A new implementation algorithm for multiplexing and channel coding scheme in WCDMA

Dai Lin, Zhou Shidong, Yao Yan
State Key Lab on Microwave & Digital Communications
Tsinghua University, Beijing, 100084, P.R.China
Email address: dlcathy@wireless.mdc.tsinghua.edu.cn

Abstract

Multiplexing and channel coding is of great importance in WCDMA. It provides a data passageway from MAC (Medium Access Control) layer to physical layer to assure the efficient and reliable transmission of data stream. Channel coding scheme is a combination of error detection, error correcting, and rate matching, interleaving and transport channels mapping onto/splitting from physical channels.

In this paper, the scheme of multiplexing and channel coding in WCDMA is introduced and its advantages and disadvantages are discussed. After analyzing the performance of the direct implementation algorithm of this scheme, including its operation complexity, memory requirement and process delay, a new algorithm is put forward by simplification of the scheme. The new algorithm is shown to be much more efficient than the direct one.

Keywords

WCDMA, multiplexing, channel coding, operation complexity, memory requirement, processing delay, transport channel

I Introduction

The 3rd generation mobile telecommunication becomes more and more important because of its capability of multimedia services transmission. One of the main differences on technique requirement between the 2nd generation and the 3rd generation is that the 3rd generation should be able to support multiple services with multiple variable bit rates. In WCDMA, the solution is to use complex multiplexing and channel coding. It is used to handle the data stream from/to MAC and higher layers and offer transport services over the radio transmission link. The scheme of multiplexing and channel coding in WCDMA is powerful and perfect. However, it is found that to implement the scheme directly is of great complexity and may cause intolerable processing delay.

A new implementation algorithm is presented in this paper. It shows much better performance than the direct one. A comparison of operation complexity, memory requirement and processing delay of the two algorithms is listed. Furthermore, an example is given to indicate the advantages of the new algorithm over the direct one.

The scheme of multiplexing and channel coding in WCDMA is introduced in part II. After discussing the disadvantages of the direct implementation algorithm in part III, a new algorithm is illustrated in part IV. In order to prove the superiority of the new one, the performances of the two algorithms are compared. The conclusion is emphasized through an example in the end of the paper.

II Multiplexing and channel coding scheme in WCDMA

Figure 2.2 and figure 2.3 show the multiplexing and channel coding structure for uplink and downlink in WCDMA.

For uplink, first, each transport block is appended with CRC bits for error detection. After concatenating all transport blocks and segmenting them in code blocks, one of the following channel coding schemes is applied: Convolutional coding, Turbo coding and no channel coding. Then "radio frame size equalisation" is performed to insure that the number of bits in each frame is equal during each transmission time interval (TTI). After 1st interleaving is performed, data stream is segmented into radio frames through "radio frame segmentation". The next step "rate matching" ensures that the total bit rate after TrCH multiplexing is identical to the total channel bit rate of the allocated dedicated physical channels when the number of bits between different transmission time intervals is changed.

After handling data stream in each transport channel, all TrCHs are multiplexed. The radio frames from each TrCHs are serially multiplexed into a coded composite transport channel (CCTrCH). Then data stream is divided into different PhCHs through "physical channel segmentation", and the "2nd interleaving" is performed with bits of each physical channel.

The structure for downlink is very similar to that for uplink except the following two main differences. First, in structure for downlink, there is a step of "insertion of DTX indication", which isn't in structure for uplink.

0-7803-6394-9/00/$10.00©2000 IEEE.

Second, rate matching is performed before 1st interleaving, not after it. Thus it's performed for each TTI instead of each radio frame. Since after "rate matching" and "insertion of DTX", the number of bits in each TTI is a multiple of number of frames in a TTI, "radio frame equalisation" is not used.

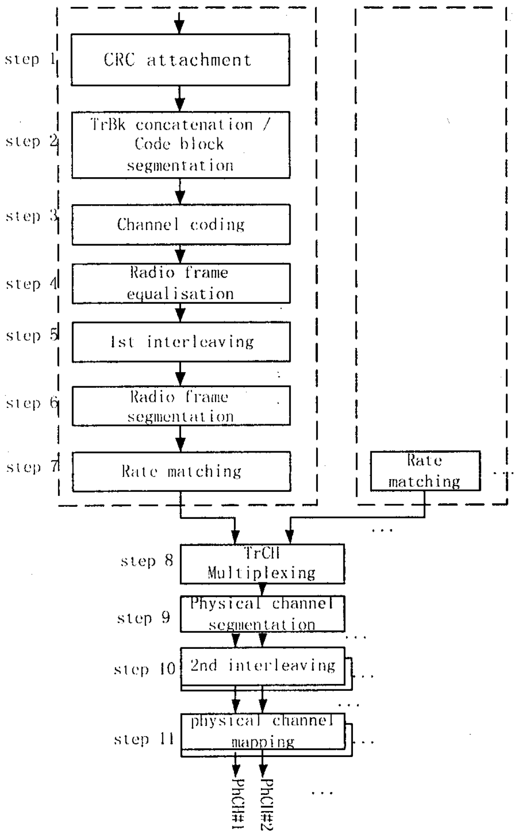

Figure 2.2: Transport channel multiplexing structure for uplink

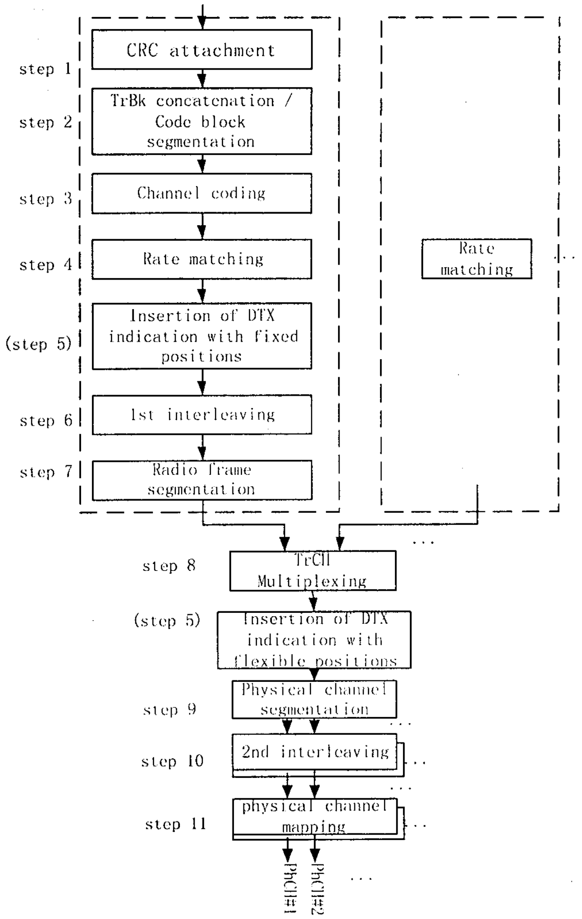

Figure 2.3: Transport channel multiplexing structure for downlink

(Where: TrCH is the abbreviation of a transport channel;
TrBK is the abbreviation of a transport block;
PhCH is the abbreviation of a physical channel
DTX is abbreviation of discontinuous transmission)

III Direct implementation algorithm

From the illustration of the structure for uplink and downlink, a direct implementation of multiplexing and channel coding can be considered naturally. That is, according to every procedure stated in the specification, handle the data stream step by step. However, through the analysis of the scheme, we find it's difficult to implement in this way due to its high complexity.

This conclusion could be explained in three respects: operation complexity, memory requirement and processing delay. (The following discussion is limited to the parts from step 4 to 11.)

First, operation complexity of a bit in every step can be counted, as shown in table 3.1:

Step	add	subtract	judge	read/ write
4			1	1
5				1
7	1	1	1	1
8			1	1
10				1
Uplink				
Step	Add	Subtract	judge	read/ write
4	1	1	1	1
5			1	1
6				1
8			1	1
10				1
downlink				

Table 3.1: Statistics of operation complexity of a bit for uplink and downlink

We can see that for each bit there should be one "add" operation, one "subtract" operation, three "judge" operations and five "read/write" operations.

Second, since data should be operated in blocks step by step, for each step, the intermediate results of the corresponding block have to be stored. Thus the total memory requirement is very large.

Third, total processing delay of TrCH i can be calculated through the following formula:

$$\begin{aligned} P_D &= N_{TTI} \times D_{ct} + D_{pt} + N_f \times D_{cf} + D_{pf} \\ &\quad + D_{pipeline} \\ &= (N_{TTI} \times D_{ct} + N_f \times D_{cf}) + (D_{pt} + D_{pf}) \\ &\quad + D_{pipeline} \\ &\approx D_c + D_p \qquad (f3.1) \end{aligned}$$

where: N_{TTI} is number of bits in a transmission time interval on TrCH i;

N_f is number of bits in a radio frame on CCTrCH;

D_{ct} is processing delay of every bit from step 4 to step 7;

D_{pt} is calculation delay of parameters required from step 4 to step 7;

D_{cf} is processing delay of every bit from step 8 to step 11;

D_{pf} is calculation delay of parameters required from step 8 to step 11;

$D_{pipeline}$ is delay of data stream when being handled in pipeline;

D_c is the total processing delay of every bit from step 4 to step 11;

D_p is the total calculation delay of parameters required from step 4 to step 11.

Since $D_{pipeline} << D_c + D_p$, it could be ignored.

According to Table 3.1, it can be obtained that

$$\begin{aligned} D_{ct} &= (\alpha_a + \alpha_s + 2\alpha_j + 3\alpha_g) / f_{clk}, \\ D_{cf} &= (\alpha_j + 2\alpha_g) / f_{clk} \qquad (f3.2) \end{aligned}$$

where: α_a is number of clock periods required for one "add" operation;

α_s is number of clock periods required for one "subtract" operation;

α_j is number of clock periods required for one "judge" operation;

α_g is number of clock periods required for one "read/' write" operation;

f_{clk} is the clock frequency.

Thus

$$\begin{aligned} D_c &= N_{TTI} \times D_{ct} + N_f \times D_{cf} \\ &= N_{TTI} \times (\alpha_a + \alpha_s + 2\alpha_j + 3\alpha_g) / f_{clk} \\ &\quad + N_f \times (\alpha_j + 2\alpha_g) / f_{clk} \qquad (f3.3) \end{aligned}$$

Suppose that $\alpha_a = \alpha_s = \alpha_j = \alpha_g = 1$, f_{lk} =20Mb/s, N_{TTI}=2Mb/s*10ms, N_f=2Mb/s*10ms,

Then D_c= (2*0.01*7+2*0.01*3)/20=10ms.

It is quite large for a service with TTI 10ms.

The analysis and computation above show that the direct implementation will lead to high operation complexity, large memory requirement and long processing delay. Thus a better algorithm is needed to get better performance.

Ⅳ New implementation algorithm

Through the study of the direct implementation algorithm, it is found that large memory requirement and long processing delay is due to too many steps in the structure. Merging some steps into one can reduce unnecessary access operations to save memory and shorten delay. Furthermore, we find that high operation complexity is caused by dealing with data stream directly. If we move data according to the corresponding addresses before and after coding instead of handling the data stream, the scheme can be simplified.

Based on this idea, a new implementation algorithm can be obtained. As shown in Figure 4.1, step 4 includes steps from "radio frame equalisation" to "rate matching" for uplink. For downlink, it includes steps from "rate matching" to "radio frame segmentation". Step 5 includes steps from "TrCH multiplexing" to "physical channel mapping" for both uplink and downlink.

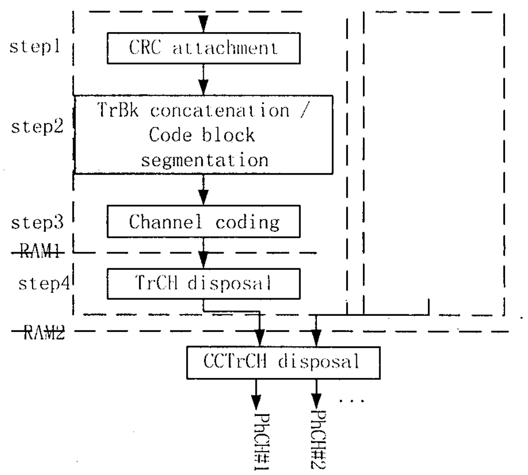

Figure.4.1 the structure of the new algorithm

We take the example of uplink structure to illustrate the new algorithm.

For every bit that is going to carry out step 4, it needn't be written into RAM1, but be stored directly into RAM2 with a new address (a_1) which is calculated as the position of this bit in CCTrCH. Similarly, for every bit that is going to be transmitted, its correspondent address (a_2) in RAM2 is calculated. Then read it from RAM2 at (a_2) and send it to the lower layer. Thus the whole processing is completed without any direct operation on data. We only move them from one place to the other. Moreover, since whenever a bit transmits, its address in ram2 will be calculated immediately, and the processing delay of step 5 is much less than transmission delay, D_{cf} can be ignore.

The performance of the new algorithm is also analyzed in three respects: operation complexity, memory requirement and processing delay.

First, operation complexity of a bit in step4 and 5 can be counted, as shown in table 4.1 :

Step	add	subtract	multiple	judge	read/write
4	1	1		1	1
5	1		1		1

Table 4.1: Statistics of operation complexity of a bit in the new algorithm

Second, it is obvious that the memory requirement in the new algorithm is much less as a result of fewer steps.

Third, similar to analysis in Ⅲ, according to operation complexity shown in table 4.1,

$$\begin{aligned} D_c &= N_{TTI} \times D_{ct} \\ &= N_{TTI} \times (\alpha_a + \alpha_s + \alpha_j + \alpha_g) / f_{clk} \qquad (f4.1) \end{aligned}$$

(D_{cf} could be ignored)

Also taking the example in Ⅲ, it can be calculated that D_c=4ms. Compared with 10ms in direct implementation algorithm, it is much shorter.

Moreover, if we assume that each step in the implementation system can be performed in parallel or by pipeline, then D_c only depends on α_g. According to (f3.3) and (f4.1), it can be calculated that D_c =5ms in the direct algorithm and D_c =1ms in the new algorithm.

Ⅴ Example

Suppose that for an uplink multiplexing and channel coding scheme in WCDMA, bit rate of CCTrCH is 2 Mb/s and bit rate of a TrCH is 2Mb/s whose transmission time interval is 10ms. Assume that $\alpha_a = \alpha_s = \alpha_j = \alpha_g = 1$, f_{clk} =20Mb/s, D_p is 0.1ms.

Then we can compute the operation complexity, memory requirement and total delay of the two algorithms. The result is summarized in Table 5.1.

algorithm	Memory requirement	Total delay	
		Parallel system	Serial system
Direct	10Mb	6ms	11ms
New	2Mb	2ms	5ms

algorithm	Complexity				
	+	−	×	judge	Read/write
Direct	1	1	0	3	5
New	2	1	1	1	2

Table 5.1 Comparison of performances of the two algorithms

Ⅵ Summary

From the analysis and example above, we can conclude that it is not a good way to realize the multiplexing and channel coding scheme in WCDMA directly because of its high operation complexity, large memory requirement and long processing delay. The cause of the bad performance is analyzed and based on the analysis, the new algorithm is put forward. It has shown to have much better performance than the direct one.

If we do the reverse every step of the scheme of multiplexing and channel coding, the scheme of demultiplexing and channel decoding could be obtained. According to the idea of new implementation algorithm, exchange its input and output, then it can apply the demultiplexing and channel decoding scheme.

References

[1] "Physical layer-General description",3G TS 25.201 V3.0.0 (1999-10)

[2] "Multiplexing and channel coding",3G TS 25.212 V2.2.0 (1999-09)