits of the uncoded block to encode and N_i is the length in bits of the i^{th} -encoded block, as illustrated in Fig. 4.

Evidently, the receiver channel decoder must be able to cope with any of the possible puncturing patterns the channel encoder may generate for the same mother code. It must be able to perform the decoding of the combination of all the encoded blocks received after a number of retransmissions of a given block. The global coding rate (covering the initial transmission and subsequent retransmissions) after combining is adjusted incrementally after each retransmission. After n retransmissions, this global coding rate is:

$$global_coding_rate = \frac{M}{\sum_{i=0}^n N_i - \text{number_of_overlapping_bits}} \quad (2)$$

Where N_0 is the number of bits of the encoded block of the initial transmission, N_1 the number of bits of the encoded block of the 1st retransmission, and so on.

Soft combining of the overlapping bits can improve the

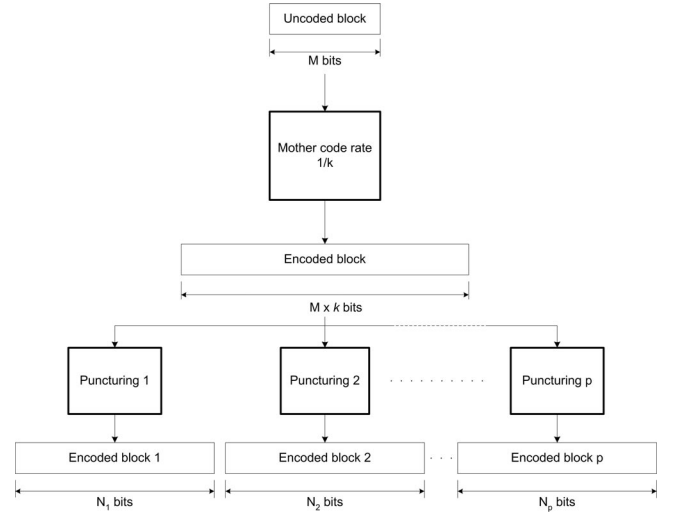


Fig. 4: Encoding at rate M/N with different puncturing patterns.

decoding performance, resulting in the following approximate coding rate:

$$global_coding_rate = \frac{M}{\sum_{i=0}^n N_i} \quad (3)$$

The number of mutually exclusive puncturing patterns achieving a code rate M/N for a mother code of rate $1/k$, an uncoded block of size M bits, and encoded blocks of size N bits is $\frac{M \times k}{N}$.

IV. INCREMENTAL REDUNDANCY FOR FLO

Considering FLO, the rate matching algorithm controls puncturing and repetition of bits in an encoded block. Therefore, if incremental redundancy is to be incorporated in FLO it requires modification of the rate matching algorithm.

A. Rate matching algorithm for FLO without IR

The rate-matching algorithm for FLO is based on similar principles to those used in the rate matching of the UMTS Terrestrial Radio Access Network (UTRAN) physical layer [3]. The rate matching algorithm operates in two steps:

1. Calculate the number of bits to be repeated or punctured for each transport channel.
2. Perform the puncturing/repetition using a pattern generation algorithm.

The number of bits to be repeated or punctured (step 1 above) is calculated from the available bandwidth and the number of active transport channels. A detailed description of the calculation can be found in [2, 4]. The output of the calculation is the parameter $\Delta N_{i,j}$, denoting the number of

bits to be repeated or punctured. If $\Delta N_{i,j}$ is negative, bits should be punctured, and if positive, bits should be repeated. Based on this value, as well as the number of bits in the coded block to be transmitted ($N_{i,j}$), three values are computed that determine the operation of the rate matching algorithm:

e_{ini} - Initial value of variable e in the rate matching pattern generation algorithm.

e_{plus} - Increment of variable e in the rate matching pattern generation algorithm.

e_{minus} - Decrement value of variable e in the rate matching pattern generation algorithm

These values are inputs to step 2 of the rate matching procedure, which computes the rate matching pattern using the following algorithm [2]:

```

if  $\Delta N_{i,j} < 0$ 
   $e = e_{ini}$ 
   $m = 1$ 
  do while  $m \leq N_{i,j}$ 
     $e = e - e_{minus}$ 
    if  $e \leq 0$  then
      puncture bit  $b_{i,m}$ 
       $e = e + e_{plus}$ 
    end if
     $m = m + 1$ 
  end do
else if  $\Delta N_{i,j} > 0$ 
   $e = e_{ini}$ 
   $m = 1$ 
  do while  $m \leq N_{i,j}$ 
     $e = e - e_{minus}$ 
    do while  $e \leq 0$ 
      repeat bit  $b_{i,m}$ 
       $e = e + e_{plus}$ 
    end do
     $m = m + 1$ 
  end do
else do nothing
end if

```

where

$$e_{ini} = 1 \quad (4)$$

$$e_{plus} = 2 \times N_{i,j} \quad (5)$$

$$e_{minus} = 2 \times |\Delta N_{i,j}| \quad (6)$$

The outcome of the algorithm is a puncturing or repetition

pattern, where the bits to be punctured or repeated are uniformly distributed. The parameter e_{ini} determines the starting position of the puncturing pattern, and the parameters e_{plus} and e_{minus} represent the length of the pattern and the number of bits to be punctured or repeated, respectively.

B. Modified rate matching algorithm for FLO

Since the value of e_{ini} determines the starting point of the bits to be punctured, this parameter can be used to generate different puncturing patterns. Table I gives an example of an average distance between punctured bits of 3 ($9/3=3$). In order to generate unique puncturing patterns, it can be seen that the first pattern P1 (obtained with the existing rate matching algorithm) must be shifted *forward* to produce P2 and then P3. If P1 were shifted backward, there would only be 2 punctured bits in P2 because of the way the rate matching operates.

TABLE I: EXAMPLE OF AVERAGE DISTANCE BETWEEN PUNCTURED BITS = 3

Bits	1	2	3	4	5	6	7	8	9
punctured P1	Y	-	-	Y	-	-	Y	-	-
punctured P2	-	Y	-	-	Y	-	-	Y	-
punctured P3	-	-	Y	-	-	Y	-	-	Y

Y – the bit is punctured

Table II gives an example of an average distance between punctured bits of 1.5 ($9/6=1.5$). In order to generate unique puncturing patterns, it can be seen that the first pattern P1 (obtained with the existing rate matching algorithm) must be shifted *backward* to produce P2 and then P3. If P1 were shifted forward, there would only be 2 transmitted bits in P2.

TABLE II: EXAMPLE OF AVERAGE DISTANCE BETWEEN PUNCTURED BITS = 1.5

Bits	1	2	3	4	5	6	7	8	9
punctured P1	Y	Y	-	Y	Y	-	Y	Y	-
punctured P2	Y	-	Y	Y	-	Y	Y	-	Y
punctured P3	-	Y	Y	-	Y	Y	-	Y	Y

Y – the bit is punctured

In order to generate different puncturing patterns, the rate matching algorithm and e_{ini} must therefore take into account the average distance between punctured bits given by:

$$d = \frac{e_{plus}}{e_{minus}} = \frac{N_{i,j}}{|\Delta N_{i,j}|} \quad (7)$$

Generally, when the average distance between punctured bits is greater than or equal to 2, unique puncturing patterns are obtained by shifting the first pattern forward. When the average distance between punctured bits is smaller than 2, unique puncturing patterns are obtained by shifting the first pattern backwards.

The value of e_{ini} is then determined from the equations below:

if $\frac{e_{plus}}{e_{minus}} \geq 2$ then

$$d = \frac{e_{plus}}{e_{minus}} = \frac{N_{i,j}}{|\Delta N_{i,j}|} \quad (8)$$

$$e_{ini} = 1 + (R \bmod \lceil d \rceil) \times e_{minus} \quad (9)$$

else

$$d = \frac{e_{plus}}{e_{plus} - e_{minus}} = \frac{N_{i,j}}{N_{i,j} - |\Delta N_{i,j}|} \quad (10)$$

$$e_{ini} = 1 + (R \bmod \lceil d \rceil) \times (e_{plus} - e_{minus}) \quad (11)$$

The parameter R is a retransmission counter that is signaled to the transmitter. Each time a retransmission is requested, the value of R will increment by one. As seen from the equations above, this variable d indicates the number of different puncturing patterns that can be generated for a given encoded block. For instance, if d is 4, the term $(R \bmod \lceil d \rceil)$ will mean that four different patterns are generated.

Essentially, it can be seen that IR for FLO is based on the FLO rate matching algorithm, which determines the number of bits to be punctured. The parameter R , which increments for every retransmission, controls the particular puncturing pattern generated (by changing e_{ini} , a controlling variable for the rate matching algorithm), ensuring a new puncturing pattern is generated every retransmission. A short example of this operation is given in the following section.

C. Example of Puncturing Pattern Generation

Consider a simple setup where an encoded block of 15 bits is to be transmitted using a single TrCH. The physical channel can carry a payload of 10 bits, which means that 5 bits are to be punctured. Thus, the output of step 1 of the rate matching algorithm will, according to (9), be:

$$\Delta N_{i,j} = -5$$

$$N_{i,j} = 15$$

Recall that $\Delta N_{i,j}$ denotes the number of bits to be punctured and $N_{i,j}$ denotes the number of bits on the TrCH. From this, the two increment and decrement values for the rate matching pattern generation can be calculated as:

$$e_{plus} = 2 \times N_{i,j} = 2 \times 15 = 30$$

$$e_{minus} = 2 \times |\Delta N_{i,j}| = 2 \times 5 = 10$$

Now, since e_{plus} and $e_{minus} = 30/10 = 3$, the value of e_{ini} will be:

$$e_{ini} = 1 + (R \bmod \lceil d \rceil) \times e_{minus} = 1 + (R \bmod \lceil 3 \rceil) \times 10$$

From this it can be seen that three different puncturing patterns can be generated, since $(R \bmod \lceil 3 \rceil)$ can take the values 0,1 and 2. This means that e_{ini} will take the following values for different retransmissions:

TABLE III: PARAMETER VALUES FOR THE EXAMPLE OF PUNCTURING PATTERN GENERATION

Retransmission number	R	$R \bmod \lceil 3 \rceil$	e_{ini}
1 st Initial transmission	R = 0	0	1
1 st Retransmission	R = 1	1	11
2 nd Retransmission	R = 2	2	21
3 rd Retransmission	R = 3	0	1

Fig. 5 shows the puncturing patterns generated for each of the transmissions above. The input to the pattern generation, modeling the coded block, is a ramp function having values in the range from 1 to 15. As can be seen from the figure, the puncturing patterns for the different retransmissions are shifted versions of the first pattern (Trans. 0).

V. CONTROLLING IR FOR FLO

As mentioned previously, the particular transport format (coding rate, CRC size, input block size, etc) of a transport channel is indicated to the receiver using the Transport Format Combination Indicator (TFCI), which is basically a physical layer header. From this, the receiver is able to extract information of the received block and perform decoding accordingly. The use of incremental redundancy for a given service can be signaled to the physical layer by introducing the retransmission parameter R described before, to the TFCI. That is, the network configures a number of transport formats at call setup, corresponding to the number of retransmissions

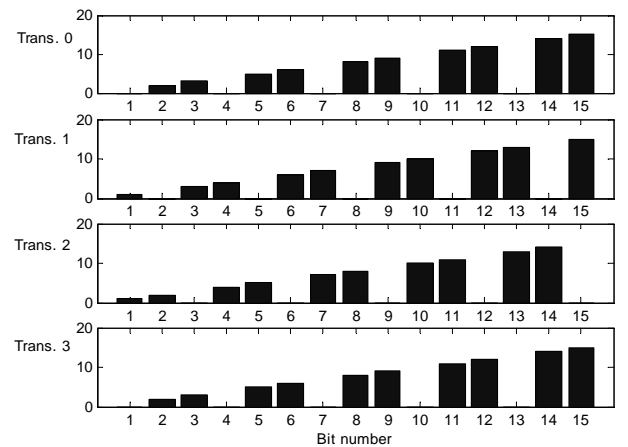


Fig. 5: Puncturing patterns from the first transmission and three retransmissions.

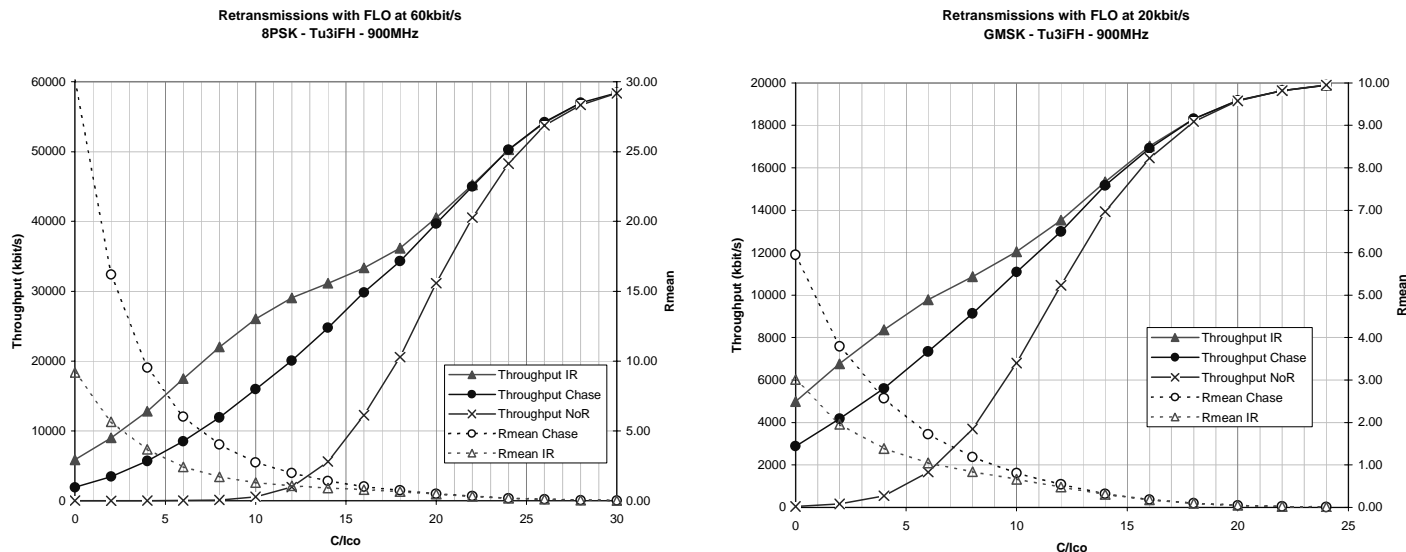


Fig. 6: Throughput when using hybrid type II ARQ, compared to Chase combining, and also the case with no retransmissions at all. Leftmost plot shows the performance when using 8PSK modulation and rightmost plots shows the performance when using GMSK modulation.

that are allowed for one transport block of a given service. Each of these formats will then have a different value of R , and will thus result in different puncturing patterns generated by the rate matching algorithm. The advantage of this approach is that once the service has been initiated, control of incremental redundancy is handled efficiently, with minimum signaling, since the TFCI always has to be decoded.

VI. PERFORMANCE RESULTS

In order to evaluate the performance gain when implementing incremental redundancy in FLO, simulations were run for a typical urban channel, using ideal frequency hopping over 20,000 frames, in a scenario limited by co-channel interference. The mobile speed was 3 km/h and typical MS impairments were included. Both GMSK and 8PSK channels were simulated. Interleaving was block rectangular over 4 bursts. Retransmissions were sent without delay; as soon as the decoding of a transport block fails (determined by CRC decoding), a retransmission is sent immediately. Since ideal frequency hopping was used, this simplification does not affect the average throughput (only end-to-end delays). RLC/MAC headers were not transmitted.

For GMSK channels transport blocks of 400 bits (20kbit/s) were transmitted, and on 8PSK channels transport blocks of 1200 bits (60kbit/s) were transmitted. Link level results are shown in Fig. 6. Rmean denotes the average number of retransmissions required for successful decoding of the transport blocks.

The performance when using incremental redundancy is compared to simple Chase combining, where each retransmission uses the same puncturing pattern. Combining is made on the soft bit level for both schemes; that is, the

combining is made by simple addition of soft bits after equalization. The throughput without retransmissions is also shown for reference purposes.

The results show that IR in general provides higher throughput and requires less retransmissions than Chase combining. As an example, IR offers 49% more throughput and requires 46% less retransmissions at 4dB of C/I for GMSK channels. At 10dB of C/I on 8PSK channels, IR offers 63% more throughput than Chase combining and requires 53% fewer retransmissions.

VII. CONCLUSION

This paper has introduced the Flexible Layer One for GERAN, and described the implementation of incremental redundancy for acknowledged traffic with FLO. FLO is to be introduced in Release 6 of the 3GPP standards to ensure an efficient means for provision of future services in the GSM/EDGE Radio Access Network. The paper has shown that incremental redundancy is a promising means of enhancing the throughput of acknowledged traffic, regardless of whether traffic is control or data traffic. Furthermore, incremental redundancy can be introduced to FLO in a relatively simple manner, since it can be incorporated in the rate matching function already a part of FLO.

REFERENCES

- [1] 3GPP Technical Specification TS 23.228, IP Multimedia Subsystem – Stage 2.
- [2] 3GPP Technical Report TS 45.902, Flexible Layer One.
- [3] 3GPP Technical Specification TS 25.212, Multiplexing and Channel Coding (FDD).
- [4] B. Sébire, T. Bysted, K. Pedersen. "Flexible Layer One for the GSM/EDGE Radio Access Network", *Proc. ICT2003*, in press.