

Analysis of Selective-Repeat ARQ via Matrix Signal-Flow Graphs

Kamtorn Ausavapattanakun and Aria Nosratinia, *Senior Member, IEEE*

Abstract—In this paper, we present a new method to analyze the throughput and delay of the selective-repeat (SR) automatic repeat-request (ARQ) protocol. Previous work on SR ARQ has concentrated on reliable feedback or two-state Markovian feedback errors. We solve a wider class of problems by characterizing both the forward and reverse channels by general hidden Markov models (HMMs). The moment-generating function (MGF) technique is used to find throughput and delay. To calculate the MGF, we construct *matrix* signal-flow graphs for the hidden Markov process. This procedure can be useful for a variety of other HMM problems, and is of interest by itself. Practical issues such as erasure errors and timeouts are included in our analyses, which are verified by extensive simulations.

Index Terms—Automatic repeat-request (ARQ), delay, hidden Markov model (HMM), selective-repeat (SR) protocol, throughput.

I. INTRODUCTION

AUTOMATIC repeat-request (ARQ) protocols are widely used in communication networks. In this paper, we analyze the throughput and delay of selective-repeat (SR) ARQ under very general conditions on the forward and reverse channels.

The SR ARQ has been previously analyzed under certain conditions. In particular, it is known that when the average packet-failure probability is ϵ , the throughput of the SR ARQ is $1 - \epsilon$. This simple answer is independent of channel dynamics, but is valid only when feedback is reliable [1]. When the feedback is unreliable, the analysis becomes complicated. Several works have modeled the feedback errors by random errors [2] or a simple two-state Markov process [3]–[5]. To date, SR ARQ has not been analyzed under more elaborate channel models.

This paper analyzes SR ARQ under unreliable feedback, using a general *hidden* Markov characterization for both the forward and reverse channels. Due to the generality of this approach, the extension of previous results from two-state Markov feedback to finite-state Markov feedback is also implied in our work.

In many practical scenarios, the channel state is not the same as the channel observation, leading to the hidden Markov

model (HMM).¹ In HMM, the channel quality is described by a (hidden) state that determines the error probability of the channel. In many cases, HMM is more accurate than the simple Markov model; for example, the HMM is better than simple Markov models in describing packet communication over the fading wireless channel [6]–[8]. In this paper, we use HMM models as overall first-order approximations for the behavior of ARQ packet communication channels, including the effects of the transmission medium (e.g., a wireless channel, or other physical layer channel) as well as coding and other effects present in-between the ARQ transmitter and receiver.

We use the moment-generating function (MGF) approach, and for calculation, we employ the flow-graph technique of Mason, which has been used for ARQ analysis in [1], [3], and [9]. However, the flow-graph methods in the existing literature cannot conveniently solve the multistate Markov or the hidden Markov problem. We construct, for the hidden Markov processes, an extension of Mason graphs with matrix-valued link labels. Flow graphs with matrix labels, called matrix signal-flow graphs (MSFGs), were introduced in 1957 for the purpose of multiterminal circuit analysis [10].

We use the MSFG to calculate the MGFs of two important random variables: transmission time (how many times a packet must be transmitted) and delay. From the generating functions, the average throughput and delay can be computed. We consider several practically important issues, e.g., we model feedback errors as erasures instead of simple “bit reversal.” Even though the bit-reversal model would make the analysis simpler, it is not practical, since common usage of a cyclic redundancy check (CRC) means that despite losses, acknowledgement (ACK) and negative ACK (NACK) are not mistaken for one another. We also analyze the effect of timeout policies on throughput and delay.

To summarize, the contribution and novelty of this paper is as follows. This paper generalizes the analysis of SR ARQ using hidden Markov modeling. We note that although Go-Back-N ARQ has been analyzed under HMM and bit-reversal feedback errors [11], SR ARQ analysis has been available only under much simpler models. We construct MSFGs for HMM to make the SR analysis tractable; our construction can be used for other HMM problems, as well. The analysis in the following augments the methods of [11] by characterizing erasure errors and including the effect of timeout.

II. CHANNEL MODEL

At time t , the status of each transmission, denoted by X_t , is a Bernoulli random variable taking values in $\mathcal{X} = \{0, 1\}$,

¹The Markov model is, trivially, a special case of the HMM.

Paper approved by R. Schober, the Editor for Detection, Equalization, and MIMO of the IEEE Communications Society. Manuscript received July 5, 2005; revised February 6, 2006 and April 26, 2006. This paper was presented in part at the Asilomar Conference on Circuits, Systems, and Computers, 2005.

The authors are with the Department of Electrical Engineering, The University of Texas at Dallas, Richardson, TX 75083-0688 USA (e-mail: kxa025000@utdallas.edu; aria@utdallas.edu).

Digital Object Identifier 10.1109/TCOMM.2006.885092

where value 0 denotes an error-free frame, and 1 means the frame is erroneous. The probability of error is denoted with ε , which is a function of channel condition. This channel condition is modeled by a multistate Markov process S_t , with states $\mathcal{S} = \{1, \dots, K\}$ and probability transition matrix \mathbf{P} . Each state $S_t = j$ gives rise to a different error probability ε_j . We denote the set of all such probabilities $\boldsymbol{\varepsilon} = [\varepsilon_1, \dots, \varepsilon_K]$. The process X_t , which is driven by the Markov process S_t , is called a *hidden Markov process*, and is characterized by the quartet $\{\mathcal{S}, \mathcal{X}, \mathbf{P}, \boldsymbol{\varepsilon}\}$.

For the purposes of our analysis, it is useful to define the joint probabilities of channel state and observation at time t , given the channel state at time $t - 1$

$$\begin{aligned} \Pr(S_t = j, X_t = 1 | S_{t-1} = i) \\ &= \Pr(S_t = j | S_{t-1} = i) \Pr(X_t = 1 | S_t = j, S_{t-1} = i) \\ &= \Pr(S_t = j | S_{t-1} = i) \Pr(X_t = 1 | S_t = j) \\ &= p_{ij} \varepsilon_j \end{aligned}$$

which we can collect into a new matrix of transition probabilities $\mathbf{P}_1 = \mathbf{P} \cdot \text{diag}\{\boldsymbol{\varepsilon}\}$. Similarly, we can define another matrix of transition probabilities $\mathbf{P}_0 = \mathbf{P} \cdot \text{diag}\{\mathbf{1} - \boldsymbol{\varepsilon}\}$. The entries in matrices \mathbf{P}_0 and \mathbf{P}_1 are state-transition probabilities when viewed jointly with (conditional) channel observations. Note that $\mathbf{P}_0 + \mathbf{P}_1 = \mathbf{P}$. The HMM can now be characterized by $\{\mathcal{S}, \mathcal{X}, \mathbf{P}_0, \mathbf{P}_1\}$, which is useful for the purposes of analysis.

Now consider a two-way channel whose forward link $\{\mathcal{S}^{(f)}, \mathcal{X}^{(f)}, \mathbf{P}_0^{(f)}, \mathbf{P}_1^{(f)}\}$ and reverse link $\{\mathcal{S}^{(r)}, \mathcal{X}^{(r)}, \mathbf{P}_0^{(r)}, \mathbf{P}_1^{(r)}\}$ are mutually independent. The composite channel states are $\mathcal{S}^{(c)} = \mathcal{S}^{(f)} \times \mathcal{S}^{(r)}$, the Cartesian product of forward and reverse states. The combined observation set is $\mathcal{X}^{(c)} = \mathcal{X}^{(f)} \times \mathcal{X}^{(r)} = \{00, 01, 10, 11\}$, where, for example, $X_t^{(c)} = 00$ means both forward and reverse channels are good, while $X_t^{(c)} = 01$ means the reverse channel is erroneous. For $X_t^{(c)} = 11$, the joint probability of the combined observation and the composite state at time t , given the composite state at time $t - 1$, is

$$\begin{aligned} \Pr\left(S_t^{(c)} = (j, m), X_t^{(c)} = 11 | S_{t-1}^{(c)} = (i, k)\right) \\ &= \Pr\left(S_t^{(f)} = j, S_t^{(r)} = m, X_t^{(f)} = 1, \right. \\ &\quad \left. X_t^{(r)} = 1 | S_{t-1}^{(f)} = i, S_{t-1}^{(r)} = k\right) \\ &= \Pr\left(S_t^{(f)} = j, X_t^{(f)} = 1 | S_{t-1}^{(f)} = i\right) \\ &\quad \times \Pr\left(S_t^{(r)} = m, S_t^{(r)} = 1 | S_{t-1}^{(r)} = k\right) \\ &= \left(p_{ij}^{(f)} \varepsilon_j^{(f)}\right) \cdot \left(p_{km}^{(r)} \varepsilon_m^{(r)}\right). \end{aligned}$$

Or, in compact notation, $\mathbf{P}_{11}^{(c)} = \mathbf{P}_1^{(f)} \otimes \mathbf{P}_1^{(r)}$, where \otimes denotes the Kronecker product of matrices. The probability matrices for other observations can be found in the same way. Finally, the composite channel is characterized by $\{\mathcal{S}^{(c)}, \mathcal{X}^{(c)}, \mathbf{P}_{00}^{(c)}, \mathbf{P}_{01}^{(c)}, \mathbf{P}_{10}^{(c)}, \mathbf{P}_{11}^{(c)}\}$, where $\mathcal{S}^{(c)} = \mathcal{S}^{(f)} \times \mathcal{S}^{(r)}$, $\mathcal{X}^{(c)} = \mathcal{X}^{(f)} \times \mathcal{X}^{(r)}$, and each of the observation probability matrices is described by a Kronecker product, i.e., $\mathbf{P}_{ij}^{(c)} = \mathbf{P}_i^{(f)} \otimes \mathbf{P}_j^{(r)}$ for $i, j = 0, 1$ (see also [12]).

III. ANALYSIS OF SR ARQ

In this section, we calculate the throughput and delay of SR ARQ in channels described in the previous section. We assume erasure errors in the reverse channel, i.e., an ACK cannot be decoded as a NACK, and vice versa.

A. Protocol Description

SR ARQ allows the receiver to accept frames out of order. The out-of-order frames will be stored in a buffer, sorted, and passed to higher layers in the correct order. For the purposes of this analysis, we disregard buffer overflows. The feedback consists of ACKs and NACKs for error-free frames and erroneous frames, respectively. The round-trip time is k , i.e., it takes $k - 1$ time slots between transmission of a frame and receipt of its feedback (ACK or NACK).

At the transmitter, a timeout mechanism is used to prevent deadlock. When a frame is (re)transmitted, the timeout associated with this frame is set to T . If the timeout expires and no acknowledgment is received, the frame will be retransmitted. Clearly, the timeout has to be greater than or equal to the round-trip time ($T \geq k$).

We assume ACK/NACK will include the information about all correctly received frames. So the frame whose ACK is lost will be acknowledged by subsequent ACKs/NACKs. If the succeeding ACKs/NACKs are successfully received before timer expiration, the frame will not be retransmitted. When a frame is lost and its NACK is received, the frame will be retransmitted immediately. If the NACK is also lost, the frame will be retransmitted after the timer expires.

In practice, ACK and NACK do not update the transmitter about the status of any previous frames. A frame whose ACK is lost has to always be retransmitted. We observe that the timeout has negligible effect on the throughput, but it increases the delay, so to minimize the delay, the timeout should be as small as possible ($T = k$). The performance of SR variations are compared in the next section.

B. MSFGs

A signal-flow graph [13] is a diagram of directed branches connecting a set of nodes. The graph also represents a system of equations. The nodes are variables in the equations and the branch labels, also known as branch transmissions, represent relationships among the variables. To simplify the flow graphs, there are three basic equivalences known as parallel, series, and self-loop. A thorough discussion of signal-flow graphs can be found in [14].

Scalar-flow graphs have been used to find the MGF of transmission and delay times in throughput and delay analysis, respectively [1], [3], [9]. The transmission time (τ) is defined as the number of frames being transmitted per a successful frame, while the delay time (D) is defined as the time from when a frame is first transmitted to when its ACK is received. Both transmission and delay times are discrete random variables with positive integer outcomes. Using the MGF, the expected values can be calculated and the throughput is the reciprocal of the transmission time.

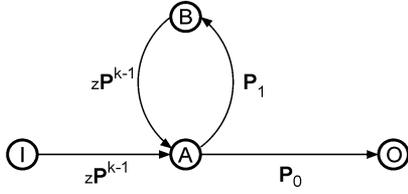


Fig. 1. Matrix flow graph for throughput analysis of SR protocol in reliable feedback.

We briefly review the basic technique for building our flow graphs. The graph nodes correspond to the states of the transmitter. One input node (I) represents the start of transmission, and one output node (O) represents correct reception of acknowledgment. Other nodes represent intermediate states. As events in the network unfold, the transmitter goes from one state to the other. Each state transition is accompanied with a certain value for n , the random variable of interest, and a probability p , which together appear in the branch gain pz^n . The input–output gain of the entire graph is, thus, a polynomial in z , whose coefficients are the probabilities of corresponding values of n . This polynomial is equivalent to $E[z^n]$, the MGF for n .

Scalar-flow graphs are useful for two-state Markov channels, but for multistate models and HMMs, the flow graphs become prohibitively complicated. To streamline the analysis, we propose labeling the branches with observation probability matrices. Flow graphs with matrix branch transmissions and vector node values are called MSFGs. In this method, the matrix gain of the graph is calculated using the usual basic operations, then the desired MGF is calculated by pre- and postmultiplications of row and column vectors, respectively, as shown in the following.

To demonstrate the MSFG methodology, we compute the throughput of SR ARQ in HMM with noiseless feedback. Let \mathbf{P}_0 and \mathbf{P}_1 , respectively, be the success and error probability matrices of an HMM. $\mathbf{P} = \mathbf{P}_0 + \mathbf{P}_1$ is the state transition-probability matrix. Let k be the round-trip time, so that an acknowledgment will be received $k - 1$ time slots after a frame is sent.

Fig. 1 shows the matrix flow graph for SR ARQ, assuming error-free feedback. In this figure, node I represents transmission of a new frame. Node A represents receipt of a feedback. The feedback can either be an ACK (transition to node O) or a NACK (transition to node B). At node O, an ACK is received and the frame will exit the system. At node B, the lost frame will be retransmitted, so the loop between node A and B represents retransmission of the erroneous frame until it is correctly received. Using basic node reduction, the matrix-generating function of transmission time will be

$$\begin{aligned}\Phi_\tau(z) &= z\mathbf{P}^{k-1}(\mathbf{I} - z\mathbf{P}_1\mathbf{P}^{k-1})^{-1}\mathbf{P}_0 \\ &= z(\mathbf{I} - z\mathbf{P}^{k-1}\mathbf{P}_1)^{-1}\mathbf{P}^{k-1}\mathbf{P}_0\end{aligned}$$

where \mathbf{I} is the identity matrix. Let π_I be the probability vector of transmitting a new frame. In this simple case, $\pi_I = \pi\mathbf{P}_0$,

where π is the stationary vector of \mathbf{P} . The stationary vector (π) can be found by solving the following system of equations:

$$\begin{aligned}\pi\mathbf{P} &= \pi \\ \pi\mathbf{1} &= 1\end{aligned}\quad (1)$$

where $\mathbf{1}$ is a column vector of ones. Let ϵ be the frame-error rate. In this scenario, we have $\epsilon = \pi\mathbf{P}_1\mathbf{1}$ and $1 - \epsilon = \pi\mathbf{P}_0\mathbf{1}$. The generating function can be calculated by left- and right-multiplying the matrix-generating function with the input row vector and the column vector of ones

$$\begin{aligned}\phi_\tau(z) &= \frac{\pi_I\Phi_\tau(z)\mathbf{1}}{\pi_I\mathbf{1}} \\ &= \frac{1}{1 - \epsilon}\pi\mathbf{P}_0\Phi_\tau(z)\mathbf{1}.\end{aligned}\quad (2)$$

The average transmission time ($\bar{\tau}$) can be found by evaluating the derivative of $\phi_\tau(z)$ at $z = 1$. Substituting and taking the derivative, we get

$$\begin{aligned}\bar{\tau} &= \frac{1}{1 - \epsilon}\pi\mathbf{P}_0 \left[(\mathbf{I} - \mathbf{P}^{k-1}\mathbf{P}_1)^{-1} + (\mathbf{I} - \mathbf{P}^{k-1}\mathbf{P}_1)^{-1} \right. \\ &\quad \left. \times \mathbf{P}^{k-1}\mathbf{P}_1(\mathbf{I} - \mathbf{P}^{k-1}\mathbf{P}_1)^{-1} \right] \mathbf{P}^{k-1}\mathbf{P}_0\mathbf{1} \\ &= \frac{1}{1 - \epsilon}\pi\mathbf{P}_0(\mathbf{I} - \mathbf{P}^{k-1}\mathbf{P}_1)^{-1}(\mathbf{I} - \mathbf{P}^{k-1}\mathbf{P}_1)^{-1}\mathbf{P}^{k-1}\mathbf{P}_0\mathbf{1} \\ &= \frac{1}{1 - \epsilon}\pi\mathbf{P}^{k-1}\mathbf{P}_0(\mathbf{I} - \mathbf{P}^{k-1}\mathbf{P}_1)^{-1} \\ &\quad \times (\mathbf{I} - \mathbf{P}^{k-1}\mathbf{P}_1)^{-1}\mathbf{P}^{k-1}\mathbf{P}_0\mathbf{1} \\ &= \frac{1}{1 - \epsilon}\end{aligned}$$

where we have used the following identity for a square matrix \mathbf{A} :

$$\frac{d}{dz}(\mathbf{I} - \mathbf{A}z)^{-1} = (\mathbf{I} - \mathbf{A}z)^{-1}\mathbf{A}(\mathbf{I} - \mathbf{A}z)^{-1}.$$

Throughput is the reciprocal of the average transmission time, thus $\eta = 1/\bar{\tau} = 1 - \epsilon$. The throughput of SR ARQ under HMM is, therefore, similar to the well-known previous results in k -order Markov channels [1].

C. Performance Evaluation in Unreliable Feedback

For calculating the throughput, consider the composite channel $\{\mathcal{S}, \mathcal{X}, \mathbf{P}_{00}, \mathbf{P}_{01}, \mathbf{P}_{10}, \mathbf{P}_{11}\}$. For simplicity, we define

$$\begin{aligned}\mathbf{P}_{0x} &= \mathbf{P}_{00} + \mathbf{P}_{01} & \mathbf{P}_{1x} &= \mathbf{P}_{10} + \mathbf{P}_{11} \\ \mathbf{P}_{x0} &= \mathbf{P}_{00} + \mathbf{P}_{10} & \mathbf{P}_{x1} &= \mathbf{P}_{01} + \mathbf{P}_{11}.\end{aligned}$$

So \mathbf{P}_{1x} is the probability matrix of error in the forward channel, and \mathbf{P}_{x0} is the probability matrix of success in the reverse channel.

The flow graph in Fig. 2 describes the operation of SR ARQ, where the timeout greater than the round-trip time ($T > k$). Node I represents that a new frame is transmitted. After sending the new frame, its feedback will be received $k - 1$ time slots

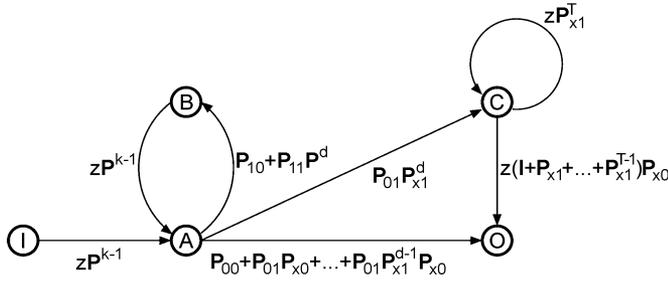


Fig. 2. Simplified matrix-flow graph for throughput analysis of SR protocol in unreliable feedback with $T > k$.

later. This state is represented by node A. There are now three possibilities.

First, if the feedback is an error-free ACK (with probability \mathbf{P}_{00}), or if it is an erroneous ACK but an error-free ACK/NACK is received before timer expiration (with probability $\mathbf{P}_{01}\mathbf{P}_{x0} + \dots + \mathbf{P}_{01}\mathbf{P}_{x1}^{d-1}\mathbf{P}_{x0}$), the system will transit to state O and the frame will be removed from the system.

Second, if the feedback is an erroneous ACK and the timer expires before receiving any error-free ACKs/NACKs, the system will transit to state C, the frame will be retransmitted, and the timeout will be reset. The frame will then be acknowledged when a succeeding ACK/NACK is correctly received. Otherwise, the frame will be retransmitted again (the self-loop at state C).

Finally, if the feedback is a error-free NACK (with probability \mathbf{P}_{10}) or an erroneous NACK and the timer expires (with probability $\mathbf{P}_{11}\mathbf{P}^d$), the system goes to state B, where the lost frame will be retransmitted, so that the loop between states A and B represents retransmission of the erroneous frame until it is correctly received. The matrix-generating function (the input–output relationship) is

$$\begin{aligned} \Phi_\tau(z) &= z\mathbf{P}^{k-1}(\mathbf{I} - z\mathbf{P}_{10}\mathbf{P}^{k-1} - z\mathbf{P}_{11}\mathbf{P}^{T-1})^{-1} \\ &\times \left[\mathbf{P}_{00} + \mathbf{P}_{01} \left(\sum_{j=0}^{d-1} \mathbf{P}_{x1}^j \right) \mathbf{P}_{x0} \right. \\ &\quad \left. + \mathbf{P}_{01}\mathbf{P}_{x1}^d (\mathbf{I} - z\mathbf{P}_{x1}^T)^{-1} z \left(\sum_{i=0}^{T-1} \mathbf{P}_{x1}^i \right) \mathbf{P}_{x0} \right]. \end{aligned}$$

In the special case of $T = k$, the matrix-generating function is simplified to

$$\begin{aligned} \Phi_\tau(z)|_{T=k} &= z\mathbf{P}^{k-1}(\mathbf{I} - z\mathbf{P}_{1x}\mathbf{P}^{k-1})^{-1} \\ &\times \left[\mathbf{P}_{00} + \mathbf{P}_{01} (\mathbf{I} - z\mathbf{P}_{x1}^T)^{-1} z \left(\sum_{i=0}^{T-1} \mathbf{P}_{x1}^i \right) \mathbf{P}_{x0} \right]. \end{aligned}$$

As mentioned earlier, to calculate the desired MGF, one also needs a vector probability; in this case, π_I , the probability vector of transmitting a new frame. This is achieved by solving a system of equations involving several vector probabilities, which are derived as follows.

Referring to Fig. 2, a frame will be transmitted only at states I, B, and C, which represent, respectively, transmission of a new

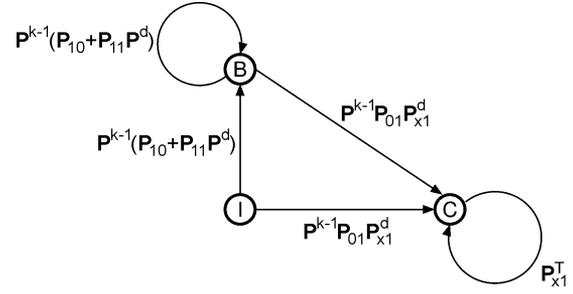


Fig. 3. Finite-state machine to derive the probability vector of transmitting a new frame.

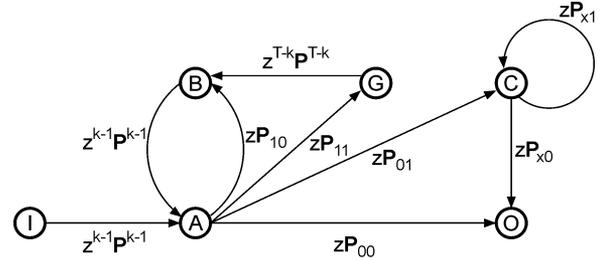


Fig. 4. Matrix-flow graph for delay analysis of SR protocol in unreliable feedback.

frame, retransmission of an erroneous frame, and retransmission of a frame that was correctly received, but its timer expires. We can simplify the graph by setting $z = 1$ to consist of only these nodes, as shown in Fig. 3. Let π_I , π_B , and π_C be the probability vectors of states I, B, and C, respectively. These probability vectors can be found by solving the following system of equations:

$$\begin{aligned} \pi_B &= (\pi_I + \pi_B)\mathbf{P}^{k-1}(\mathbf{P}_{10} + \mathbf{P}_{11}\mathbf{P}^{T-k}) \\ \pi_C &= (\pi_I + \pi_B)\mathbf{P}^{k-1}\mathbf{P}_{01}\mathbf{P}_{x1}^{T-k} + \pi_C\mathbf{P}_{x1}^T \\ \pi &= \pi_I + \pi_B + \pi_C \end{aligned} \quad (3)$$

where the last equation comes from the fact that the transmitter always has a frame to transmit, and π is the stationary vector of the state-transition matrix. Solving for π_I from the system (3), the generating function of transmission time (ϕ_τ) can be derived by (2). The throughput can, therefore, be computed by

$$\eta = 1/\bar{\tau} = 1/\phi'_\tau(1).$$

We now proceed with delay analysis. The matrix-flow graph for delay analysis is shown in Fig. 4. Nodes I, A, B, and O represent the same states as in the throughput analysis. The transmitter receives a feedback at node A. There are four possibilities, an error-free ACK (transition to state O), an erroneous ACK (transition to state C), an error-free NACK (transition to state B), and an erroneous NACK (transition to state G). Node G represents the case where a NACK is lost and the transmitter is waiting for timeout, and node C represents that an ACK is lost and the frame will be acknowledged by a subsequent ACK/NACK. The self-loop at node C represent the delay from losing subsequent ACKs/NACKs. When a NACK

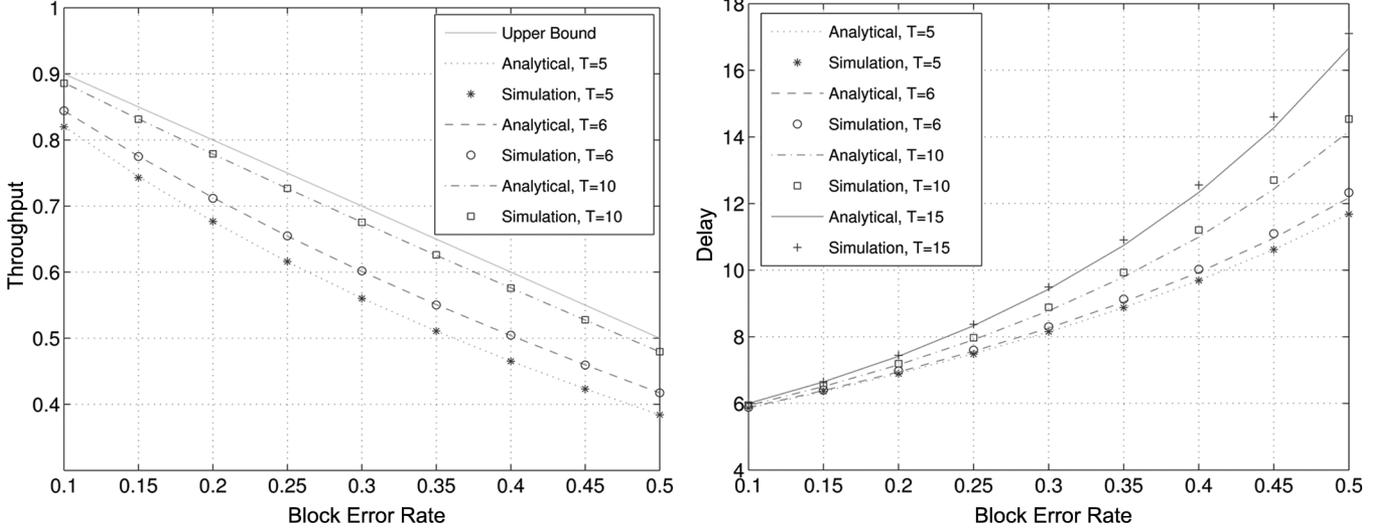


Fig. 5. Throughput η , and delay D , versus block-error rate ϵ , in Markov errors for $r = 0.3$ and $k = 5$.

is lost, the corresponding frame cannot be retransmitted immediately, but it will be retransmitted after the timer expires, which involves a delay (transition from node G to node B). Using basic manipulations, the matrix-generating function of delay will be

$$\begin{aligned} \Phi_D(z) &= z^{k-1} \mathbf{P}^{k-1} (\mathbf{I} - z^k \mathbf{P}_{10} \mathbf{P}^{k-1} - z^T \mathbf{P}_{11} \mathbf{P}^{T-1})^{-1} \\ &\quad \times [z \mathbf{P}_{00} + z \mathbf{P}_{01} (\mathbf{I} - z \mathbf{P}_{x1})^{-1} z \mathbf{P}_{x0}] \\ &= \mathbf{P}^{k-1} (\mathbf{I} - z^k \mathbf{P}_{10} \mathbf{P}^{k-1} - z^T \mathbf{P}_{11} \mathbf{P}^{T-1})^{-1} \\ &\quad \times [z^k \mathbf{P}_{00} + z^{k+1} \mathbf{P}_{01} (\mathbf{I} - z \mathbf{P}_{x1})^{-1} \mathbf{P}_{x0}]. \end{aligned}$$

The generating function of delay can be computed as

$$\phi_D(z) = \frac{\pi_I \Phi_D(z) \mathbf{1}}{\pi_I \mathbf{1}}$$

where π_I can be also computed from the system (3). Finally, the average delay will be the derivative of $\phi_D(z)$ at $z = 1$

$$\begin{aligned} \bar{D} &= \left. \frac{d}{dz} \phi_D(z) \right|_{z=1} \\ &= \frac{1}{\pi_I \mathbf{1}} \pi_I \left\{ \mathbf{P}^{k-1} (\mathbf{I} - \mathbf{P}_{10} \mathbf{P}^{k-1} - \mathbf{P}_{11} \mathbf{P}^{T-1})^{-1} \right. \\ &\quad \times [k \mathbf{P}_{10} \mathbf{P}^{k-1} + t \mathbf{P}_{11} \mathbf{P}^{T-1}] \\ &\quad \times (\mathbf{I} - \mathbf{P}_{10} \mathbf{P}^{k-1} - \mathbf{P}_{11} \mathbf{P}^{T-1})^{-1} \\ &\quad \times [\mathbf{P}_{00} + \mathbf{P}_{01} (\mathbf{I} - \mathbf{P}_{x1})^{-1} \mathbf{P}_{x0}] \\ &\quad + \mathbf{P}^{k-1} (\mathbf{I} - \mathbf{P}_{10} \mathbf{P}^{k-1} - \mathbf{P}_{11} \mathbf{P}^{T-1})^{-1} \\ &\quad \times [k \mathbf{P}_{00} + (k+1) \mathbf{P}_{01} (\mathbf{I} - \mathbf{P}_{x1})^{-1} \mathbf{P}_{x0} \\ &\quad \left. + \mathbf{P}_{01} (\mathbf{I} - \mathbf{P}_{x1})^{-1} \mathbf{P}_{x1} (\mathbf{I} - \mathbf{P}_{x1})^{-1} \mathbf{P}_{x0} \right\} \mathbf{1}. \end{aligned}$$

IV. NUMERICAL RESULTS AND DISCUSSION

In this section, numerical results are computed for the channels with two states (good and bad). Let the state-transition matrix be

$$\mathbf{P} = \begin{bmatrix} 1-q & q \\ r & 1-r \end{bmatrix}$$

where the first row and column corresponds to the good state (G), and the second row and column to the bad state (B). From the transition matrix, the stationary probability vector can be found using (1). The probability of block error (ϵ) can be computed from the stationary vector and the probability of error in each state (ϵ_G and ϵ_B). We will, therefore, use r, ϵ, ϵ_G , and ϵ_B as parameters of the forward channel. Note that $1/r$ represents the average error burst. We assume that the reverse channel has the same parameters as the forward channel.

The two-state Markov channel is a special case of HMM, where ϵ_G and ϵ_B equal 0 and 1, respectively. The throughput and delay of the SR protocol in the Markov channel is shown in Fig. 5. As expected, when the timer (T) is increased, both throughput and delay are higher. The throughput is upper bounded by $(1 - \epsilon)$ as $T \rightarrow \infty$; naturally, there is no such bound for delay.

The analytical and simulation results agree for the throughput. Simulation and analysis also agree for delay, except for a small difference in the case where the block-error rates are high. The difference comes from the fact that a retransmitted frame might need to wait in the retransmission queue. A frame will be retransmitted if its NACK is received or its timer expires, but there is a possibility (especially when the block-error rates are high) that a NACK is received for a frame at the same time that another frame's timer expires. In this case, one frame will be retransmitted, and the other frame will wait in the queue.

To show that this was the only source of discrepancy, we analyzed the SR protocol without NACK. When an erroneous frame is received, feedback is sent to update the transmitter

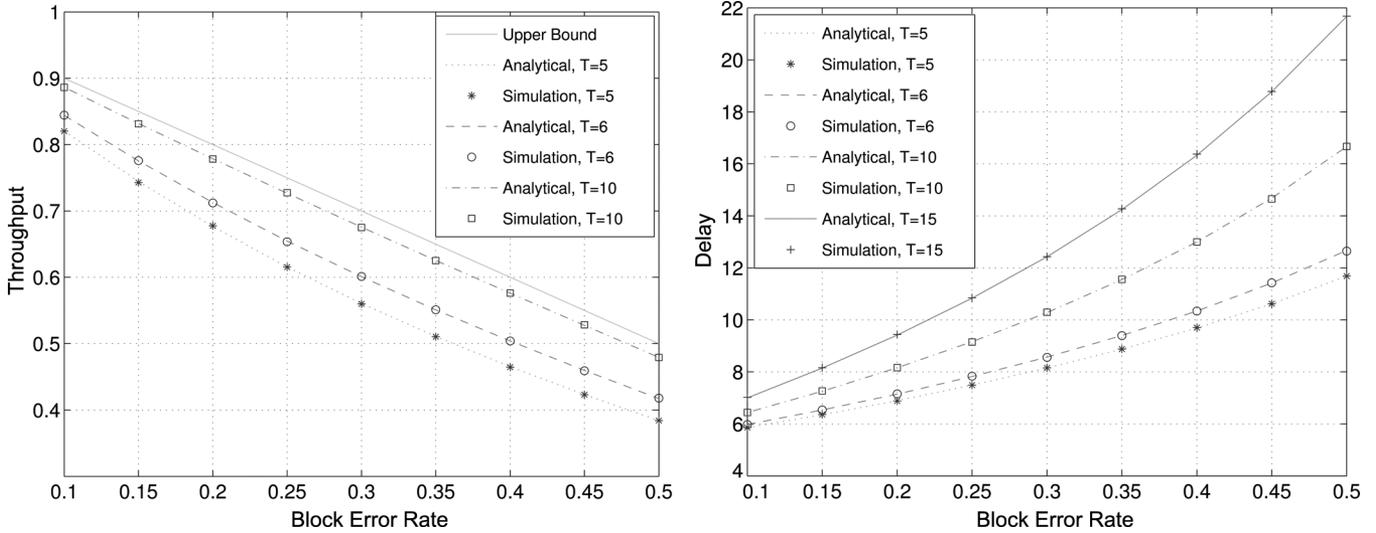


Fig. 6. Throughput η , and delay D , versus block-error rate ϵ , in Markov errors without NACK for $r = 0.3$ and $k = 5$.

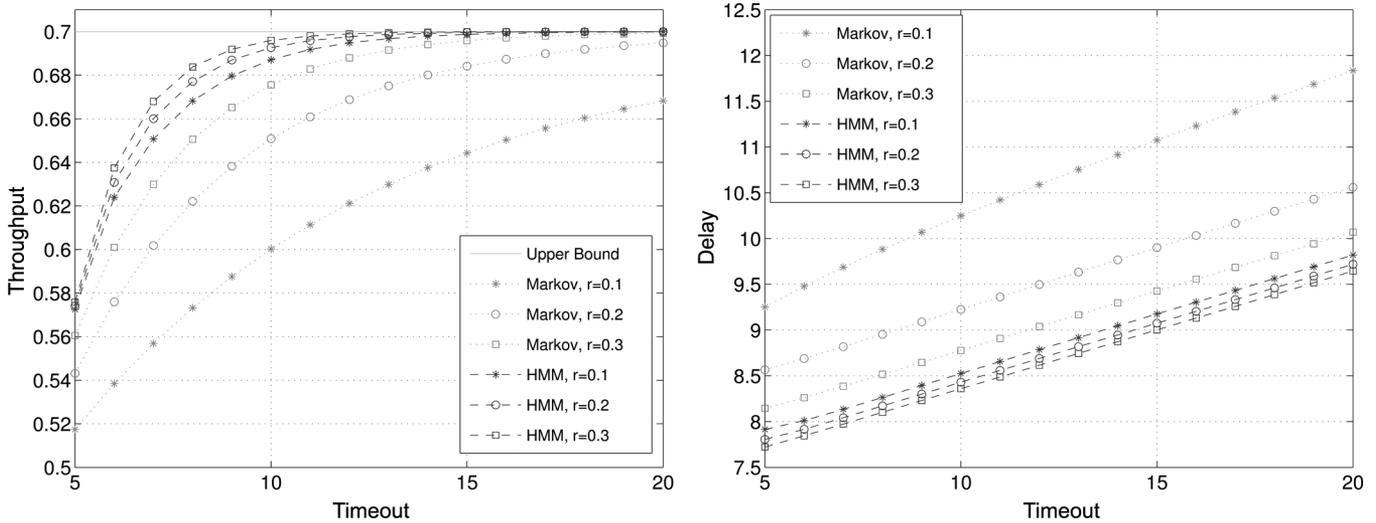


Fig. 7. Throughput η , and delay D , versus timeout T , in Markov errors for $\epsilon = 0.3$ and $k = 5$, and in HMM for $\epsilon = 0.3$, $\epsilon_G = 0.07$, $\epsilon_B = 0.7$, and $k = 5$.

about the receiver status. In this scenario, a frame is retransmitted only because of timer expiration, so two frames cannot experience timeout at the same time (this is the source of the aforementioned discrepancy). We derive the expected throughput and delay in the same manner as before. Fig. 6 shows that now, analytical and simulation results agree perfectly. Note the throughput of SR with and without NACK is almost the same. Similar behavior is also observed in hidden Markov channels, but in the interest of brevity, the details are omitted.

We analyze the effect of timeout on the SR performance. The plots of the throughput and delay versus timeout are shown in Fig. 7. The plots compare the SR performance in both Markov and hidden Markov channels. With noiseless feedback, the error burst will not effect the SR throughput, a fact that has been reported in [1]. We see that the higher the error burst, the lower the SR throughput in noisy feedback. We also notice that the

throughput in HMM is less sensitive to parameter r , compared with the Markov channel. Moreover, the throughput approaches the bound faster in HMM than in the Markov channel. Also note that delay increases almost linearly with timeout. So timeout should be configured such that throughput is maximized, while maintaining acceptable delay.

Fig. 8 compares two SR variations. In method SR_1 , the feedback includes information about all successfully received frames, while the extra information is excluded in method SR_2 . For brevity, we omit the flow graphs and analysis of the individual variations, and only mention that the same technique that was developed earlier in the paper was used for their analysis.

We observe that method SR_1 has higher throughput and lower delay. Also, the throughput of SR_2 is less sensitive to the value of timeout, compared with SR_1 ; however, the delay of SR_2 is more sensitive. Similar results are obtained in hidden Markov channels, but are omitted for the sake of brevity.

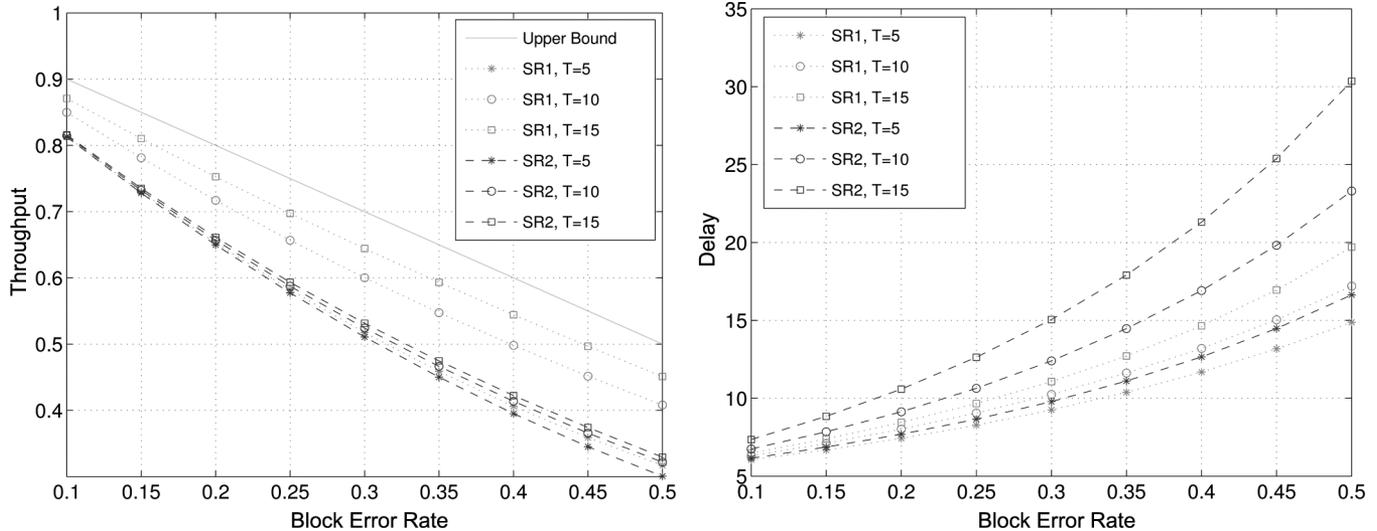


Fig. 8. Throughput η , and delay D , versus block-error rate ϵ , in Markov errors for $r = 0.1$ and $k = 5$.

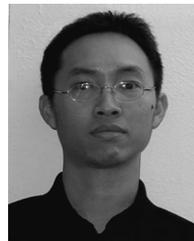
V. CONCLUSION

This paper analyzes the performance of SR ARQ by a generalization of Mason's graphs that allows matrix branch weights. We find the MGF of transmission and delay time, and the average throughput and delay are calculated using the MGF. We apply this technique to the case of noiseless feedback, as well as noisy feedback, and demonstrate the application with examples. This technique can be applied to any finite-state HMM, and might be useful in analysis of other protocols in HMMs.

REFERENCES

- [1] D. L. Lu and J. F. Chang, "Performance of ARQ protocols in nonindependent channel errors," *IEEE Trans. Commun.*, vol. 41, no. 5, pp. 721–730, May 1993.
- [2] S. R. Kim and C. K. Un, "Throughput analysis for two ARQ schemes using combined transition matrix," *IEEE Trans. Commun.*, vol. 40, no. 11, pp. 1679–1683, Nov. 1992.
- [3] Y. J. Cho and C. K. Un, "Performance analysis of ARQ error controls under Markovian block error pattern," *IEEE Trans. Commun.*, vol. 42, no. 2-4, pp. 2051–2061, Feb.–Apr. 1994.
- [4] M. Zorzi and R. R. Rao, "Throughput performance of ARQ selective-repeat protocol with time diversity in Markov channels," in *Proc. IEEE GLOBECOM*, Nov. 1995, vol. 3, pp. 1673–1677.
- [5] —, "Bounds on the throughput performance of ARQ selective-repeat protocol in Markov channels," in *Proc. IEEE GLOBECOM*, Jun. 1996, vol. 2, pp. 782–786.
- [6] W. Turin and R. van Nobelen, "Hidden Markov modeling of flat fading channels," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 9, pp. 1809–1817, Dec. 1998.
- [7] P. Bergamo, D. Maniezzo, A. Giovanardi, G. Mazzini, and M. Zorzi, "An improved Markov chain description for fading processes," in *Proc. IEEE ICC*, Apr.–May 2002, pp. 1347–1351.
- [8] —, "Improved Markov model for Rayleigh fading envelope," *IEE Electron. Lett.*, vol. 38, no. 10, pp. 477–478, May 2002.
- [9] D. L. Lu and J. F. Chang, "Analysis of ARQ protocols via signal flow graphs," *IEEE Trans. Commun.*, vol. 37, no. 3, pp. 245–251, Mar. 1989.
- [10] S. J. Mason, "About such things as unistors, flow graphs, probability, partial factoring and matrices," *IRE Trans. Circuit Theory*, vol. CT-4, pp. 90–97, Sep. 1957.

- [11] W. Turin, "Throughput analysis of the Go-Back-N protocol in fading radio channels," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 5, pp. 881–887, May 1999.
- [12] —, *Performance Analysis and Modeling of Digital Transmission Systems*. New York: Kluwer/Plenum, 2004.
- [13] S. J. Mason and H. J. Zimmermann, *Electronic, Circuits, Signals, and Systems*. New York: Wiley, 1960.
- [14] R. A. Howard, *Dynamic Probabilistic Systems*. New York: Wiley, 1971.



Kamtorn Ausavapattanakun was born in Bangkok, Thailand, in 1975. He received the B.E. degree in electronics and telecommunications from King Mongkut's Institute of Technology, Thonburi, Thailand, in 1996, and the M.S. degree in electrical engineering from the University of Texas at Dallas, Richardson, in 2005.

His thesis work is on ARQ analysis under finite-state hidden Markov models.



Aria Nosratinia (M'97–SM'04) received the B.S. degree in electrical engineering from the University of Tehran, Tehran, Iran, in 1988, the M.S. degree in electrical engineering from the University of Windsor, Windsor, ON, Canada, in 1991, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign in 1996.

From 1995 to 1996, he was with Princeton University, Princeton, NJ. From 1996 to 1999, he was a Visiting Professor and Faculty Fellow at Rice University, Houston, TX. Since 1999, he has been with the University of Texas at Dallas, Richardson, where he is currently an Associate Professor of Electrical Engineering. His research interests are in the broad areas of information theory, coding, and signal processing.

Dr. Nosratinia is currently an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and IEEE WIRELESS COMMUNICATIONS. He was the recipient of the National Science Foundation Career award in 2000, and has twice received chapter awards for outstanding service to the IEEE Signal Processing Society.