

Fountain-Code Aided File Transfer in 802.11 WLANs

H.Chen, R.Maunder and L. Hanzo

School of ECS, University of Southampton, SO17 1BJ, UK.

Tel: +44-23-8059 3125, Fax: +44-23-8059 4508

Email: {hc07r,rm,lh}@ecs.soton.ac.uk; http://www-mobile.ecs.soton.ac.uk

Abstract—Fountain codes constitute novel erasure codes, which have been standardized for Forward Error Correction (FEC) in broadcast network protocols and by the Third-Generation Partnership Project (3GPP). The basic operational units of Fountain codes are source packets, which have a particular fixed length. These codes are invoked here in an 802.11 Wireless Local Area Network (WLAN) scenario for protecting file transfers. More specifically, the optimal packet length is selected by considering the 802.11 Media Access Control (MAC) retransmission rate and the properties of the physical layer's modulation scheme. Naturally, owing to the limited memory of the encoders/decoders, large source files must be decomposed into shorter transport blocks. Therefore, methods for partitioning the file and acknowledging the successful transmission of each block are also proposed here. Compared to the file transfer regime operating without FEC over the classic TCP protocol, the proposed regime requires a lower threshold SNR for accomplishing a successful file transfer and hence enhances the transmission efficiency by about 50%.

I. INTRODUCTION

In recent years, the IEEE 802.11 WLAN protocol [1] has gained popularity for end-user connectivity. However, 802.11 data transmissions may suffer from high Packet Loss Ratios (PLRs) owing to the hostile characteristics of wireless channels, like multipath fading, shadowing and noise. Therefore, improving the PLR of 802.11 WLAN schemes constitutes a promising area of research. Traditional methods of protecting data transmissions include Automatic Repeat reQuest (ARQ) [2], Forward Error Correction (FEC) [3] and hybrid FEC-ARQ schemes. The 802.11 MAC protocol has adopted the acknowledgment (ACK) and retransmission based approach for its reliable data exchange. Indeed, many recent studies [4, 5] have considered channel-quality-dependent adaptations of the physical (PHY) and MAC layers, in order to improve their reliabilities.

Nonetheless, it would be beneficial to improve the achievable end-to-end Quality of Service (QoS) without modifying the lower layers. For example, Reed-Solomon (RS) codes have found popularity as Application Layer FECs (AL-FECs) [6–8] owing to their maximum minimum free distances, which facilitate robust communications.

However, Fountain codes [9] have been shown to offer advantages over RS codes in terms of reduced complexity and improved coding efficiency [10, 11]. Again, RS codes have maximum-minimum free distances and their coding rate is K/N , where K is the number of non-binary information symbols, N is the number of encoded symbols and an (N, K) RS code is capable of recovering (i.e filling) up to $(N - K)$ erased symbols, provided that the index of the erased symbols is known. Similar statements may also be made in terms of filling entire erased packets. By contrast, Fountain codes

are rateless erasure filling codes, which can send exactly the required number of encoded packets, which is needed to ensure that the original data file is recovered. For a block of K source packets, the Fountain decoder can recover the entire block with a high probability, when it receives at least $N = K(1 + \epsilon)$ encoded packets, where ϵ is referred to as the transmission overhead. Furthermore, Fountain codes are more amenable to soft-decision decoding than RS codes. Luby Transform (LT) and Raptor codes, both of which constitute members of the Fountain codes family have also been shown to impose significantly lower complexities [9, 12]. Owing to these benefits, the related family of Raptor codes was standardized both for Digital Video Broadcasting (DVB) [13, 14] and by the Third-Generation Partnership (3GPP) [15] as an AL-FEC scheme in their Internet Protocol based Television (IPTV) solution.

When implementing Fountain codes, it is necessary to specify the packet length L_{data} , the algorithm used for partitioning the source file into transport blocks, the FEC payload ID format and so on. The appropriate selection of these parameters controls the receiver's delay, while maintaining a low PLR. Each of above mentioned standards has paid attention to these considerations. The standards [13–15] also define the method of constructing source blocks, the source block index, etc. However they were designed for broadcast networks, rather than for half-duplex 802.11 WLAN environments.

II. FOUNTAIN CODING PACKET SIZE

In this section, we elaborate on the Fountain codes invoked in the application layer of 802.11 wireless LANs for file delivery and design an efficient cross-layer operation aided FEC scheme. As Figure 1 shows, after deciding the packet length L_{data} using the technique to be outlined later in this section, the FEC process begins by partitioning the L_{file} -byte source file into Z number of source blocks. Annex C in the Content Delivery Protocol description of the DVB standard [13] recommends a source-partitioning method, which uses the same number of source packets in all source blocks. We propose a number of justified modifications, since the maximum block size is based on the memory available in the decoder. The file is decomposed into $T = \left\lceil \frac{L_{file}}{L_{data}} \right\rceil$ source packets, which are then mapped to the Z number of source blocks as seen in Figure 1.

Each of the Fountain-encoded packet seen in Figure 1 is generated by the modulo-2 addition of randomly chosen source packets of a specific source block. The number of source packets that contribute to a specific Fountain-encoded packet is referred to as the 'degree' of the encoded packet, which is a random integer depending on the specific degree distribution

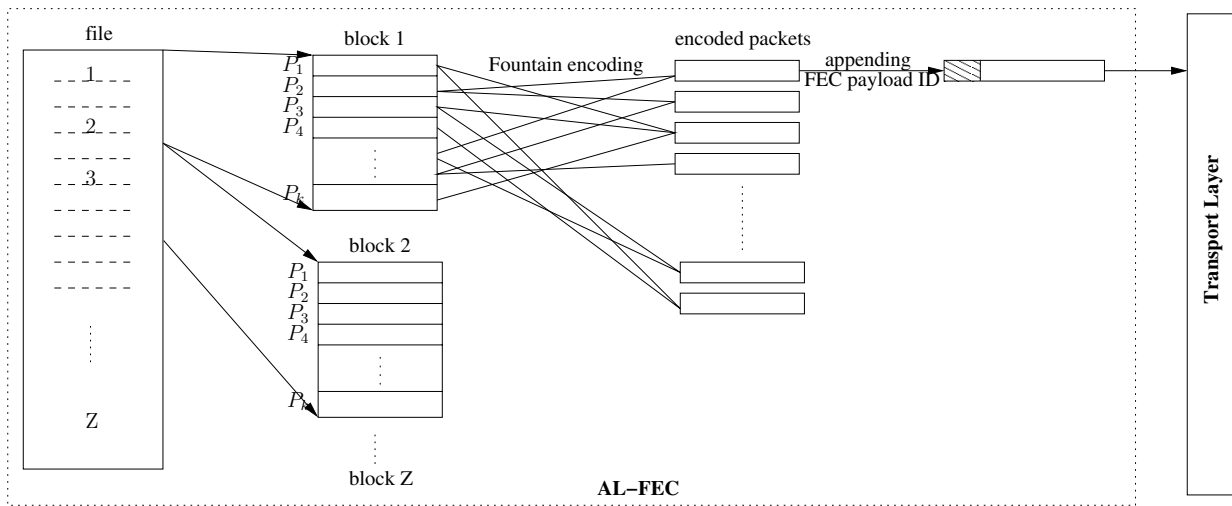


Fig. 1. AL-FEC packetization using Fountain codes.

employed [9], where the average degree of all encoded packets determines the complexity of the encoder and decoder. Finally, the Fountain-encoded packets of Figure 1 are given an FEC payload ID, which includes a Source Block Index (SBI) and the seed for the pseudo-random number generator, which will be used to generate the same random sequence in the decoder.

The transmitter continuously transmits the Fountain-encoded packets of a particular block, until an acknowledgement flag's reception indicates that the destination has successfully reconstructed the block. However, in poor channels, this feedback message may get lost. In order to circumvent this problem, the decoder may opt for continuing to transmit the feedback flag, as long as it continues to receive redundant Fountain-encoded packets having the same SBI as the reconstructed block. If the source transmitter consistently fails to receive the ACK message, after a timeout it will cease its transmission attempts, when the number of transmitted packets reaches the limit of $N_{max} = (1 + \alpha)N$, where $\alpha > 0$ is a parameter of the file transfer scheme. This mechanism is capable of preventing indefinite network loading, when the PLR is excessive.

Our design objective is to maximize the average transmission efficiency required to successfully convey each block by carefully selecting the FEC parameters. Considering the PLR P_{lost} in the application layer, the transmission efficiency may be simply defined as

$$R = (1 - P_{lost}) \cdot \frac{L_{data}}{D}, \quad (1)$$

where D is the total load imposed by each packet on the network, which is expressed in bytes. Therefore $\frac{L_{data}}{D}$ may also be referred to as the normalized channel utilization ratio.

A Fountain-encoded packet will be extended by attaching the redundant header of the Fountain code, a User Datagram Protocol (UDP) header, an Internet Protocol (IP) header, a MAC and a Physical Layer Convergence Procedure (PLCP) header in sequence. Therefore, the size of the transmitted packet is the sum of L_{data} and the total size L_h of all

the headers. Intuitively, long packets are more likely to be rejected by the 802.11 Cyclic Redundancy Check (CRC), since they comprise more bits which may become corrupted. In this event, the Fountain encoder will be required to send additional packets. On the other hand potentially, less packets are required for transmitting a given file, if large packets are employed, reducing the overhead associated with the various headers. Hence our forthcoming discussions will address striking an attractive tradeoff.

A. Packet Loss Ratio

Again, Fountain codes are intended for use in erasure channels, where packets are either correctly received or lost. These packet loss events may be imposed in 802.11 WLANs for example owing to routing queue overflow, node mobility, co-channel interference, fading etc. In this paper, we only consider the effects of wireless propagation induced symbol errors, assuming that other conditions do not inflict packet loss events, while the extension to other effects will be modelled in our future work.

When a Fountain-coded packet is transmitted over the channel, it will be modulated using an M -ary symbol alphabet, where we have $M = 2$ in BPSK, $M = 4$ in QPSK and so on. The packet will be discarded, if residual symbol errors are detected by the CRCs of 802.11. The Symbol Error Ratio (SER) can be used to derive the PLR associated with an uncorrelated Rayleigh fading channel, where the SER P_s of a certain modulation scheme is a decreasing function of the Signal Noise Ratio (SNR). Indeed the PLR $P_p(L)$ can be expressed as

$$P_p(L) = 1 - (1 - P_s)^{\frac{L \cdot 8}{\log_2(M)}}, \quad (2)$$

where L is the length of the packet in bytes.

However, the PLR expression of the application layer is less straightforward, because the 802.11 MAC adopts Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) [3], which is further supported by ARQ for the sake of

improved peer-to-peer reliability. The 802.11 MAC provides two mechanisms for transmitting packets in its basic medium access protocol using the so-called Distributed Coordination Function (DCF) [1]. Firstly, the basic access method only includes data and ACK interaction. By contrast, the more sophisticated four-way handshaking procedures of 802.11 uses ‘Request To Send’ (RTS) and ‘Clear To Send’ (CTS) signals to test, whether the channel is sufficiently uninterfered before transmitting its data. In this section, we consider the more sophisticated four-way handshaking mode, but our analysis may be readily simplified for the case, where no RTS and CTS signals are employed.

In the four-way handshaking mechanism, when a node has a packet to transmit, it will wait for the medium to become idle. Next, it will send an RTS message to make a request and wait for the corresponding CTS confirmation from its intended destination. If no CTS packet is received after the window of Short Interframe Space (SIFS) owing to a packet collision or bit errors, more RTS attempts are made, until the so-called short retry limit R_1 is reached. After a successful RTS/CTS exchange, the source node transmits its data packet and waits for the corresponding ACK message. In the case, If no ACK is not received after a window constituted by the sum of the data packet propagation delay plus SIFS, the data packet must be retransmitted, which requires a new RTS/CTS exchange. Likewise, the number of data packet retransmissions is constrained by the so-called long retry limit R_2 . The packet will be discarded all together, if either of the short or the long retry limits of R_1 and R_2 are reached. The probability of an RTS/CTS exchange failure may be expressed as

$$P_r = [P_p(L_{rts} + L_{cts})]^{R_1}, \quad (3)$$

where L_{rts} and L_{cts} are the RTS and CTS packet length expressed in bytes, respectively.

In the forthcoming analysis, we do not have to consider ACK failures, since they do not prevent the packet from being successfully received. The i th transmission¹ of the data packet will be required, when its first $(i - 1)$ transmissions fail but the RTS/CTS exchanges in all of these $(i - 1)$ transmissions were successful². The probability of this event is given by

$$P_{f_i} = [(1 - P_r) \cdot P_p(L_{data} + L_h)]^i, \quad (4)$$

where $(L_{data} + L_h)$ is the aforementioned data packet size expressed in bytes. Therefore, the total PLR experienced at the application layer is constituted by the sum of the probabilities of RTS/CTS exchange failures in any round and the data packet failure during the last retransmission, which can be expressed as

$$P_{lost} = \sum_{i=0}^{R_2-1} P_{f_i} \cdot P_r + P_{f(R_2)}, \quad (5)$$

¹This is acutally a i th retransmission for the same packet in the 802.11 MAC layer and the value of it is confined by the retry limit R_2 .

²If the RTS/CTS exchange fails in any round, the transmission of the packet is terminated immediately and the packet is then discarded. In this case, another transmission will not happen.

where $(P_{f_i} \cdot P_r)$ denotes the packet loss probability during the i th transmission attempt imposed by the RTS/CTS exchange failure. Note that the data failure during the i th transmission attempt will not cause a packet loss, if i does not reach the limit R_2 . However, the data packet failure during the last transmission must be counted, hence $P_{f(R_2)}$ is embodied in Equation 5. Finally, for fixed values of P_s , L_{rts} , L_{cts} , L_h , R_1 and R_2 , the packet loss probability P_{lost} is an increasing function of L_{data} .

B. Retransmission Cost

In an 802.11 WLAN, each packet may be retransmitted and acknowledged several times. Furthermore, each transmission may invoke several RTS/CTS exchanges. Therefore the successful delivery of each packet may impose significantly longer channel occupancy than their duration. The average channel occupancy associated with each RTS/CTS signal is given by

$$D_r = L_{rts} + [1 - P_p(L_{rts})] \cdot L_{cts}, \quad (6)$$

where $[1 - P_p(L_{rts})]$ quantifies the probability that the RTS packet is received by the destination, therefore triggering the transmission of a CTS signal. Hence, the total RTS/CTS related extra channel occupancy D_R associated with each transmission is given by

$$D_R = \sum_{i=0}^{R_1} [P_p(L_{rts} + L_{cts})]^i \cdot D_r, \quad (7)$$

where $[P_p(L_{rts} + L_{cts})]^i$ indicates the probability that the i th transmission of the RTS/CTS signal will be required.

Whenever a data packet is successfully received, an ACK message will be generated. These ACK messages must be carefully considered, because they impose extra channel occupancy, hence extra interference and another retransmission related to the data packet will be performed, whenever the ACK is not successfully returned. The average channel occupancy imposed by each data packet’s transmission is given by

$$D_d = D_R + (1 - P_r) \cdot \{(L_{data} + L_h) + [1 - P_p(L_{data} + L_h)] \cdot L_{ack}\}, \quad (8)$$

where L_{ack} denotes the number of bytes constituting the ACK message, $(L_{data} + L_h)$ is the data packet length in bytes, and $[1 - P_p(L_{data} + L_h)] \cdot L_{ack}$ represents the channel occupancy imposed by ACK based on the premise of a successful data packet reception.

Then, the probability that another retransmission will be attempted for this data packet is given by

$$P_{df_i} = [(1 - P_r) \cdot P_p((L_{data} + L_h) + L_{ack})]^i. \quad (9)$$

Finally, the total channel occupancy imposed by each data packet is composed of the sum of each retransmission’s

occupancy, which is given by

$$D = \sum_{i=0}^{R_2} P_{df_i} \cdot D_d . \quad (10)$$

C. Maximum Transmission Efficiency

Theoretically, the average number of packets to be transmitted for the successful delivery of a single source packet, namely X may become high, if the PLR is high. However, in practice, only a limited latency can be tolerated and hence an upper limit $X_{max} = (1 + \alpha)$ is imposed on X , since each block has a timeout, as discussed before. Hence, the maximum tolerable PLR can be expressed as $P_{max} = \alpha / (1 + \alpha)$. Indeed, this confines the maximum data packet length $L_{data,max}$ according to Equation 5. On the other hand, the data packet length L_{data} must have a value of at least one byte, which hence determines the minimum value α_{min} .

Our objective is to select α and L_{data} in order to strike an attractive compromise, when transferring a file. Since L_{data} is required to be an integer between one and $L_{data,max}$ bytes, all legitimate values can be substituted into Equation 1 in order to determine the optimum packet size, which maximizes the transmission efficiency.

In the NS2 simulator, the total header size is 82 bytes for a data packet and the specific implementation of the 802.11 MAC employs a 44-byte RTS packet plus a 38-byte CTS and ACK packet. As a result, Figure 2 illustrates the optimal packet length versus SNR for these parameters and others parameterised by the retransmission limits and the value of α for QPSK modulation, when communicating over a non-dispersive uncorrelated Rayleigh channel. The SER P_s of QPSK was found for a Rayleigh channel in [16, Chapter 14.4.2] which is substituted into Equation 1, yielding Figure 2.

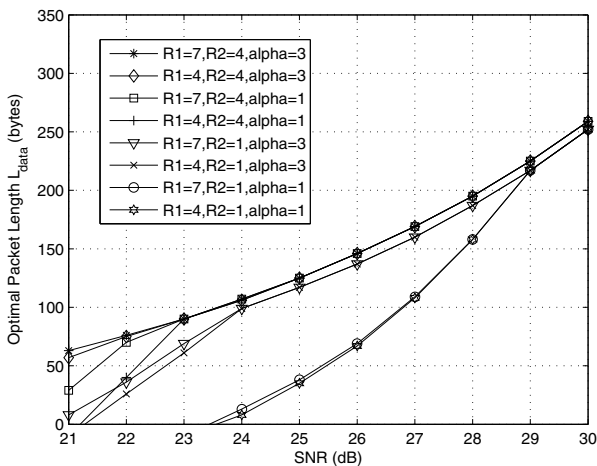


Fig. 2. Optimal packet length in bytes versus the channel SNR for $L_h = 82$ bytes, $L_{rts} = 44$ bytes, $L_{cts} = 38$ bytes and $L_{ack} = 38$ bytes, when communicating over a non-dispersive uncorrelated Rayleigh fading channel.

III. PERFORMANCE RESULTS

In order to demonstrate the benefits of the Fountain-coded scheme, we simulated an 802.11 WLAN using the NS2 simulator for file transfer employing LT codes. Again, QPSK modulation was employed for transmission over an uncorrelated non-dispersive Rayleigh fading channel. The Destination Sequenced Distance Vector (DSDV) routing protocol was adopted. We considered the transmission of a medium sized file constituted by 201 582 bytes and set the memory limit of the decoder to 102 400 bytes. Each packet was forwarded at the application layer at 0.5s intervals. The value of α was set to $\alpha = 3$, since this value would tolerate a maximum PLR of 0.75. Finally, the retransmission parameters were set to the default values of the 802.11 MAC specification, namely to $R_1 = 7$, $R_2 = 4$. All above values of different network layers are summarized in Table I.

TABLE I
PARAMETERS USED IN THE SIMULATION.

Application Layer	
α	3
Trans Interval	0.5s
Mem Limit	102400bytes
File Size	201582bytes
Transport Layer	UDP
Routing Protocol	DSDV
MAC	
R_1	7
R_2	4
PHY Modulation Scheme	QPSK
Channel	uncorrelated non-dispersive Rayleigh

Sufficiently long simulations were conducted in order to generate statistically relevant results. We chose four different packet lengths for each SNR, namely 16bytes, the relevant optimal length minus 32 bytes, the optimal length and the optimal length plus 32 bytes for comparison of the attainable transmission efficiency. Firstly, the PLR versus the SNR recorded at the application layer was illustrated in Figure 3, which verifies the accuracy of Equation 5 with the aid of Monte-Carlo simulations. Similarly, Figure 4 shows the relationship between the theoretical and Monte-Carlo simulation results quantified in terms of the channel occupancy ratio, which verifies the accuracy of Equation 10.

Finally, in Figure 5, the acronym ‘FEC-FT over UDP’ represents the file transfer using LT codes with the aid of the UDP transport layer protocol. The curve corresponding to the optimum packet length with ‘FEC-FT over UDP’ is always at the top, confirming that the FEC scheme has the highest transmission efficiency. The simulation results recorded for file transfer using the Transmission Control Protocol (TCP) (i.e., ‘FT over TCP’) are also provided in Figure 5. The transmission efficiency of ‘FT over TCP’ is zero, when the SNR is below 27dB, since the file transfer cannot succeed at all within a timeout of 5000s, which is sufficiently long for ‘FEC-FT over UDP’ to successfully complete the file transfer.

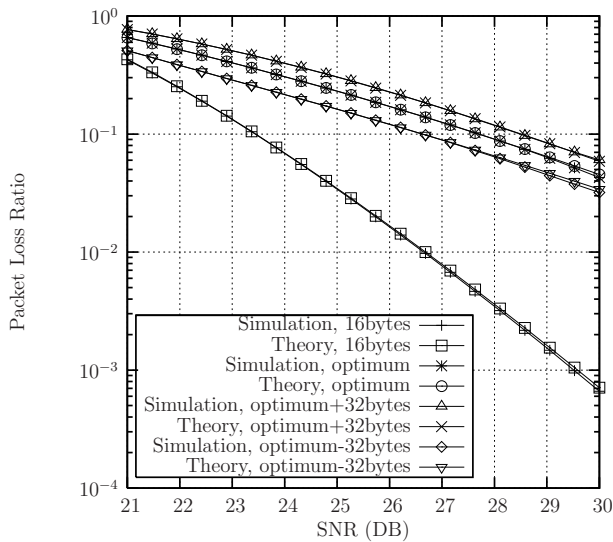


Fig. 3. PLR versus SNR at the application layer. The theoretical results were evaluated from Equation 5.

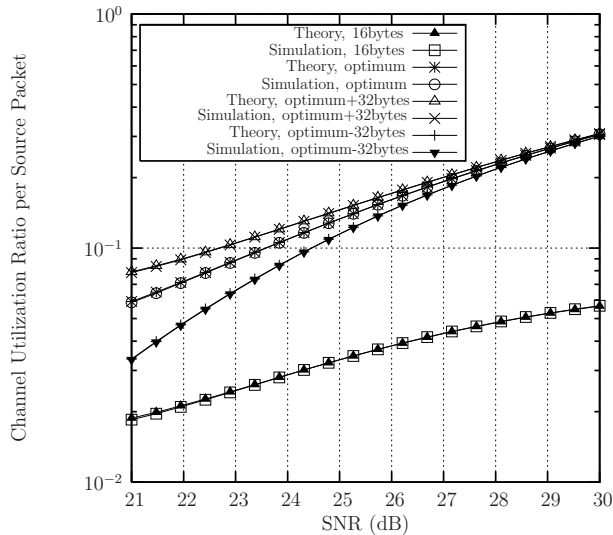


Fig. 4. Channel utilization ratio per source packet versus SNR for transmission over a non-dispersive uncorrelated Rayleigh channel. The theoretical results were evaluated from Equation 10.

IV. CONCLUSION

In conclusion, from the application layer's perspective, the retransmission and ACK mechanism of the 802.11 MAC is capable of reducing the PLR, but it imposes a substantial overhead. The appropriate choice of the packet length in the application layer indeed reduces the associated overhead. Fountain codes improve the successful file transfer probability even in hostile channel conditions. Compared to the file transfer regime operating without FEC over the classic TCP protocol, the proposed regime requires a lower threshold SNR for accomplishing a successful file transfer and hence enhances the transmission efficiency by about 50%.

REFERENCES

[1] IEEE Computer Society, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. New York, USA: IEEE

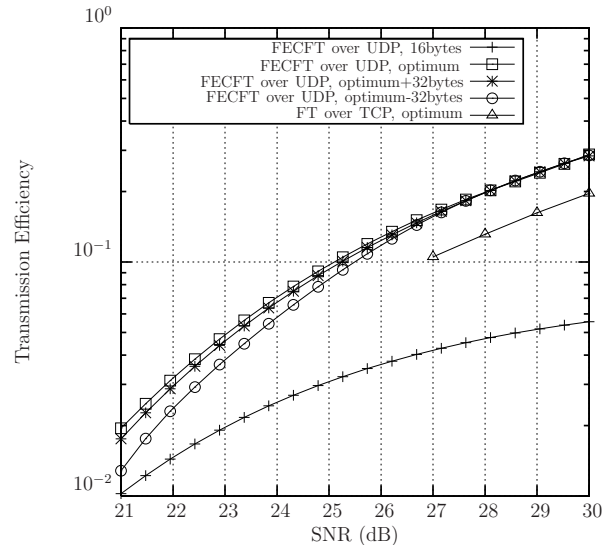


Fig. 5. Transmission Efficiency

Press, 2007.

[2] L. Hanzo, T.H. Liew, B.L. Yeap, *Turbo Coding, Turbo Equalisation and Space-Time Coding for Transmission over Fading Channels*. New York, USA: John Wiley IEEE Press, 2002.

[3] A.S. Tanenbaum, *Computer networks (4th Edition)*. New Jersey, USA: Prentice Hall PTR, 2002.

[4] D. Kliazovich and F. Granelli, "DAWL: A Delayed-ACK Scheme for MAC-Level Performance Enhancement of Wireless LANs," *Mobile Networks and Applications*, pp. 607–615, 10 2005.

[5] A. Basalamah, T. Sato, "Adaptive FEC Reliable Multicast MAC Protocol for WLAN," *IEEE VTC-2007 Fall*, pp. 244–248, Oct. 2007.

[6] A. Basalamah and T. Sato, "A Comparison of Packet-Level and Byte-Level Reliable FEC Multicast Protocols for WLANs," *IEEE GLOBE-COM*, pp. 4702–4707, Nov. 2007.

[7] E. Park, S. Han, H. Kim, K. Son, L. Jing, "Efficient Multicast Video Streaming for IPTV Service over WLAN Using CC-FEC," *ICICSE 2008 International Conference*, pp. 215–222, Jan. 2008.

[8] A. Moid, A.O. Fapojuwo, "Heuristics for Jointly Optimizing FEC and ARQ for Video Streaming over IEEE802.11 WLAN," *IEEE 2008 WCNC*, pp. 2141–2146, Mar. 2008.

[9] D.J.C. MacKay, "Fountain codes," *IEE Proceedings-Communications*, vol. 152, pp. 1062–1068, Dec. 2005.

[10] M. Luby, T. Stockhammer, M. Watson, "IPTV Systems, Standards and Architectures: Part II - Application Layer FEC In IPTV Services," *IEEE Communications Magazine*, vol. 46, pp. 94–101, May 2008.

[11] U. Demir, O. Aktas, "Raptor versus Reed Solomon forward error correction codes," *On International Symposium, Computer Networks*, pp. 264–269, June 2006.

[12] A. Shokrollahi, "Raptor codes," *IEEE Transactions, Information Theory*, vol. 52, pp. 2551–2567, June 2006.

[13] ETSI, *Digital Video Broadcasting (DVB); IP Datacast over DVB-H: Content Delivery Protocols*, Dec. 2006. TS 102 472 V1.2.1.

[14] ETSI, *Transport of MPEG 2 Transport Stream (TS) Based DVB Services over IP Based Networks*, Sep. 2007. DVB Document A086 Rev. 6.

[15] "Third Generation Partnership(3GPP) Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs(Release 7)." Technical Specification Group Services and System Aspects, June, 2008. Downloadable at <http://www.3gpp.org/FTP/Specs/html-info/26346.htm>.

[16] J.G. Proakis, *Digital Communication (Fourth Edition)*. Bei Jing, CHINA: Publishing House of Electronics Industry, 2001.