PAPER

Aggressive Packet Combining for Error Control in Wireless Networks

Yiu-Wing LEUNG[†], *Member*

SUMMARY In uplink data communication in wireless networks, a portable computer may retransmit a packet multiple times before the base station receives the correct one. Each retransmission consumes communication bandwidth and battery energy of the portable computer. Therefore, it is desirable to reduce the number of retransmissions. In this paper, we propose the *aggressive packet combining* scheme for this purpose. The base station executes this scheme to combine multiple erroneous copies as follows: (1) perform bit-by-bit majority voting on the erroneous copies to produce a combined packet, (2) identify the least reliable bits in this combined packet, and (3) search the correct bit pattern for these bits. Then the base station may recover the correct packet, thereby reducing the mean number of retransmissions. The proposed scheme has several advantages: (1) it is more powerful than the majority packet combining scheme, (2) it can complement many existing ARQ protocols to improve their performance, (3) it does not add additional bits to the packet and hence it does not consume extra bandwidth in the wireless channel, and (4) it is only executed by the base station and hence the portable transceiver can be kept simple. The simulation results show that the proposed scheme is more bandwidth-efficient and energy-efficient than the majority packet combining scheme.

key words: *wireless networks, error control, ARQ protocols, packet combining*

1. Introduction

The bit error rate of wireless channels is high [1], [2]. In uplink data communication [3], a portable computer may retransmit a packet multiple times before the base station receives the correct one. Each retransmission consumes communication bandwidth and transmission energy. The bandwidth is a scarce resource because many portable computers may be sharing a limited free-space spectrum. The energy is also a scarce resource because the portable computer is usually powered by battery. Therefore, it is desirable to reduce the number of retransmissions in order to utilize the bandwidth and battery energy efficiently. On the other hand, it is also desirable to keep the transceivers in the portable computers simple, so that wireless data communication can be more affordable and popular.

In many error control protocols, the receiver discards the erroneous packet and requests the transmitter for a retransmission. However, an erroneous packet may contain both erroneous bits and correct bits and hence it may still contain useful information. The receiver may be able to combine this information from multiple erroneous copies to recover the correct packet.

Wicker [4] proposed the *majority packet combining* scheme. This scheme performs bit-by-bit majority voting on the erroneous copies and then performs error detection on the resulting combined packet. If the combined packet is found to be correct, the receiver accepts it; otherwise, the receiver requests the transmitter for a retransmission. Figure 1 shows an example. This scheme can complement many existing ARQ protocols to improve their performance. In particular, Zhao, Sato and Kimura [5] applied this scheme for a point-to-multipoint ARQ protocol. In addition, Daraiseh and Baum [6], [7] proposed two generalizations of this scheme for a frequency-hop spread-spectrum communication system, in which every packet consists of multiple Reed-Solomon codewords and packet combining is done at the codeword level.

In this paper, we propose the *aggressive packet combining* scheme for error control in uplink data communication. This scheme is more powerful than the majority packet combining scheme, because it further exploits the information available in the erroneous copies to recover the correct packet. We study the properties of this scheme and evaluate its performance by computer simulation.

2. Aggressive Packet Combining

We refer to the portable computer and the base station as the transmitter and the receiver respectively. Each packet consists of N bits, and we number these bits such that the most significant bit is the first bit.

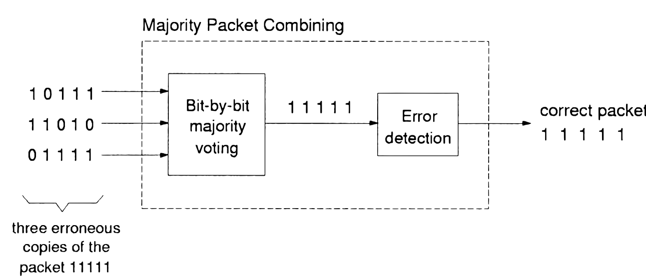


Fig. 1 Majority packet combining. In this example, it can recover the correct packet.

Manuscript received June 24, 1999.

Manuscript revised September 13, 1999.

[†]The author is with the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong.

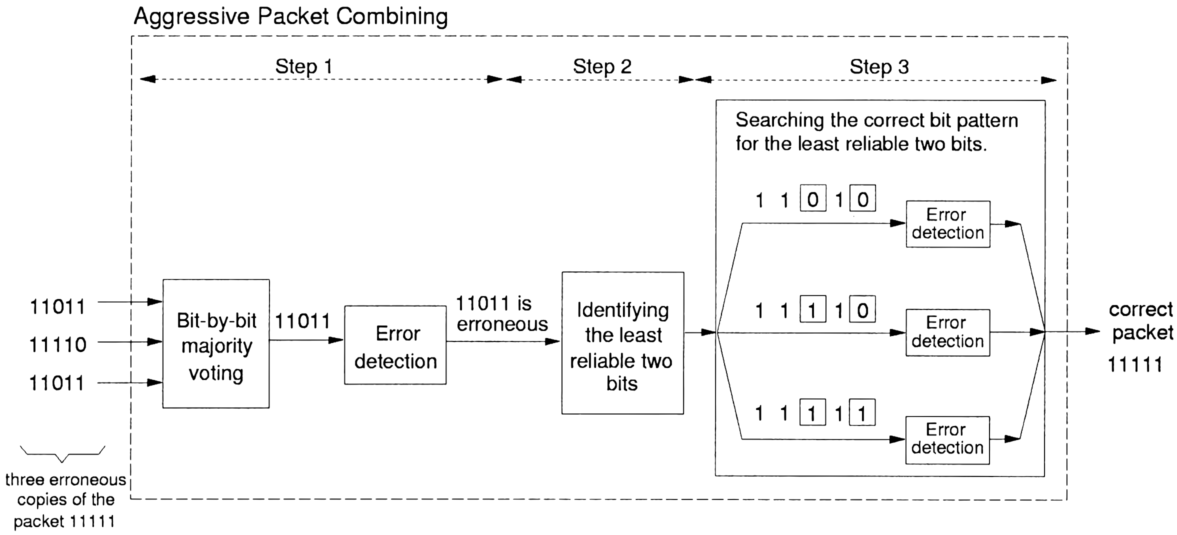


Fig. 2 Aggressive packet combining. In this example, it can recover the correct packet, but the majority packet combining scheme cannot.

2.1 Basic Scheme

In this subsection, we describe the aggressive packet combining scheme for the traditional ARQ protocols, in which every packet contains an error detection code. In Sect. 2.3, we will consider the other ARQ protocols.

To explain our main ideas, we consider the example shown in Fig. 2. The receiver has received three erroneous copies of the packet 11111. It executes the following three steps to combine them:

- In step 1, the receiver performs bit-by-bit majority voting on the erroneous copies to produce the combined packet 11011. Then it performs error detection and finds that the combined packet is erroneous.
- In step 2, the receiver identifies the least reliable two bits in the combined packet. The third bit and the fifth bit are obtained based on majority decisions, while the other bits are obtained based on total consensus. Therefore, it is more likely that the third bit or the fifth bit are erroneous. In other words, these two bits are the least reliable.
- In step 3, the receiver searches the correct bit pattern for the third and the fifth bits. These two bits are 01 in the combined but erroneous packet, and hence they may be either 00, or 10 or 11. The receiver replaces the two bits with each of these three bit patterns, and performs error detection on the resulting packet (see Fig. 2). Then, the receiver can get the correct packet 11111.

We generalize the above ideas as follows. Suppose the receiver can buffer M erroneous copies of a packet in a first-in-first-out manner. M is a design parameter. If M is larger, the scheme has a better performance but the receiver requires a larger buffer size. As we shall demonstrate in Sect. 3, M is small in practice (say, $M \leq 5$). In step 1, the receiver performs bit-by-bit majority voting on the erroneous copies in the buffer and then performs error detection on the resulting combined packet. If this combined packet is correct, the receiver accepts it and stops processing. In step 2, the receiver identifies the least reliable A bits in the combined packet. A is a design parameter. If A is larger, the scheme has a better performance but the receiver has to do more processing. As we shall discuss and demonstrate, A is chosen to be small in practice (say, $A \leq 10$). Among the ith bit of the erroneous copies, let $n_{i,0}$ be the number of 0 and $n_{i,1}$ be the number of 1. In addition, let $n_i = max\{n_{i,0}, n_{i,1}\}$ and $\mathbf{n} = [n_1, n_2, \cdots, n_N]$. Since the ith bit of the combined packet is obtained by majority voting, it is less reliable if $max\{n_{i,0}, n_{i,1}\}$ is smaller. To identify the least reliable A bits, the receiver identifies the smallest A elements of $\mathbf{n}$. In step 3, the receiver searches the correct bit pattern for these A bits. It replaces the A bits with each of the $2^A - 1$ possible bit patterns, and then performs error detection on the resulting packet. It continues searching until it finds a correct packet or it has searched all the $2^A - 1$ bit patterns. The details of the aggressive packet combining scheme are given as follows:

Step 1: Perform bit-by-bit majority voting on the erroneous copies as follows: the ith bit of the combined packet is 0 if $n_{i,0} > n_{i,1}$; it is 1 if $n_{i,1} > n_{i,0}$; and it is chosen randomly if $n_{i,0} = n_{i,1}$. Perform error detection on the combined packet. If it is correct, accept it and stop.

Step 2: Identify the smallest A elements of $\mathbf{n}$ in order to identify the least reliable A bits in the combined packet.

Step 3: Search the correct bit pattern for the least reliable A bits as follows. Replace the A bits with each of the $2^A - 1$ possible bit patterns, and then perform error detection on the resulting packet. If it is correct, accept it and stop; otherwise, repeat the search for the next bit pattern. If all the $2^A - 1$ bit patterns have been searched but the correct packet cannot be found, request the transmitter for a retransmission.

2.2 Properties

The aggressive packet combining scheme has several desirable properties:

(1) Step 1 is the same as the majority packet combining scheme. In steps 2 and 3, the proposed scheme further exploits the information available in the erroneous copies to recover the correct packet. Therefore, the proposed scheme is more powerful than the majority packet combining scheme.

(2) As the example in Fig. 2 demonstrates, the proposed scheme can perform some error corrections while it does not add additional bits to the packet. This is different from the traditional concept of forward error correction. Consequently, the proposed scheme does not consume extra bandwidth in the wireless channel. On the other hand, the proposed scheme can also be used to supplement the forward error correction methods to provide a better error correction capability, and the details are given in Sect. 2.3.

(3) The proposed scheme is only executed by the base station, and the portable computer is not required to do any additional processing. Therefore, the portable transceiver can be kept simple.

(4) The basic scheme can be generalized to complement various ARQ protocols, and the details are given in Sect. 2.3.

The aggressive packet combining scheme has two possible shortcomings, but these shortcomings should be acceptable in practice:

(1) In step 3, the receiver searches the correct bit pattern for the least reliable A bits. In the worse case, it has to search $2^A - 1$ bit patterns. Nevertheless, the processing time required can be small, because A is chosen to be small in practice, every search takes a very short time, and all the searches can be performed in parallel. Specifically, every search only involves an error detection. If cyclic redundancy check (CRC) is adopted [8], an error detection requires $N + D$ clock cycles, where D is the degree of the CRC generator polynomial. When $N = 1000$, $D = 32$, and the CRC hardware is operated at 200 MHz, an error detection requires 5.16 μs. Therefore, even if $A = 10$ and we search the bit patterns one after the other, the total searching time in the worst case is $(2^{10} - 1) \times 5.16\,\mu$s= 5.28 ms. This searching time is small compared with the packet transmission time (e.g., if the data rate is 64 kbps, the packet transmission time is 15.63 ms). In addition, each CRC detector can be realized using D shift registers and at most D exclusive OR gates [8], and hence it is feasible to implement many CRC detectors on the same chip. Then the receiver can search multiple bit patterns in parallel, so that the searching time can be further reduced. For the above example, if we search eight bit patterns in parallel, the total searching time in the worse case is reduced to 0.66 ms only, and it is much smaller than the packet transmission time 15.63 ms.

(2) In principle, among the $2^A - 1$ bit patterns, there may be bit patterns that are valid codewords but they are not the transmitted codeword. In practice, the probability of having an undetected error is negligibly small. For example, suppose A is chosen to be 10 and a 32-bit CRC [8] is used for error detection. For *the very extreme and worse case* when the ten least reliable bits and the other erroneous bits constitute an error burst of length greater than 32, the probability of having an undetected error is only $\frac{1}{2^{31}} \times 2^{10} = 4.8 \times 10^{-7}$. For the general cases, this probability is even smaller.

2.3 Generalization

The basic scheme can be generalized to complement many other ARQ protocols in which the transmitter retransmits the same packet to the receiver. We give three examples in the followings:

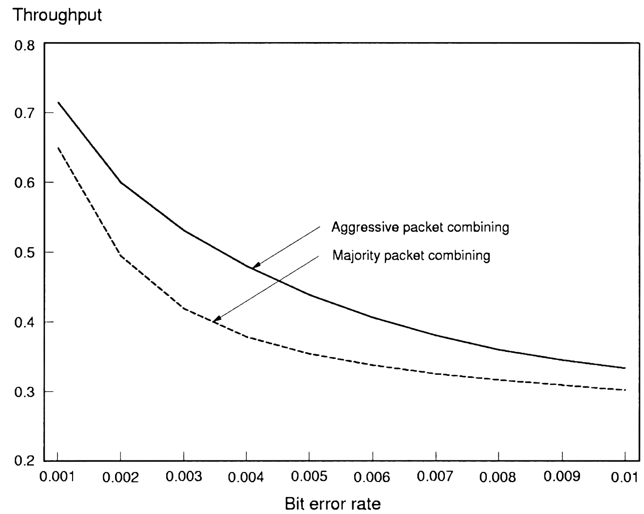
- *Multicopy ARQ Protocols* [9]: Every packet contains an error detection code. In each transmission or retransmission, the transmitter sends L copies of a packet to the receiver. If all these L copies are erroneous, the receiver executes the basic scheme to combine all the M erroneous copies in the buffer.
- *Type-I Hybrid ARQ Protocols* [2]: Every packet contains an error control code which has both error correction and detection capability. In each retransmission, the transmitter sends the same packet to the receiver. To combine the erroneous copies, we slightly modify the basic scheme as follows. In step 1, the receiver performs error correction and then error detection on the combined packet. In step 3, the receiver performs error correction and then error detection in each of the $2^A - 1$ searches.
- *Type-II Hybrid ARQ Protocols* [2]: Every data

unit is encoded into two packets: (1) a data packet that contains an error detection code and (2) a parity packet that, in combination with the data packet, does both error correction and detection. The transmitter sends these two packets in the following sequence: data packet, parity packet, data packet, parity packet, $\cdots$. To combine the erroneous copies, we slightly modify the basic scheme as follows. In step 1, the receiver applies bit-by-bit majority voting to obtain a combined data packet and a combined parity packet. Then it performs error correction on the combined data packet using the combined parity packet, and then performs error detection on the resulting packet. In step 2, the receiver identifies the least reliable A bits in the combined data packet and the combined parity packet. In step 3, in each of the $2^A - 1$ searches, the receiver performs error correction on the combined data packet using the combined parity packet, and then performs error detection on the resulting packet.

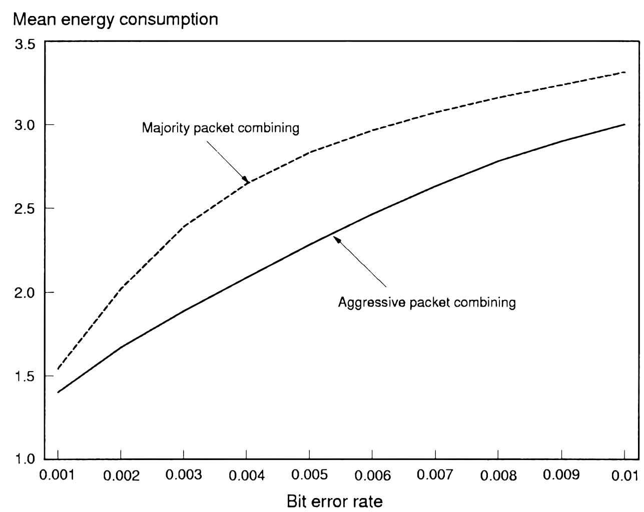
3. Simulation Results

We apply the aggressive packet combining scheme to complement the selective repeat protocol. We adopt a two-state Markov chain to model the channel error [10]. If the channel state is 0, the transmitted bit is correct; otherwise, it is erroneous. Given that the current channel state is 0 (1), we let $\alpha(\beta)$ be the probability that the next channel state is 0 (1). The bit error rate can be found to be $(1-\alpha)/(2-\alpha-\beta)$. We let $\beta = 0.5$, and we choose α to fulfill any given bit error rate (i.e., if the bit error rate is e, we choose $\alpha = (1-2e+e\beta)/(1-e)$). In the simulation, if the combined packet is different from the transmitted packet, this combined packet is detected to be erroneous. We let $N = 1000$, and let every transmission or retransmission consume one unit of energy. We measure the performance in terms of: (1) the normalized throughput and (2) the mean energy consumption (i.e., the mean energy consumed by the portable computer in transmitting and retransmitting a packet).

The typical bit error rate of a wireless channel ranges from 10^{-3} [1] to 10^{-2} [2]. In Fig. 3, we show the performance of the proposed scheme and the majority packet combining scheme over this range of bit error rate ($A = 10$ and $M = 4$). We see that the proposed scheme has a significantly better performance. When the bit error rate is 10^{-3}, the proposed scheme increases the throughput by 10.1% and reduces the mean energy consumption by 9.2%; when the bit error rate is 5×10^{-3}, the proposed scheme increases the throughput by 24.0% and reduces the mean energy consumption by 19.4%. Therefore, the proposed scheme is more bandwidth-efficient and energy-efficient than the majority packet combining scheme.



(a) Throughput.

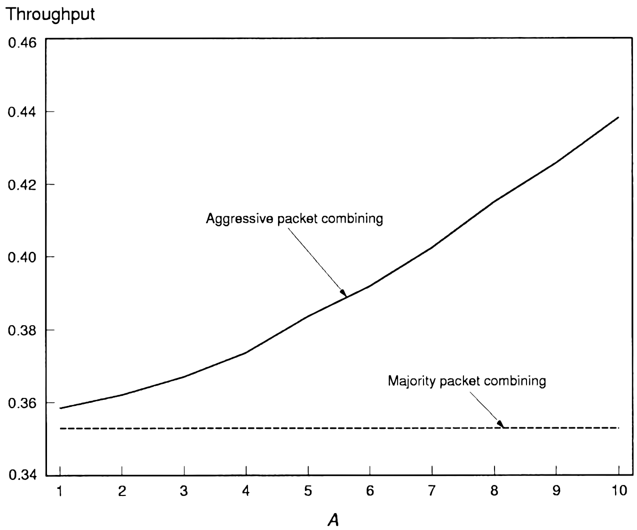


(b) Mean energy consumption.

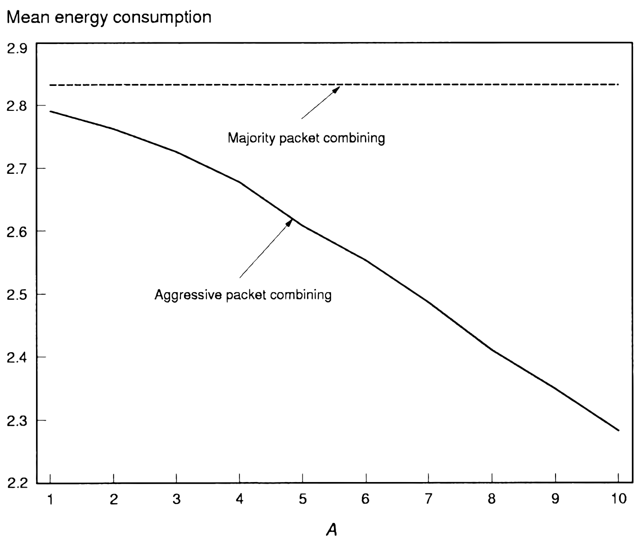
Fig. 3 Performance of aggressive packet combining and majority packet combining under different bit error rate; $A = 10$ and $M = 4$.

Figure 4 shows the performance of the proposed scheme versus the design parameter A (the bit error rate is 5×10^{-3} and $M = 4$). When A is larger, the performance is significantly better. Therefore, A should be chosen to be as large as the processing time is acceptable. When A is chosen to be 10, the proposed scheme can already increase the throughput by 24.0% and reduce the mean energy consumption by 19.4%. Nevertheless, the processing time required is still small (for the example discussed in Sect. 2.2, the processing time is 33.8% of the packet transmission time for sequential processing, and it is only 4.2% of the packet transmission time for parallel processing of eight bit patterns).

Figure 5 shows the performance of the proposed scheme versus the design parameter M (the bit error

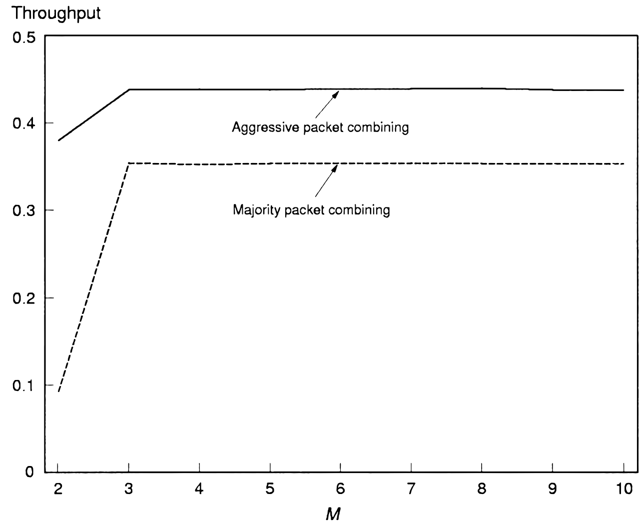


(a) Throughput.

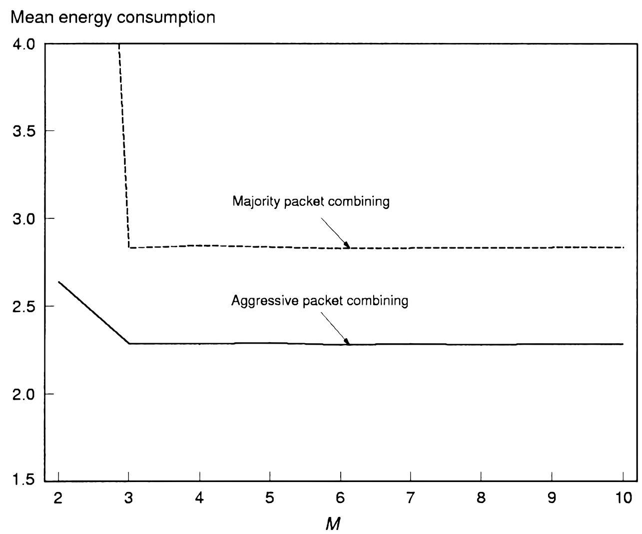


(b) Mean energy consumption.

Fig. 4 Selection of the design parameter A; bit error rate is 5×10^{-3} and $M = 4$.



(a) Throughput.



(b) Mean energy consumption.

Fig. 5 Selection of the design parameter M; bit error rate is 5×10^{-3} and $A = 10$.

rate is 5×10^{-3} and $A = 10$). We see that the performance increases rapidly with M for $M \leq 3$, but the performance is nearly insensitive to M for $M \geq 3$. In this case, we can choose $M = 3$ to attain a good performance using a small buffer size.

4. Conclusions

We proposed the *aggressive packet combining* scheme for error control in uplink data communication in wireless networks. The proposed scheme has several advantages: (1) it is more powerful than the majority packet combining scheme because it further exploits the information available in the erroneous copies to recover the correct packet, (2) it can complement many existing ARQ protocols to improve their performance, (3) it does not add additional bits to the packet and hence it does not consume extra bandwidth in the wireless channel, and (4) it is only executed by the base station and hence the portable transceiver can be kept simple. The numerical results show that the proposed scheme can result in a larger throughput and a smaller mean energy consumption than the majority packet combining scheme. In addition, the design parameter A should be chosen to be as large as the processing delay is acceptable, while it is sufficient to choose a small value for the design parameter M.

References

[1] O. Yuen, “Design trade-offs in cellular/PCS systems,” IEEE Commun. Mag., vol.34, no.9, pp.146–152, Sept. 1996.

[2] H. Liu, H. Ma, M.E. Zarki, and S. Gupta, “Error control

schemes for networks: An overview," Mobile Networks and Applications, vol.2, pp.167–182, 1997.
[3] K. Pahlavan and A.H. Levesque, "Wireless data communications," Proc. IEEE, vol.82, no.9, pp.1398–1430, Sept. 1994.
[4] S.B. Wicker, "Adaptive rate error control through the use of diversity combining and majority-logic decoding in a hybrid-ARQ protocol," IEEE Trans. Commun., vol.39, no.3, pp.380–385, March 1991.
[5] H. Zhao, T. Sato, and I. Kimura, "New go-back-N ARQ protocols for point-to-multipoint communications," IEICE Trans. Commun., vol.E77-B, no.8, pp.1013–1022, Aug. 1994.
[6] A.G.A. Daraiseh and C.W. Baum, "Methods for packet combining in HARQ systems over bursty channels," Mobile Networks and Applications, vol.2, pp.213–224, 1997.
[7] A.G.A. Daraiseh and C.W. Baum, "Packet combining in frequency-hop spread-spectrum communication systems," IEEE Trans. Commun., vol.46, no.1, pp.23–33, Jan. 1998.
[8] W. Stallings, Data and Computer Communications, pp.166–171, 5th ed., Prentice Hall, 1997.
[9] T.H. Lee, "Throughput performance of a class of continuous ARQ strategies for burst-error channels," IEEE Trans. Veh. Technol., vol.41, no.4, pp.380–386, Nov. 1992.
[10] E.N. Gilbert, "Capacity of burst noise channels," Bell Syst. Tech. J., vol.39, pp.1253–1256, 1960.

Yiu-Wing Leung received the B.Sc. and Ph.D. degrees from the Chinese University of Hong Kong in 1989 and 1992 respectively. He completed his Ph.D. research under the supervision of Prof. T.S. Yum. After graduation, he worked in the Hong Kong Polytechnic University. Since 1997, he has been an Associate Professor in the Department of Computer Science of the Hong Kong Baptist University, Hong Kong. He has a wide range of research interests, including computer and communication networks, internet computing, multimedia systems, and algorithmic and heuristic techniques. He has published over 40 journal papers in these areas. Since 1996, he has been a senior member of IEEE.