# Optimum Sub-packet Transmission for Turbo-coded Hybrid ARQ Systems

Y.Q. Zhou and J. Wang
Department of Electrical and Electronic Engineering
University of Hong Kong, HONG KONG
yqzhou@eee.hku.hk

***Abstract*-Turbo codes are popular for their near-shannon capacity of bit-error-rate (BER) at low signal-to-noise ratio (SNR) with long packet lengths. They have been proposed in automatic-repeat-request (ARQ) systems as the forward-error correction (FEC) codes. Although sub-packet schemes were proposed in ARQ systems, optimum sub-packet transmission is more effective to maximize throughput in a dynamic channel. Sub-packet schemes can provide additional error correction capability with iterative turbo decoding algorithms. An efficient method is proposed to estimate the optimum number of sub-packets, and adaptive sub-packet schemes, i.e., schemes that enable a system to employ different optimum numbers of sub-packets under variable conditions, are suggested to achieve the maximum throughput of the system. Simulations show that the adaptive sub-packet scheme is effective at moderate SNR, e.g. from 2.6dB to 5.0dB, and can provide higher throughput than conventional packet schemes.**

## I. INTRODUCTION

Since they were proposed by Berrou and Glavieux in 1993 [1], turbo codes have been a very hot research topic due to their wonderful error correction capabilities at small signal-to-noise ratio (SNR). As parallel concatenated convolutional codes (PCCC) [2], turbo codes employ two or more recursive systematic convolutional (RSC) codes as constituent codes (CC), and separate them with pseudorandom interleavers, which are called "*internal interleavers*". At the receiver, the optimum maximum likelihood (ML) decoding becomes intractable because of the internal interleaver. Instead, a sub-optimal decoding is popular, in which constituent codes are decoded with a maximum-a-posteriori (MAP) algorithm (e.g., BCJR algorithm [3]) in which *extrinsic information* is employed within an iterative structure [1-3].

The automatic-repeat-request (ARQ) technique was introduced in the early days of data communication as a consequence of the development of parity-check codes [4]. The main feature of this technique is that it can adapt to the channel conditions at low complexity. In 1960, FEC and error detection were introduced into ARQ protocols, and the results are known as hybrid ARQ protocols today. The hybrid systems can achieve throughput similar to the forward-error correction (FEC) systems and provide the good reliability and flexibility of pure ARQ protocols.

Due to their powerful error correction capability, turbo codes have been considered for hybrid ARQ systems. Recently, it was shown that turbo codes converge to a small number of errors at moderate to high SNR [5]. However, a conventional complete-packet ARQ scheme is inefficient in this case, because the whole packet will be retransmitted even though there are only one or two bit errors. This situation can be improved if sub-packet schemes are employed. In sub-packet transmissions, only those sub-packets that include errors need be retransmitted.

Sub-packet schemes also have advantages in burst error systems. All of the bit errors may be located in one sub-packet, leaving other sub-packets error-free. Fortunately, turbo codes are observed in simulations to have burst errors at moderate to high SNR. The burst error pattern is believed to be a result of the nonlinear iterative decoding algorithm of turbo codes. Since it is very complicated to analyze the performance of the iterative decoding algorithm (even if possible), computer simulations seem to be the only way to show its property.

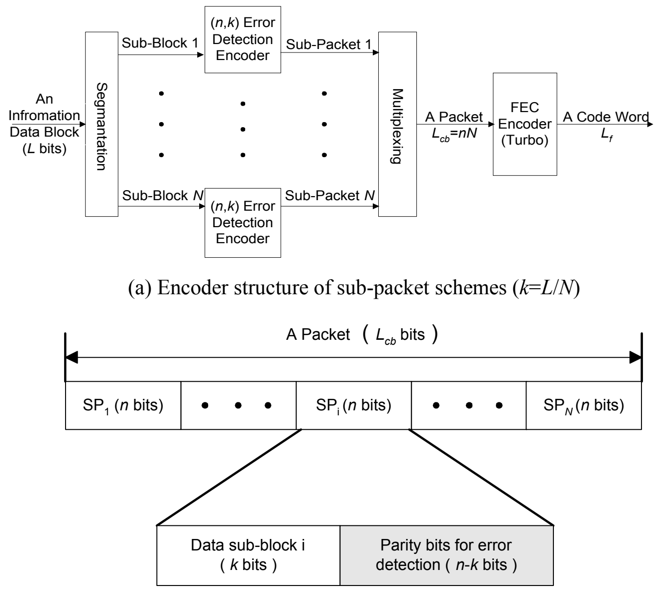
Although sub-packet schemes were proposed in ARQ systems [6], there is no paper discussing the optimum number of sub-packets, and a fixed sub-packet scheme can only provide better throughput in a small range of SNR. Ascertaining the optimum number of sub-packets in one transmission is the objective of this paper. An adaptive sub-packet scheme is proposed to provide the highest throughput in a dynamic channel.

## II. SYSTEM OVERVIEW

In a hybrid FEC/ARQ system, the information data blocks are encoded with error detection and FEC codes. The coded bit stream is modulated by binary phase shift keying (BPSK). After passing through the channel (additive white Gaussion noise (AWGN) channel in this paper) and being demodulated, the base-band signals are decoded by *error correction* and checked by *error detection*. Finally, acknowledgment (ACK) and negative acknowledgment (NAK) signals, are transmitted back to the transmitter over an ideal feedback channel that is assumed to be error-free.

Suppose that an information data block with $L$-bit length is transmitted with ARQ techniques. The whole information data block is segmented into $N$ sub-blocks as illustrated in Fig. 1 (a), which consists of $N$ error detection encoders and one FEC. Each

0-7803-7802-4/03/$17.00 © 2003 IEEE




(a) Encoder structure of sub-packet schemes ($k$=$L$/$N$)

(b) Sub-packet structure

Figure 1. Sub-packet schemes

detection code is a ($n$,$k$) systematic block ($k = L/N$ is an integer). The output of the FEC encoder is a channel-coded word of length $L_f$. The sub-packet structure is shown in Fig. 1 (b) where each sub-packet includes a systematic error detection code of $n-k$ bits. In the following analysis, two different PER are used in the $N$ sub-packet schemes. One is the sub-packet error rate that is denoted as $P_{sp}(N)$, and the other is the error rate of the whole packet, which is denoted as $P_w(N)$. Then the PER of the complete-packet scheme is simply represented by $P_w(1)$ or $P_{sp}(1)$.

A systematic CRC code with 16 parity bits ($L_{crc}$ = 16) in CCITT standards is used as the error detection code [4]. Its generator polynomial is $g(D) = D^{16} + D^{12} + D^5 + 1$, where $D$ stands for bit delay operation. The performance of CRC codes is mainly determined by $L_{crc}$. The longer the $L_{crc}$, the better the performance. The CRC code with $L_{crc}$ = 16 can provide adequate detection for most applications.

A turbo code of rate $R$ is chosen as the FEC code. The turbo code is a 1/3 rate code [7] that is composed of two constituent codes ($RSC_1$ and $RSC_2$) with identical rates of 1/2, whose generator polynomial is $G(D) = [1, (1 + D + D^3)/(1 + D^2 + D^3)]$. Pseudorandom interleavers with size $L_{cb}$, which is the same as the packet length shown in Fig. 1, are chosen as the internal interleaver. Higher rate codes are obtainable from the 1/3-rate code by puncturing when SNR is large. In turbo decoding, the maximum iterative times is set to a specific value (e.g., 20) that is large enough to exploit the *iterative gain*.

Consider the ARQ protocol of sub-packet schemes. As shown in Fig. 1 (b), sub-packets are independent of each other due to different data sub-blocks individually encoded by CRC codes. At the receiver side, by the end of each iteration of the MAP algorithm, a hard decision is made and the syndrome of CRC code is calculated for each sub-packet. If CRC code detects no error in the sub-packet, the corresponding decoded sub-block is stored in buffer and an ACK is sent to the transmitter. The iterative decoding goes on until all of the sub-packets are error free or the maximum iteration times are reached. If there are still some erroneous sub-packets after the decoding is finished, then the corresponding NACK and sub-packet length are sent to the transmitter for retransmission. However, correctly detected sub-blocks are stored in a buffer, and sequentially forwarded to the higher layer. The maximum transmission times for a particular sub-packet is limited to $M$. If the sub-packet is still in error after $M$ transmissions, then it will be dropped. It should be noted that high quality feedback channel is needed because wrongly received acknowledgment signals and sub-packet length will degrade the system performance.

Within this ARQ protocol, sub-packet turbo codes can make use of the oscillating number of bit errors of turbo codes to reduce the error probability. The oscillating property of turbo codes that were presented in [8] means that the number of bit errors in a packet oscillates as decoding iteration goes on. Table I shows an example of the number of bit errors in a packet at different iteration stages when $L_{cb}$ = 3072, $R$ = 4/5, and $SNR_b$ = 3.75 dB ($SNR_b$ is defined in section IV). Besides the oscillating number, it is shown that the error positions are found to change with iteration stages. Then sub-packet schemes can gather the correct information in different iteration stages. From this point of view, sub-packet schemes have additional error correction ability with turbo codes because of the oscillations that are induced by the iterative decoding algorithm.

## III. Theoretical Bounds

By use of the *uniform interleaver*, which is a probabilistic device that acts as the average of all possible interleavers of a given size, the *average* weight enumerating function (AWEF) of turbo codes can be constructed from constituent codes $RSC_1$ and $RSC_2$, as illustrated in [2]. Using maximum likelihood (ML) soft decoding and AWEF, *average* upper bounds of PER ($P_p$) and BER ($P_b$) can be obtained on the turbo code of rate $R$ with BPSK modulation over AWGN channel:

TABLE I

Oscillating Number Of Bit Errors In A Packet

($L_{cb}$ = 3072, $R$ = 4/5, $SNR_b$ = 3.75 dB)

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Iteration Stages | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Number of errors | 28 | 1 | 1 | 9 | 9 | 5 | 6 | 11 | 7 | 25 |
| Iteration Stages | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Number of errors | 13 | 16 | 4 | 16 | 24 | 33 | 7 | 4 | 8 | 15 |

$$P_p < \sum_d A_d P_d = \sum_d A_d Q\left(\sqrt{\frac{2dRE_b}{N_0}}\right) \quad (1)$$

$$P_b < \frac{1}{L_{cb}} \sum_d B_d P_d = \frac{1}{L_{cb}} \sum_d B_d Q\left(\sqrt{\frac{2dRE_b}{N_0}}\right) \quad (2)$$

where $A_d$ is the *average* number of code words in turbo codes with weight $d$, $B_d$ is the *average* total number of non-zero information bits of all code words with weight $d$, $E_b$ is the energy of each information bit, and $N_0$ is the single-sided power spectral density of the channel noise.

Define

$$N_{err} = \frac{P_b \cdot L_{cb}}{P_p} \quad (3)$$

as the average number of bit errors in an incorrect packet. This parameter will be used in next section to decide the SNR value when sub-packet schemes begin to outperform complete packet schemes in a turbo-coded system.

Note that turbo codes with bad interleavers will have higher error rates than the bounds in (1) and (2), as these upper bounds are average bounds. It has been shown that for moderate to high SNR, i.e., at the error floor of error rate curves, the asymptotic performance of PER and BER is dominated by the first few terms with low Hamming weight in the bounds (see (1) and (2)). It can be predicted that the erroneous packets at moderate to high SNR should be caused by very few bit errors in most cases and sub-packet schemes can be used to improve the system performance. Moreover, for the higher rate turbo codes, the PER error floor will be higher and flatter since there are more code words with lower weight because of puncturing. Thus, sub-packet schemes will be even more helpful in improving the performance of higher rate turbo codes.

## IV. Analysis And Simulation

As discussed, sub-packet schemes are suitable for turbo-ARQ systems. This section is devoted to finding the optimum sub-packet schemes to maximize the system throughput.

Consider a $N$-sub-packet scheme with a $(n,k)$ error detection code. The effective code rate of the packet scheme is defined as:

$$R_{eff} = \frac{L}{L_f} = \frac{N \cdot k}{L_f} = \frac{L_{cb} - N \cdot L_{crc}}{L_f} \quad (4)$$

which accounts for the redundancy introduced by error detection codes. Define $SNR_b = E_b / N_0$ as the SNR per information bit, and $SNR_c = R_{eff} \cdot SNR_b$ as the SNR per channel bit.

### *A. Optimum Sub-packet Schemes*

Throughput is chosen as the criterion for optimizing the number of sub-packets, $N$. Consider a truncated system that employs a Selective Repeat-ARQ (SR-ARQ) protocol with $M$ maximum transmissions. The throughput is defined as [4]:

$$\eta = R_{eff} \cdot (1 - P_{sp}(N)) = \frac{L_{cb} - N \cdot L_{crc}}{L_f} \cdot (1 - P_{sp}(N)) \quad (5)$$

When $L_{cb}$, $L_{crc}$, code rate (or $L_f$), and SNR are given, the throughput only depends on $N$. The optimum number of sub-packets ($N_{opt}$) that achieve the highest throughput can be obtained by setting the differentiation $\partial\eta/\partial N$ to zero. For the turbo-ARQ system employing a $N$-sub-packet scheme, at moderate to high SNR, the error rate of the whole packet, $P_w(N)$, can be approximated by (1). Furthermore, with the assumption of long packet lengths, the effect on weight distribution and code rate on turbo codes from CRC encoders can be ignored. Thus, $P_w(N) \approx P_w(1)$. And for moderate to high SNR, BER is small and bit errors are in burst, so that $P_{sp}(N)$ can be approximated by $P_w(1)/N$ and $N_{opt}$ can then be obtained:

$$N_{opt} \approx \sqrt{\frac{L_{cb}}{L_{crc}} \sum_d A_d Q\left(\sqrt{\frac{2dE_b}{N_0} \cdot \frac{L_{cb} - L_{crc}}{L_f}}\right)} \quad (6)$$

Given (6), the system can decide $N_{opt}$ theoretically.

Since finding the weight enumeration of turbo codes is not a trivial task, another method to avoid this problem is to use the simulated PER of complete-packet schemes to replace its bound. Thus, one obtains:

$$N_{opt} \approx \sqrt{\frac{L_{cb} \cdot P_w(1)}{L_{crc}}} \quad (7)$$

Equation (7) clearly describes the relationship between $N_{opt}$ and other system parameters. Firstly, $N_{opt}$ is limited by the redundancy introduced by error detection codes. When CRC is long (large $L_{crc}$), $N_{opt}$ may take a value as small as one. When $L_{crc}$ is very small, $N_{opt}$ will become large. However, for CRC codes, $L_{crc}$ should be large enough to ensure the detection capability. Therefore, to balance both CRC detection capability and sub-packet scheme, an appropriate value of $L_{crc}$ should be chosen. Secondly, when $L_{crc}$ and $P_w(1)$ are given, $N_{opt}$ increases with $L_{cb}$, which means more benefit from large $N$ sub-packet schemes in long packet length systems. Thirdly, a lower bound of $P_w(1)$ where sub-packet schemes are assumed can be obtained from (7) because $N_{opt} \geq 1$, and is given by:

$$\underline{P_w} = L_{crc} / L_{cb} \quad (8)$$

Thus, a longer packet length $L_{cb}$ can result in a lower $\underline{P_w}$, which will increase the effective SNR region of sub-packet schemes. In the SNR region where $P_w(1) < \underline{P_w}$, $N_{opt}$ is one, which means that complete-packet schemes are the best packet schemes in this region.

Therefore, given $L_{crc}$ and $L_{cb}$, the optimum packet scheme can be constructed. First, the PER performance when $N = 1$ is obtained by simulations. Then the SNR value should be found where the turbo code begins to present sparse and burst error patterns. The corresponding PER is denoted as $\overline{P_w}$. In the region between $\underline{P_w}$ and $\overline{P_w}$, $N_{opt}$ is calculated using (7).

The proposed method of constructing adaptive sub-packet schemes is verified as follows. Consider a system with $L_{cb}$ = 7168, $L_{crc}$ = 16 and $R$ = 4/5. The PER and BER performance when $N$ = 1 is obtained via simulations, and it is found that when $N_{err} \le 0.01 \cdot L_{cb}$, sub-packet schemes outperform complete packet schemes. Thus, $N_{err} \le 0.01 \cdot L_{cb}$ is adopted as a criterion to decide the point where sub-packet schemes are good. With this criterion and (8), the $SNR_b$ range for sub-packet schemes is from 2.6 to 5.5 dB. In this range, the optimum $N_{opt}$ can be found by using (7). For comparison, simulations are carried out to find $N_{opt}$. The adaptive sub-packet schemes that are obtained from the proposed algorithm and simulations are depicted in Fig. 2. The real value that is obtained from (7) is also shown as a reference. This value is extremely inaccurate at low SNR (or high PER). As SNR increases, the analyzed value is a little different from that simulated. When SNR increases further, the estimated $N_{opt}$ coincides with the value that is obtained from the simulations. In summary, the proposed analytical method can accurately estimate the $N_{opt}$ at wide range of the SNR.

It is seen that the optimum packet scheme is not a constant $N$ scheme. To the contrary, $N_{opt}$ changes with SNR. The scheme that enables a system to employ different optimum $N$ under variable SNRs is called "*an adaptive sub-packet scheme*". SNR estimation is thus needed to employ this scheme. In fast changing channels, the estimation error will deteriorate the system performance. However, in slowly changing channels, using the common channel estimation technique and a predefined table of $N_{opt}$ at different SNR, the adaptive sub-packet scheme can be easily realized. In the following discussion, the system performance of throughput is evaluated with the optimum adaptive scheme.

### *B. Throughput*

In a truncated system that employs a SR-ARQ protocol with $M$ maximum transmissions, the throughput is given by (5). Given that $L_{cb}$ = 7168, $L_{crc}$ = 16, and $R$ = 4/5, the system throughput is illustrated in Fig. 3 with different fixed $N$ sub-packet schemes. It is shown that any fixed $N$ sub-packet schemes can just provide the highest throughput only in some part of the considered SNR region. Hence, only when the number of sub-packets in the system changes with SNR, i.e., when an adaptive sub-packet scheme is employed, the system can achieve the highest throughput all the time. It is also seen that the throughput improves less at high SNR. At high SNR, $P_w(1)$ is quite low and the decrement in PER is small. As indicated by (5), the throughput improvement in this case is not so large as that at moderate SNR.

In order to investigate the effect of packet length on the optimum sub-packet scheme, Fig. 4 shows the throughput ($\eta$) as a function of the number of sub-packets ($N$) with different packet lengths when $SNR_b$ = 3.125 dB and $R$ = 4/5. It is clearly shown that as $N$ increases, the system throughput improves at first, then the improvement of the throughput performance becomes smaller as $N$ increases further, and at last, it decreases because there is too much redundancy introduced by the CRC codes. Hence, the $N_{opt}$ can easily be found from the top points of these convex curves. It is seen that the system with longer packet lengths can achieve higher throughput with larger $N_{opt}$. Therefore, from the throughput point of view, the system should employ a turbo code with a packet length as long as possible.

The performance of optimum sub-packet schemes is also evaluated with different code rates. Fig. 5 illustrates the system throughput as a function of the number of sub-packets ($N$) with code rates of 4/5, 5/6, and 6/7 for $L_{cb}$ = 7168. The SNRs are 3.0 dB, 3.125 dB, and 3.25 dB, respectively, for the different rates. It can be seen from this figure that with higher code rates, optimum sub-packet schemes achieve more throughput improvement and the $N_{opt}$ is also larger. This is due to the higher error floor of higher rate turbo codes, as explained in the last section. Therefore, when the channel condition is good and high rate turbo codes are employed, optimum adaptive sub-packet schemes can further increase the system throughput.

## V. Conclusions

This paper presents an optimum adaptive sub-packet scheme for turbo-ARQ systems. By means of theoretical analysis and computer simulations, it is shown that:

1) Sub-packet schemes are suitable for turbo-ARQ systems at moderate SNR. The longer the packet length, the higher the code rate, and the more favorable the sub-packet scheme;

2) For given conditions, the optimum number of sub-packets ($N_{opt}$) can be derived with the assumption of sparse and burst packet error patterns. $N_{opt}$ will change with SNR. Thus, an optimum adaptive sub-packet scheme is proposed for dynamic channels.

## References


[1] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-Codes (1)," *Proc. Of ICC'93*, pp. 1064-1070, May 1993.

[2] S. Benedetto and G. Montorsi, "Unveiling turbo codes: some results on parallel concatenated coding schemes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 409-428, March 1996.

[3] L. R. Bahl, J. Cocke, F. Jelinek and R. Raviv, "Optimal decoding of liner codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, pp. 284-287, Mar. 1974.

[4] S. Lin and D. J. Costello Jr., *Error control coding: fundamentals and*

*applications*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[5] O. Y. Takeshita, Oliver M. Collins, Peter C. Massey and Daniel J. Costello, "On the frame-error rate of concatenated turbo codes," *IEEE Trans. Commun.*, vol. 49, pp. 602-608, April 2001.

[6] N. Guo, F. Khaleghi, A. Gutierrez, J. Li and M-H. Fong, "Transmission of high speed data in cdma2000", *Wireless Communications and Networking Conference*, 1999. WCNC. 1999 IEEE, vol.3, pp. 1442-1445.

[7] 3rd Generation Partnership Project, Technical Specification Group Radio Access Network, "Multiplexing and channel coding (FDD), (Release 1999)," ETSI TS 25 212 V3.5.0, Dec. 2000.

[8] J. Hamorsky and U. Wachsmann, "Criteria for minimum bit error tate decoding of turbo-codes," in *Proc. Kleinheubacher Tagung*, Kleinheubach, Germany, 1995, pp. 607-616.

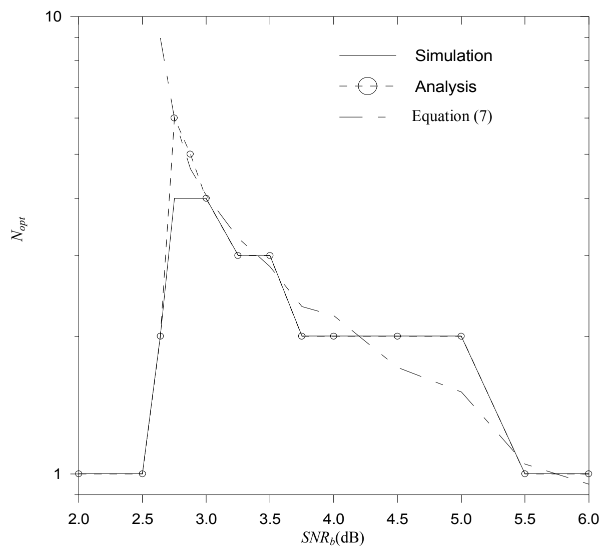



Figure 2. Comparison of the adaptive-packet schemes that were obtained from analysis and simulation

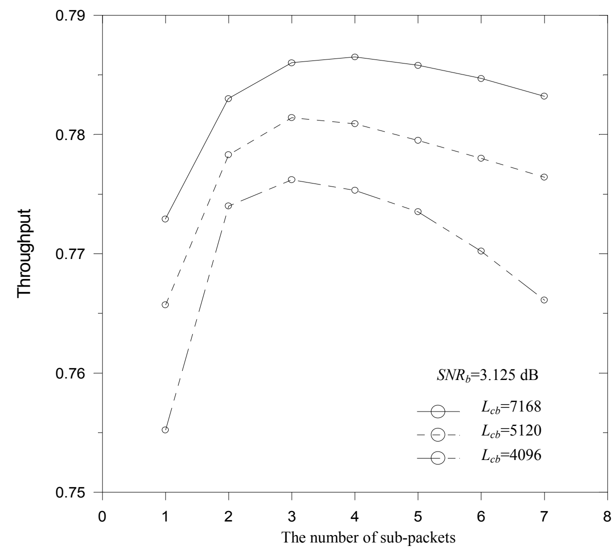



Figure 4. The system throughput as a function of $N$ with different packet lengths

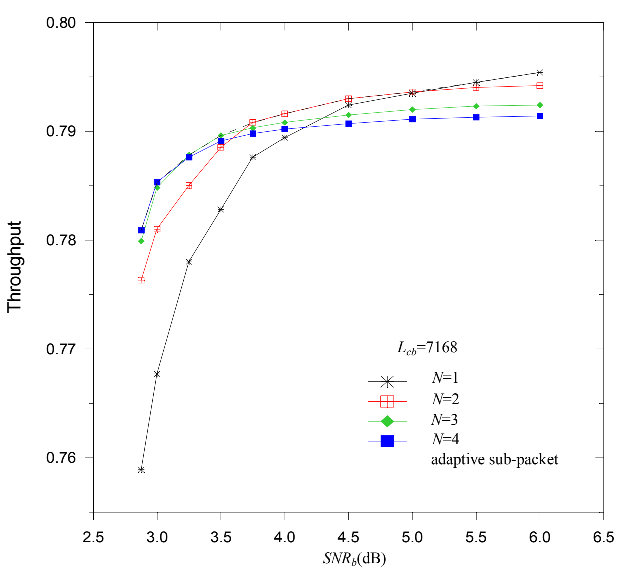



Figure 3. The system throughputs with fixed $N$ and adaptive sub-packet schemes

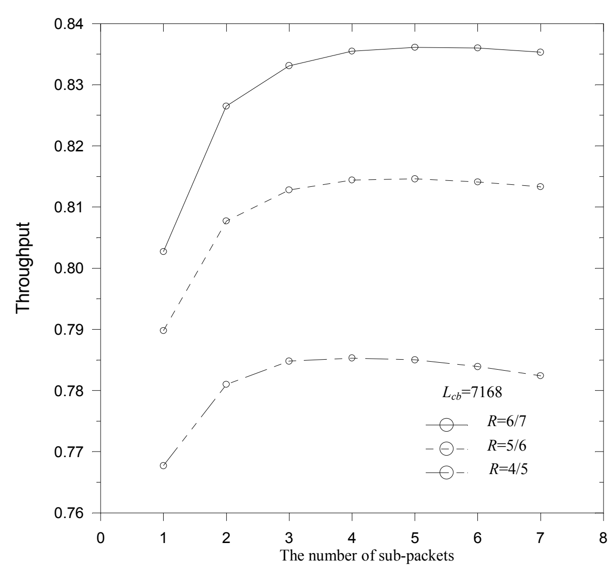



Figure 5. The system throughput as a function of $N$ with different code rates