Computer Communications 23 (2000) 1548–1557

computer communications

www.elsevier.com/locate/comcom

Novel methods for the performance analysis of adaptive hybrid selective repeat ARQ

H. Jianhua[a,*], K.R. Subramanian[a], D. Donghua[b], W. Wei[a]

[a]*Network Technology Research Center, School of EEE, Nanyang Technological University, Singapore, Singapore 639798*
[b]*Electronic and Information Department, Huazhong University of Science and Technology, Wuhan 430074, People's Republic of China*

Received 17 September 1999; revised 28 March 2000; accepted 28 March 2000

Abstract

In this paper we present two novel approximation methods for the performance analysis of selective repeat (SR) ARQ protocol with any size of buffer. The methods are called sequential method (SM) and stable function method (SFM). Both SM and SFM methods provide highly accurate results as compared to other conventional methods of analysis. However, between SM and SFM methods, it is observed that SM is computationally more complex. Hence, SFM method is proposed for our analysis. Based on SFM, we present an ad hoc algorithm for an adaptive SR ARQ scheme to optimize the performance of SR ARQ. The performance of hybrid SR ARQ (an SR ARQ protocol combined with FEC) is also analyzed. It is concluded from a comparison of different methods of analyzing the SR ARQ protocol that the stable function method (SFM) provides a simple and novel approach for network analysis.

Keywords: SR ARQ; Hybrid SR ARQ; Broadband communication

1. Introduction

Real-World data communication is subject to errors caused by various sources. Protocol for achieving efficient and reliable communication between source and receiver is very important to ensure the communication quality. One widely used technique for handling errors at the data-link layer of data communication systems is the error detection incorporated with automatic-repeated-request (ARQ). The protocol may be categorized as:

1. Stop-and-wait (SW);
2. Go-back-N;
3. Selective-repeat (SR).

Much work has been done for analyzing these basic protocols [1,3,6]. Generally speaking, the SW protocol suffers from certain inefficiency due to the channel being idle between the transmission of the message and the reception of the ACK/NACK from the receiver. This inefficiency is particularly serious when the round-trip delay between the transmitter and the receiver is long compared to transmission time of a message. SR ARQ offers the best performance in terms of throughput among the three, although it has an increased requirement for buffers at the receiver. The buffers are needed to store those packets which are received out of sequence. Variations of the basic SR ARQ protocol have also been extensively studied [4,5,17].

In systems where the packet lengths are relatively large, and where the noise and/or interference levels are high, SR ARQ with only error detection results in a low throughput due to the large number of retransmissions required. Satellite networks and packet radio systems are examples of such systems [7,8]. In these instances, a combination of error detection and error correction can offer significant advantages over an error detection only system. This is called hybrid ARQ schemes. Hybrid ARQ schemes have been widely studied [2,16,18].

Generally, the analysis on throughput efficiency of SR ARQ protocols focuses on the case of infinite buffer [2,18], or on lower bound methods by approximation [1,9]. In the case of analysis with finite buffer, the buffer size is mostly assumed to be the number of times of W packets [1,9], where W is the number of packets that can be sent in a round-trip time. The performance analysis of SR ARQ protocol with different buffers appeared for the first time in Ref. [1]. The analysis provides only a lower bound performance. In the case of W being small, the above analysis methods can often provide a reasonable throughput

* Corresponding author. Tel.: +65-7904680.
E-mail address: hejianhua@hotmail.com (H. Jianhua).

0140-3664/00/$ - see front matter
PII: S0140-3664(00)00215-2

efficiency. However, in the case of broadband satellite networks and packet radio system, W is very high [7,8]. For a simple comparison, the throughput efficiency of Go-back-N ARQ is $1/(1 + W * (\rho/1 - \rho))$, while the throughput efficiency of SR ARQ with infinite buffer is $1 - \rho$, where ρ is packet error ratio. When W is large, the difference on throughput efficiency between Go-back-N and SR ARQ with infinite buffer is also very large. In such cases, efficient analytical methods are necessary for SR ARQ protocols to investigate the impact of the receiver buffer. In this paper we estimate the throughput efficiency of SR ARQ protocol with any finite buffer. The buffer size can vary from 1 to ∞. When buffer size is set to 1, the SR ARQ protocol becomes Go-back-N ARQ.

The paper is organized as follows. In Sections 2 and 3, we propose a sequential method (SM) and a stable function method (SFM) for accurate analysis on throughput efficiency. In Section 4, we present an algorithm to optimize the performance of pure SR ARQ protocol by continuously sending an optimum number of copies of a packet. The optimum number is adaptively changed with the physical conditions. In Section 5, we analyze the performance of a hybrid SR ARQ, which combines a simple concatenated coding system. SFM is compared with two conventional methods in Section 6. Numerical results on the comparison of the performance of pure SR ARQ with those of optimum SR ARQ and hybrid SR ARQ are also presented in this section. Section 7 provides our conclusions.

2. Sequential method for throughput calculation

There are some nomenclature used in the analysis,

η	throughput efficiency
ϵ	bit error ratio
ρ	packet error ratio
K_{f}	total number of bits per packet
R	transmission rate in bit/s
B	buffer size on the receiver side
T	round trip propagation delay in seconds
K	total number of packets transmitted in a communication

Using the above nomenclature, the window size W is defined as $W = \mathrm{Int}(R * T/K_{\mathrm{f}})$.

Before analyzing the protocols, we make the following assumptions: (1) there is always a packet waiting for transmission at the transmitter and as a result, the maximum possible throughput is achievable for the given ARQ scheme; (2) feedback channel is noiseless and the ACK for a particular packet transmission arrives at the transmitter before the time-out counter for that packet expires; and (3) the time-out counter is set to expire after the transmissions of exactly W packets.

On the receiver side, the received error free packet must be delivered to the upper layer in order. We say a packet is the head packet if the packet is with the minimal sequence number, among the packet that receiver is expected to receive. After limited retransmissions, if the head packet in the receiver buffer are not correctly received yet, the receiver buffer will overflow. New transmitted packets after the buffer overflows will be discarded even if they are received correctly. We assume that each packet will cause buffer overflow. The duration is a stochastic variable. In order to characterize the duration of buffer overflowing in quantity, we introduce a variable H_n, the number of buffer overflowing. Let $H_n(k)$ denote the number of buffer overflowing caused by the kth packet, $0 < k \le K$. $H_n(k)$ being equal to n means that, after the kth packet becomes the head packet, it still requires n retransmissions to be correctly received after it causes the buffer overflow. Once the kth packet enters the end of the receiver buffer, if any previous packets stored in the buffer causes the buffer overflow, the packet will have a chance to be corrected. Let $SH_n(k)$ denote the total number of retransmissions before the kth packet becomes the head packet.

Define η as the ratio of the time to send K packets in an error free channel, to the time to send K packets correctly in a real channel. Since every packet has only one chance to become the head packet, the summation of the time spent on the head packet will be equal to the whole time to send all packets correctly. Let $T_h(k)$ denote the time for the kth head packet to be correctly received by the upper layer. Then we get a formula for calculating the throughput.

$$\eta = \frac{K * (K_{\mathrm{f}}/R)}{\sum_{k=1}^{K} E[T_{\mathrm{h}}(k)]} \tag{1}$$

The time spent on a head packet to be correctly received is the average transmission time since it becomes the head packet, plus the time when the packet forces the receiver buffer overflow. $T_h(k)$ can be expressed in the following form:

$$\begin{aligned} T_{\mathrm{h}}(k) = {} & P_N[N_r(k) = 0, SH_n(k)] * \frac{K_{\mathrm{f}}}{R} \\ & + \sum_{j=1}^{\infty} P_N[N_r(k) = j, SH_n(k)] * (j+1) * \frac{K_{\mathrm{f}}}{R} \\ & + \sum_{j=1}^{\infty} P_H[H_n(k) = j] * j * \left(T - \frac{K_{\mathrm{f}}}{R}\right) \end{aligned} \tag{2}$$

where $P_N[N_r(k)]$ is the probability that the kth packet is correctly received after exactly $N_r(k)$ retransmissions; $P_N[N_r(k) = j, SH_n(k)]$ is the joint probability that when the kth packet becomes the head packet, it is correctly received after exactly $N_r(k) = j$ more retransmissions, under the condition of $SH_n(k)$ buffer overflowing caused by previously stored packets; $P_H[H_n(k)]$ is the probability that the kth packet causes the buffer overflow exactly $H_n(k)$ times.