A GENERAL ANALYSIS TECHNIQUE FOR ARQ PROTOCOL PERFORMANCE IN HIGH SPEED NETWORKS

Ian F. Akyildiz and Wei Liu

College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332

ABSTRACT

The general behavior of the ARQ protocols is modeled by a queueing network with priorities representing the erroneous and recovery processes. A mean value analysis algorithm is developed for the performance analysis of the model. The input parameters, the erroneous period and the recovery duration, for the model are gained from the extended finite state machine model description of the protocols. By giving several examples we have demonstrated the general applicability of the new approach.

Key Words: ARQ Protocol, High Speed Networks, Queueing Networks, Extended Finite State Machine Model, Performance Measures.

1. Introduction

Packet switching networks have changed considerably in recent years. The advent of fiber optic media has pushed the transmission speed of communication links to over a Gbps. With the high speed fiber optic channel, data and voice integration service will change the nature of the carried traffic. Another influencing factor is the more reliable communication link in terms of bit error rates. These factors have a significant impact on the design of the protocols and control procedures in the packet switched networks. An important issue is the prediction and improvement of the performance of the error control protocols such that the requirements of high speed packet switching networks will be met.

In the last three decades several error recovery and flow control protocols have been proposed and implemented on the data link layer of the OSI model. Several methods have been introduced for the performance analysis of these protocols [7,8,12,17,23,26,27]. We note that these solution techniques for ARQ protocols are too specific to be applied generally. For example, with the high transmission speed (i.e., in very high speed networks), the propagation delay is becoming a more critical factor. Previous solution techniques that do not consider this factor are out of date. Another problem is the separate analysis of each ARQ scheme by different solution techniques. Even a small change of a protocol feature would lead to a new model which cannot be solved by previous techniques. First attempt for a general modeling approach of various ARQ protocols was by Towsley and Wolf [27] as well as by Anagnoston and Protonotarios [2]. Although they use a common approach the mathematical models turn out to be specific for each ARQ scheme. In contrast to their approaches we introduce a more general mathematical model to cover all schemes.

Recently different features of the standard protocols are studied in the literature. For example, Brady [5] simulates the effects of four features of LAPB and LAPD which are data link layer ARQ protocols in ISDN systems. Previous analytical approaches cannot deal with this type of comparisons due to the lack of a uniform framework to model these protocols. A general model is needed which is the abstraction of all ARQ protocols such that trivial details will not complicate the modeling process and the analysis. Moreover, the model should capture the essential behavior of the ARQ protocols to provide practical evaluation. The analytical model we present in this paper can be applied for protocol design and analysis of different features.

Another aspect is the interaction of protocol entities while a packet travels along a path [3,6,10,18,24,25]. Since the behavior of a packet in a node will affect the following node, it is necessary to take into account both the network performance and the timing of the internal behavior of a node. Suda and Watanabe [24] modeled a fast packet switching network as a tandem queueing network model. Each queue in the model represents protocol layer rather than a whole switching node. Bhargava et. al. [3] obtained similar results as [24]. Each switching node along a virtual circuit is modeled by a single queue. Bradley and Suda [6] obtained simulation results for performance of various error control schemes in a fast packet switching network environment. They examined three error control schemes. How-

---

Ian F. Akyildiz was supported by the Universite Pierre et Marie Curie, (Lab MASI) and the Ecole National Superieure Telecommunications (ENST), Paris, France.

ever, these studies do not model the error recovery process inside the switching node explicitly. Our model can be used to study both the internal and external behavior of the error control schemes.

In this paper we introduce a general model for a unified analysis of ARQ protocols. We demonstrate that the performance of the network layer is dependent on the underlying ARQ scheme performance. The paper is organized as follows. In section 2 we describe a queueing network model which captures the error and recovery behavior of the ARQ protocols. In section 3 we derive a mean value analysis algorithm for the efficient computation of performance measures. In section 4 we develop an Extended Finite State Machine Model (EFSM) for determining the input parameters for the model of section 2. In section 5 we give numerical examples for the performance analysis of a single protocol entity, of a virtual circuit, and for the performance comparison of different error control schemes.

2. Model Description

The main purpose of the ARQ protocol is to provide an error recovery mechanism for corrupted packets over an unreliable link. It is complicated by embedding the flow control mechanism into the protocol. Therefore, the entire model considers the interleaving of the packet transmission intervals with the retransmission of the corrupted packets or with the blocking due to the flow control.

A transmission (or switching) node is modeled by a First-Come-First-Service (FCFS) node. Packets are transmitted one by one upon their arrival at a node. From time to time, some errors may occur in which case the retransmission must take place. Sometimes the window for flow control may become full, and the packets have to be blocked until some previous transmitted packets are acknowledged. The objective here is to find a mechanism which will capture these aspects. For each node, we introduce a so-called virtual node and a single virtual packet as shown in Figure 1. The virtual node and the virtual packet are essential to characterize the behavior of the ARQ protocol, as explained later.

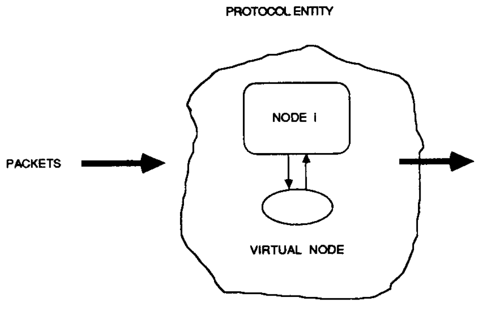

Figure 1. Model of the ARQ Protocol Entity

Each virtual node is connected to its corresponding transmission node. The actual packets and the virtual packets belong to two different classes. The virtual packet has a higher priority than the actual packet. It only circulates between the transmission node and the virtual node as shown in Figure 1.

While the virtual packet is at the virtual node, the packets at the transmission node are transmitted one after another, as in a FCFS station. When the window is full or when a NAK is received by the source node, the transmission of the subsequent packets is stopped until the window is slided or the retransmission is finished. The blocking of regular transmission of packets is modeled by letting the virtual packet to enter the transmission node from the virtual node. Since the virtual packet has higher priority than the actual packets, the node will serve the virtual packet first, thus blocking the transmission of the actual packets. When the service of the virtual packet is finished, the virtual packet goes back to the virtual node. This corresponds to the unblocking of the transmission node such as forwarding the window. The packets are started to be transmitted again.

The service time of the actual packets is the transmission time. The service time of the virtual packet at the actual node is the delay for the ACK or NAK, which is defined by s. The service time s' of the virtual packet at the virtual node depends on the window size and the error probability at the communication link. The larger the window size, the longer is the service time. The smaller the error probability, the longer is the actual packet transmission time, thus making the virtual packet's service time longer. The computation of the parameters s and s' are discussed in section 4.

3. Model Analysis

In section 3.1 we give the analysis for a single transmission node. The transmission node and the corresponding virtual node constitute an open queueing network. In section 3.2 we analyze virtual circuit networks which are modeled as closed queueing networks. For both cases we derive analytical solutions for performance measures in a closed form.

3.1. Performance for a Single Transmission Node

The model for a single transmission node in Figure 1 is a queueing network with two nodes. From reduced work rate (RWR) approximation technique [11] we obtain the mean delay time $\bar{t}$ for the packet transmission as

$$\bar{t} = \frac{\hat{t}}{1 - \rho_v} \qquad (3\text{-}1)$$

where

$\hat{t} = \frac{1/\mu}{1-\rho}$ is the mean delay time for a classical single FCFS station,

ρ_v is the utilization of the virtual packet at the transmission node,

μ is the transmission rate of the transmission node,

$\rho = \frac{\lambda}{\mu}$ is the utilization of the transmission node for a classical single FCFS station,

λ is the packet arrival rate.

Note that the virtual packet circulates between the transmission node and the virtual node with the following busy and idle period.

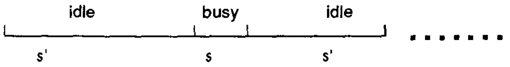

Figure 2. Busy and Idle Period

where s is the service time of the virtual packet at the transmission node and s' is the service time of the virtual packet at the virtual node. From Figure 2 we obtain

$$\rho_v = \frac{s}{s + s'} \qquad (3\text{-}2)$$

By substituting (3-2) into (3-1) we determine the mean delay time of a packet in a transmission node as

$$\bar{t} = \frac{s + s'}{\mu s'(1-\rho)} \qquad (3\text{-}3)$$

From Little's formula we obtain the mean packet length in a transmission node as

$$\bar{k} = \lambda\bar{t} = \frac{\rho(s+s')}{(1-\rho)s'} \qquad (3\text{-}4)$$

where λ is the packet arrival rate at the transmission node. It contains both the packets passing this node and the packets generated by this node.

3.2. Performance for Virtual Circuit Networks

Here we consider the effect of the protocols on the performance of the total network. We assume that there are N nodes in the network. For convenience, the virtual node corresponding to the transmission node i is labelled as $(i+N)$. Without loss of generality, the virtual packet at node i and $i+N$ is labelled as packet class i, and the actual packets are labelled as packet class $N+1$. Let the packet class i has higher priority than the packet class j, for $i<j$ and $i,j=1,2,\ldots,N,N+1$. It is obvious that the above model becomes a multiclass queueing network model with priorities.

Without loss of generality, we consider that there are arbitrary but finite number of packets. Thus, the network becomes a closed model. The solution for this queueing network model is based on the CTA algorithm [9] which is a mean value analysis algorithm for multiclass priority queueing networks. The *mean delay time* of packets in the i-th node (for $i = 1,2,\ldots,N$) is:

$$\bar{t}_{i,r} = \sum_{l=1}^{r-1} \frac{\frac{\bar{k}_{i,l}}{\mu_{i,l}}}{1-\sum_{j=1}^{l-1}\rho_{i,j}} + \frac{\frac{K_r - 1}{K_r}\,\frac{\bar{k}_{i,r}}{\mu_{i,r}}}{1-\sum_{j=1}^{r-1}\rho_{i,j}} + \frac{\frac{1}{\mu_{i,r}}}{1-\sum_{j=1}^{r-1}\rho_{i,j}} \qquad (3\text{-}5)$$

where $\rho_{i,j}$ is the utilization of node i by packets with class r.

The right hand side of this formula has the following intuitive interpretation. Part 1 is the service time for the packets with higher priority than the tagged arriving packet with class r. Part 2 is the the service time for the earlier arrived packets of the same class, adjusted by the Schweitzer/Bard factor $(K_r - 1)/K_r$. Part 3 represents the service time for the tagged packet itself. The denominator increases the effective service rate and thus slows down the service times of the servers because of the pre-emption.

We are interested in the delay time for the packets. Note that certain packets never go to certain nodes. Thus only the following delay times are interesting: $\bar{t}_{i,i}$, $\bar{t}_{N+i,i}$, and $\bar{t}_{i,N+1}$ (for $i = 1,\ldots,N$). We derive formulas of these delay times by analyzing the behavior of the virtual packets. There is at most one virtual packet in node i and $N+i$, and it always has higher priority to pre-empt other packets from service, i.e., the virtual packet gets service immediately after it enters a node. Thus,

$$\bar{t}_{N+i,i} = s'_i \qquad (3\text{-}6)$$

$$\bar{t}_{i,i} = s_i \qquad (3\text{-}7)$$

for all i.

The mean delay time for the packets can be obtained by simplifying (3.5) and considering that if $e_{ir} = 0$ then $\bar{k}_i = 0$ for all $i = 1,\ldots,N$ and $r = 1,\ldots,R$ except $i = r$:

$$\bar{t}_{i,N+1} = \frac{\frac{\bar{k}_{i,i}}{\mu_{i,i}}}{1-\rho_{i,i}} + \frac{\frac{K_{N+1} - 1}{K_{N+1}}\,\frac{\bar{k}_{i,N+1}}{\mu_{i,N+1}}}{1-\rho_{i,i}} + \frac{\frac{1}{\mu_{i,N+1}}}{1-\rho_{i,i}} \qquad (3\text{-}8)$$

where e_i is the visit ratio derived from

$$e_i = \sum_{j=1}^{N} e_j p_{j,i}$$

where $p_{j,i}$ is the probability that an actual packet goes from node j to node i.

To further simplify (3-8), we observe the behavior of the arrival rate $\lambda_{i,i}$ and utilization $\rho_{i,i}$ of the virtual packet in node i. Similar to the analysis of the idle/busy period for the virtual packet in Figure 2 we obtain

$$\rho_{i,i} = \frac{s_i}{s_i + s'_i} \qquad (3\text{-}9)$$

which provides

$$\lambda_{i,i} = \rho_{i,i}\,\mu_{i,i} = \frac{1}{s_i + s'_i} \qquad (3\text{-}10)$$

By substituting (3-7), (3-9) and (3-10) into (3-8) we obtain

$$\bar{t}_{i,N+1} = \frac{{s_i}^2}{s'_i} + \frac{\frac{K_{N+1}-1}{K_{N+1}}\,\frac{\bar{k}_{i,N+1}}{\mu_{i,N+1}}}{\frac{s'_i}{s_i + s'_i}} + \frac{\frac{1}{\mu_{i,N+1}}}{\frac{s'_i}{s_i + s'_i}} \qquad (3\text{-}11)$$

Note that only packets of class $N+1$ occur in (3-11), thus we can omit class subscript.

$$\overline{t_i} = \frac{1}{\mu_i \; B_i} [\, 1 + \frac{K-1}{K} \overline{k_i} \,] + A_i \qquad (3\text{-}12)$$

where

$$A_i := \frac{s_i^2}{s'_i} \qquad (3\text{-}13)$$

$$B_i := \frac{s'_i}{s_i + s'_i} \qquad (3\text{-}14)$$

From Little's law we obtain the mean number of actual packets in node

$$\overline{k_i} = \lambda \, e_i \, \overline{t_i} \qquad (3\text{-}15)$$

The network throughput is

$$\lambda = \frac{K}{\sum_{i=1}^{N} e_i \, \overline{t_i}} \qquad (3\text{-}16)$$

Equations (3-12),(3-15) and (3-16) form the new mean value analysis formulas [1] for analyzing the performance of virtual circuit networks with intermediate nodes transmitting the packets under ARQ protocols. To use these formulas iteratively we assume the initial values for $\overline{k_i}$ as:

$$\overline{k_i} = \frac{K}{N} \qquad \text{for} \qquad i = 1,\ldots,N. \qquad (3\text{-}17)$$

4. Extended Finite State Machine Model

We need to compute the service time parameters s and s' which occur in equations (3-3),(3-4),(3-13) and (3-14). These parameters are dependent on the specific protocols modeled. For example, the different window size is reflected by the different virtual service time of the virtual packet at the virtual node. The larger the window size, the longer is the virtual packet duration in the virtual node. Another example is the different ratio of the propagation delay to the transmission time. This is modeled by varying the service time at the transmission node. The longer the propagation delay of the ACK packet, the longer is the packet transmission blocking time. This implies a longer service time for the virtual packet at the transmission node. Higher error rates will cause the virtual packet to enter the transmission node more frequently.

There are two kinds of ARQ protocol parameters that are essential for the modeling and analysis of the protocols. One is the time interval s' when the transmitter and receiver are in normal transmitting and receiving status. The other is the time interval s when the nodes have to take care of the error recovery or flow control.

Complete definitions of s and s' are given as follows:

s is the time interval either

i) from transmitting the last packet which makes the window size full until receiving of the first ACK which advances the full window or

ii) from receiving a NAK packet until finishing the necessary retransmission.

s' is the time interval either

i) from receiving the first ACK packet which advances the full window until the transmission of the last data packet which makes the window full again or

ii) from receiving the first ACK packet which advances the full window until receiving of a NAK packet or

iii) from finishing retransmission after a NAK packet until the transmission of the last data packet which makes the window full again or

iv) from finishing retransmission of data packets until receiving of another NAK packet.

Our study reveals that these parameters characterize the internal operational behavior of ARQ protocols. In contrast, previous studies of the protocol performance usually ignore these parameters. In fact, these parameters are not only important to determine the internal operation of the protocol, but also critical for the overall performance of all protocols.

The formal specification of protocols can be described by a Finite State Machine model which can be extended with timing and branching probability labels. If we assume the transition time to be exponentially distributed, then the Extended Finite State Machine model (EFSM) [22] becomes a homogeneous Markov chain. Kritzinger [13, 14] extended this idea to model the performance of OSI communication architecture. By categorizing the transitions into different service classes he transfers the EFSM into a queueing network model with jobs constantly changing their classes. While his main concern is to model multiple protocol entities within a single node, our objective is to model the interactions of multiple packets with the ARQ protocol entities along a virtual circuit.

In the EFSM the time label indicates the expected duration of the transition between states. These timings can be derived from counting the number of instruction codes associated with each transition out of a state. Sometimes it is possible to have a meta-implementation to trace the execution of the formal protocol description [19]. Then the probability of a transition can be computed by counting the number of transitions out of every state.

The following definitions and notations are used in what follows:

K : the total number of packets transmitted;

S_τ : the expected duration of transition τ;

S : the average time of transmitting a packet;

$P_{\tau',\tau}$: the branching probability that transition τ will follow transition τ';

ξ_τ : the relative visit ratio of transition τ. (Transition τ has the following types: *(i)_Trans*, *Retrans*, *ACK*, *(i,j)_NAK*, *Timeout*. They are denoted as transition τ when detailed references are unnecessary);

(i)_Trans : the transition that transfers a packet from one state to another and the EFSM is in a state with i out-

standing packets;

Retrans : the transition caused by the corrupted packets retransmission;

W : the protocol window size;

$(W)\_timeout$: the timeout transition after the window size W is full;

$(i,j)\_NAK$: the transition caused by receiving a NAK packet requiring j packets to be retransmitted and the EFSM is in a state of having i outstanding packets. (Note that in the case of Go-Back-N protocol, j can take the value from 1 to i because one NAK packet will cause all the previous packets to be retransmitted. In Selective Repeat Request Protocol, j is always 1 because only the corrupted packet needs to be retransmitted).

Since we consider the EFSM as a Markov chain, the following flow balance equations are satisfied:

$$\xi_\tau = \sum_{\tau'} \xi_{\tau'}\, p_{\tau',\tau} \tag{4-1}$$

By solving (4.1) we obtain the relative visit ratio of each transition ξ_τ.

We further enforce that

$$\sum_{i=0}^{W-1} \xi_{(i)\_Trans} = K \tag{4-2}$$

i.e., the transitions of the transmitting packets are executed K times. From equations (4-1) and (4-2) we can obtain a unique solution. Moreover we can interpret the solutions as the number of times each transition τ is executed under the condition that all K packets are transmitted successfully exactly once. The total time taken to transmit K packets by the protocol given the condition that the packets are already transmitted is

$$T = \sum_\tau \xi_\tau\, S_\tau \tag{4-3}$$

for all transitions τ in the EFSM specification.

The total blocking time due to transition $(i,j)\_NAK$ is

$$T[blocking\ (i,j)\_NAK] = \xi_{(i,j)\_NAK}\,(S_{(i,j)\_NAK} + j\ S) \tag{4-4}$$

since $\xi_{(i,j)\_NAK}$ is the number of times that a NAK packet arrives. For each of the NAK packet arrival, j packets are retransmitted in time $j \cdot S$.

The total blocking time due to timeout is

$$T[blocking\ window\ full] = \xi_{timeout}\,(S_{timeout} + W\ S) \tag{4-5}$$

Note that W packets are retransmitted.

Since $\left(\sum_{(i,j)\_NAK} \xi_{(i,j)\_NAK} + \xi_{timeout} \right)$ is the total number of times of blocking, the probability that a particular type of blocking occurs is

$$Prob[(i,j)\_NAK] = \frac{\xi_{(i,j)\_NAK}}{\sum_{(i,j)\_NAK} \xi_{(i,j)\_NAK} + \xi_{timeout}} \tag{4-6}$$

$$Prob[window\ full] = \frac{\xi_{timeout}}{\sum_{(i,j)\_NAK} \xi_{(i,j)\_NAK} + \xi_{timeout}} \tag{4-7}$$

Thus the mean blocking time of the transmission node (i.e., the mean service time of the virtual packet in the transmission node in the models of section 2) is

$$s = \sum_{(i,j)\_NAK} Prob[(i,j)\_NAK] \cdot (S_{(i,j)\_NAK} + j\ S) + Prob[window\ full] \cdot (S_{timeout} + W\ S) \tag{4-8}$$

which is rewritten as

$$s = \frac{\sum_{(i,j)\_NAK} \xi_{(i,j)\_NAK}(S_{(i,j)\_NAK} + j\ S) + \xi_{timeout}(S_{timeout} + W\ S)}{\sum_{(i,j)\_NAK} \xi_{(i,j)\_NAK} + \xi_{timeout}} \tag{4-9}$$

To obtain the mean value of s', i.e., the mean service time of the virtual packet in the virtual node, we consider the following scenario. The virtual packet enters and leaves the virtual node in an interleaving way. Thus, this mean value of normal transmission period is

$$s' = \frac{T - \sum_{(i,j)\_NAK} T[blocking(i,j)\_NAK] - T[blocking\ window\ full]}{\sum_{(i,j)\_NAK} \xi_{(i,j)\_NAK} + \xi_{timeout}} \tag{4-10}$$

In rewritten form,

$$s' = \frac{\sum_\tau \xi_\tau S_\tau}{\sum_{(i,j)\_NAK} \xi_{(i,j)\_NAK} + \xi_{timeout}} - \frac{\sum_{(i,j)\_NAK} \xi_{(i,j)\_NAK}\,(S_{(i,j)\_NAK} + j\ S) + \xi_{timeout}\,(S_{timeout} + W\ S)}{\sum_{(i,j)\_NAK} \xi_{(i,j)\_NAK} + \xi_{timeout}} \tag{4-11}$$

Even though both expressions (4-9) and (4-11) are derived in the context of closed queueing networks, the following theorem shows that they are also applicable to open queueing networks.

Theorem 1. *Formulas given in (4-9) and (4-11) are valid for the nodes of both open and closed queueing networks.*

Proof. In a closed queueing network, the number of packets for transmission K is fixed while in open queueing networks, K may vary. We only need to show that the expressions for s and s' are independent of K. They can be applied for the network analysis regardless of the type of the queueing network models.

Assume that ξ'_τ is an arbitrary solution for (4-1) where τ belongs to the set of following transitions with $\tau \in \{(i,j)\_Retrans, timeout\}$. Then $\xi'_\tau = C\ \xi_\tau$ (with C as a constant), because both ξ_τ and ξ'_τ satisfy the same linear equation systems. Therefore,

$$\frac{\sum_\tau \xi'_\tau\ T[blocking\ \tau]}{\sum_\tau \xi'_\tau} = \frac{C\left[\sum_\tau \xi_\tau\ T[blocking\ \tau]\right]}{C\left[\sum_\tau \xi_\tau\right]} = s$$

Analogously this holds for s'. □

5. Examples

To compute the performance of ARQ protocols in a networking environment, the procedure is

i) Developing the EFSM specification of the protocol;

ii) Computing the protocol parameters s and s' from equations (4-9) and (4-11);

iii) Solving the priority queueing network model and obtaining performance measures by the mean value analysis equations (3-14), (3-15) and (3-16).

In the following, we give three examples to demonstrate the application of the models. The first example provides the performance analysis of the Stop-and-Wait, Selective Repeat Request and the Go-Back-N protocols. The second example analyzes the performance of a virtual circuit. In particular, we compare hop-by-hop and end-to-end error control schemes. The third example is a high speed network environment where we discuss the shortcomings of the error control schemes.

5.1. ARQ Protocols

This section contains the three most common ARQ protocols: Stop-and-Wait, Selective Repeat Request and Go-Back-N. We will describe how to derive the parameters s and s' from (4-9) and (4-11) for each protocol.

5.1.1. Stop and Wait Protocol

Each time a transmission node sends a packet, it waits for an ACK packet. If a NAK packet arrives during the waiting or nothing arrives until timeout, it retransmits the same packet again. When an ACK packet arrives, the transmission node is ready to transmit another packet upon its arrival.

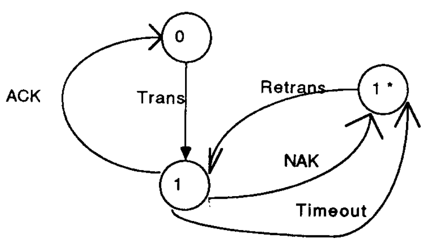

Figure 3. EFSM Modeling of a Stop-and-Wait Protocol Entity

In Figure 3, state (0) is the idle state in which the node can transmit packets upon arrival. When a packet is transmitted, then state (1) is reached. When entering state (1), if the transmission is successful (i.e., an ACK packet arrives), it goes to state (0) again. Otherwise, it goes to state (1*), which means that error occurs and retransmission is required due to a NAK packet or to timeout.

We assume the mean duration times for transitions as:

S_{ACK}=0.5	S_{Trans}=1	$S_{Retrans}$=1	S_{NAK}=0.5	$S_{Timeout}$=2

Table 1. Mean Transition Duration Times

The transition probabilities are assumed to be

Prob[ACK->Trans]=1	Prob[Trans->ACK]=0.998	Prob[Trans->NAK]=0.001
Prob[Trans->Timeout]=0.001	Prob[Retrans->ACK]=0.998	Prob[Retrans->NAK]=0.0001
Prob[Retrans->Timeout]=0.0001	Prob[Timeout->Retrans]=1	

Table 2. Transition Probabilities

In Table 2 the error probability is assumed to be very small. Therefore, the probability (0.0001) that a *NAK* packet arrives or *timeout* occurs, is a reasonable assumption. In some case the branching probability is fixed. For example, if an *ACK* packet arrives, the next only possible transition is *Trans*, thus $Prob[ACK \rightarrow Trans]$ is 1.

From the EFSM model, equation (4-9), we derive

$$s = \frac{\xi_{NAK}(S_{NAK} + S_{Retrans}) + \xi_{Retrans}\, S_{Retrans} + \xi_{Timeout}(S_{Timeout} + S_{Retrans})}{\xi_{NAK} + \xi_{Timeout}} \quad (5\text{-}1)$$

where ξ_{NAK}, $\xi_{Timeout}$, $\xi_{Retrans}$ are computed from equation (4-1).

From equation (4-11) we determine

$$s' = \frac{\xi_{Trans} S_{Trans} + \xi_{ACK} S_{ACK}}{\xi_{NAK} + \xi_{Timeout}} \quad (5\text{-}2)$$

where ξ_{Trans} and ξ_{ACK} are calculated from equation (4-1).

5.1.2. Selective Repeat Request Protocol

Selective Repeat Request protocol allows a number of packets to be transmitted without waiting for the ACK packets. If an error occurs, only the corrupted packet needs to be retransmitted. Thus, the efficiency will be improved.

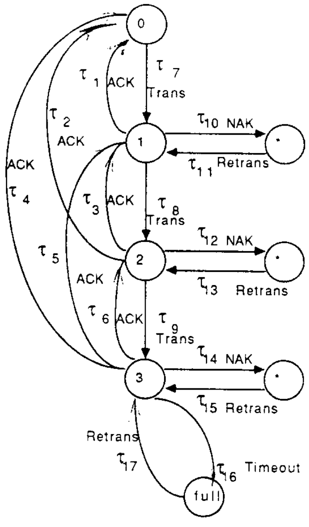

Figure 4. EFSM of a Selective Repeat Request Protocol Entity

The state with (*) indicates that an error occurred and retransmission is required. States k, for $k = 0,1,2,3$, denote that there are k packets waiting for ACK packets. State ($full$) arises from the timeout transition and needs retransmission of the whole window. Other transitions are obvious as labelled in Figure 4.

The following parameters are assumed for the model in Figure 4. Assume that the mean transmission time is 1, the window size is 3. The mean ACK/NAK packet transmission period is 0.5. The timeout interval is 4. Also assume the following transition probabilities where $p[i,j] = Prob[\tau_i \rightarrow \tau_j]$. Other transition probabilities not explicitly given are assumed to be zero.

p[1,7]=1				
p[2,7]=1				
p[3,1]=0.499	p[3,8]=0.499	p[3,10]=0.002		
p[4,7]=1				
p[5,1]=0.400	p[5,8]=0.598	p[5,10]=0.002		
p[6,2]=0.200	p[6,3]=0.200	p[6,9]=0.598	p[6,12]=0.002	
p[7,1]=0.400	p[7,8]=0.598	p[7,10]=0.002		
p[8,2]=0.200	p[8,3]=0.200	p[8,9]=0.598	p[8,12]=0.002	
p[9,4]=0.497	p[9,5]=0.300	p[9,6]=0.200	p[9,14]=0.002	p[9,16]=0.001
p[10,11]=1				
p[11,1]=0.400	p[11,8]=0.600	p[11,10]=0		
p[12,13]=1				
p[13,2]=0.200	p[13,3]=0.200	p[13,9]=0.600	p[13,12]=0	
p[14,15]=1				
p[15,4]=0.499	p[15,5]=0.300	p[15,6]=0.200	p[15,14]=0	p[15,16]=0.001
p[16,17]=1				
p[17,4]=0.498	p[17,5]=0.300	p[17,6]=0.200	p[17,14]=0.002	p[17,16]=0

Table 3. Transition Probabilities

In Table 3 we assume that when more packets are transmitted, the probability of executing an ACK transition in the EFSM is increasing.

From the above parameters for the EFSM model, equation (4-9), we derive

$$s = \frac{\sum_{i=1}^{W} \xi_{(i,1)\_NAK} \left(S_{(i,1)\_NAK} + S\right) + \xi_{timeout}\left(S_{timeout} + W\ S\right)}{\sum_{i=1}^{W} \xi_{(i,1)\_NAK} + \xi_{timeout}} \qquad (5\text{-}3)$$

where ξ_{τ_i} are obtained from equation (4-1).

From equation (4-11) we get

$$s' = \frac{\sum_{i=0}^{W-1} \xi_{(i)\_Trans} + \sum_{ACK} \xi_{ACK} S_{ACK}}{\sum_{i=1}^{W} \xi_{(i,1)\_NAK} + \xi_{timeout}} \qquad (5\text{-}4)$$

5.1.3. Go-Back-N Protocol

Go-Back-N protocol is similar to Selective Repeat Request protocol except that each time an ACK occurs, the transmission node has to retransmit all the previous unacknowledged packets up to the corrupted one. This reduces the buffer requirements but degrades the efficiency because of the redundant retransmission.

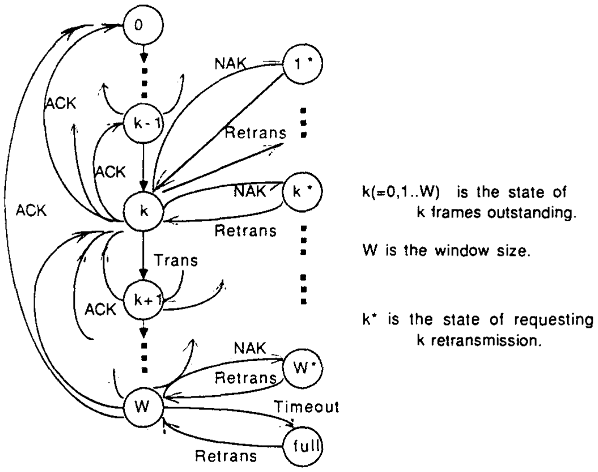

Figure 5. EFSM Modeling of a Go-Back-N Protocol Entity

The state k denotes that there are k packets waiting for an ACK. State (k^*) means that there are (k^*) packets to be transmitted. The transition $(k,k^*)\_NAK$ goes from state (k) to (k^*). The window size is denoted by W. $full$ is the state indicating that the window is full, thus causing blocking.

For a protocol of window size ($W = 7$), such as the *HDLC* in the data link layer of the OSI model, there are total of 93 transitions. For simplifying our discussion we use an example of $W = 3$ although the following process can be applied to different window size in a straightforward way. The following parameters are assumed for this model.

Trans#	From State	Trans_Type	(k)	Branching_Prob.	Trans_Type	Into State
τ_1	1	ACK	0	1	Trans.	1
τ_2	2	ACK	0			
τ_3	3	ACK	0			

Table 4. The Transitions into and out of State (0)

Trans#	From State	Trans_Type	(k)	Branching_Prob.	Trans_Type	Into State
τ_4	0	Trans.	1	0.998	Trans.	2
τ_5	2	ACK	1	0.001	ACK	0
τ_6	3	ACK	1	0.001	NAK	1*
τ_7	1*	Retrans.	1			

Table 5. The Transitions into and out of State (1)

Trans#	From State	Trans_Type	(k)	Branching_Prob	Trans_Type	Into State
τ_8	1	Trans.	2	0.988	Trans.	3
τ_9	3	ACK	2	0.005	ACK	1
τ_{10}	1*	Retrans.	2	0.005	ACK	0
τ_{11}	2*	Retrans.	2	0.001	NAK	2*
			2	0.001	NAK	1*

Table 6. The Transitions into and out of State (2)

Trans#	From State	Trans_Type	(k)	Branching_Prob	Trans_Type	Into State
τ_{12}	2	Trans.	3	0.330	ACK	2
τ_{13}	1*	Retrans.	3	0.330	ACK	1
τ_{14}	2*	Retrans.	3	0.330	ACK	0
τ_{15}	3*	Retrans.	3	0.007	timeout	full
τ_{16}	full	Retrans.	3	0.001	NAK	1*
			3	0.001	NAK	2*
			3	0.001	NAK	3*

Table 7. The Transitions into and out of State (3)

Trans#	From State	Trans_Type	j^*	Branching_Prob	Trans_Type	Into State
τ_{17}	1	NAK	1*	1	Retrans.	1
τ_{18}	2	NAK	1*	1	Retrans.	2
τ_{19}	3	NAK	1*	1	Retrans.	3
τ_{20}	2	NAK	2*	1	Retrans.	2
τ_{21}	3	NAK	2*	1	Retrans.	3
τ_{22}	3	NAK	3*	1	Retrans.	3
τ_{23}	7	timeout	full	1	Retrans.	7

Table 8. The Transitions into and out of State (k^*)

These tables indicate the transitions going into a particular state in the EFSM and the transitions going out of that state. The branching probabilities are those from a particular state to the outgoing transitions. The branching probabilities for the transitions going into that state are the same as the transition probabilities out of that state.

From equation (4-9) we obtain

$$s = \frac{\sum_{k,k^*} \xi_{(k,k^*)\_NAK} \; S_{(k,k^*)\_NAK} + \sum_{k^*} \xi_{(k^*)\_Retrans} k^* \; S_{Retrans}}{\sum_{k,k^*} \xi_{(k,k^*)\_NAK} + \xi_{Timeout}} \quad (5\text{-}5)$$

$$+ \frac{\xi_{Timeout}(S_{timeout} + W \; S_{Retrans})}{\sum_{k,k^*} \xi_{(k,k^*)\_NAK} + \xi_{Timeout}}$$

From equation (4-11) we obtain

$$s' = \frac{\sum_{k} (\xi_{(k)\_Trans} S_{Trans} + \xi_{(k,k')\_ACK} S_{ACK}}{\sum_{k,k^*} \xi_{(k,k^*)\_NAK} + \xi_{Timeout}} \quad (5\text{-}6)$$

RESULTS.

From the EFSM model the parameters for s and s', i.e., equations (5-1) to (5-6) and (3-3), and by assuming the parameters for the interarrival time as 60 ms, average packet length as 1 Kbyte and the propagation delay as 47 ms [16,17] we obtain the results given in Figure 6 where the effect of the transmission speed on the delay of the packets is shown.

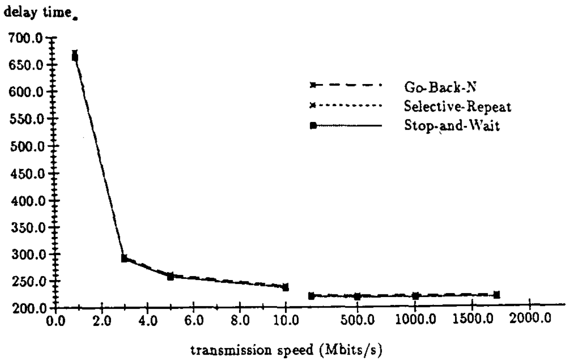

Figure 6. ARQ Performance under Different Transmission Speed

Figure 6 shows the effect of the transmission speed on the packet delay time. Obviously, for low transmission speed, the transmission time is the dominating factor affecting the packet delay. Thus, increasing the transmission speed reduces the transmission time. For high transmission speed, the packet delay time is affected by the propagation time. Thus, increasing transmission speed does not decrease the packet delay time as it should. In Figure 6 the Go-Back-N Protocol has slightly longer packet delay time because of the redundant retransmission.

5.2. Hop-by-Hop versus End-to-End Error Control

We consider a virtual circuit of five nodes. The nodes transmit packets using Selective-Repeat Request protocol to cope with the imperfect transmission link between the nodes. The number of packets transmitted is assumed to be 20. The window size is assumed to be 7. Other parameters are assumed to be the same as in Example 5.1.

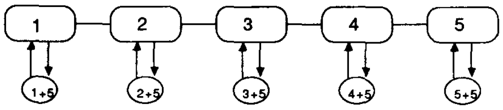

Figure 7. A Virtual Circuit with Hop-by-Hop Error Control.

The End-to-End scheme can be modeled as shown in Figure 8. It is noted that the intermediate node does not contain the virtual server and the virtual packet because the error detection and recovery procedures are eliminated on the node level.

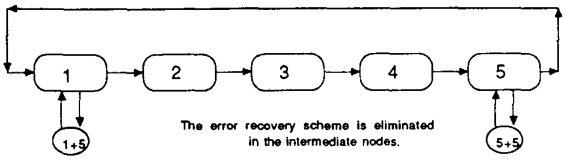

Figure 8. End-to-End Error Control and Recovery.

From equations (3-14), (3-15) and (3-16) we obtain the following results.

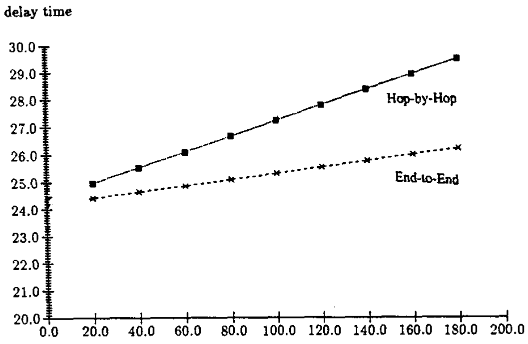

Figure 9. Hop-by-Hop and End-to-End Error Control Schemes

As shown in Figure 9 the end-to-end control scheme provides shorter packet delay time in a high speed network because the error control is eliminated in the intermediate nodes.

5.3. High Speed Network Environment

First we consider a network with fixed transmission speed of 50 Mbps, then we vary the transmission speed up to 1.7 Gbps. The virtual circuit is modeled as shown in Figure 7. Figure 10 shows the relationship between the throughput and the packet delay time.

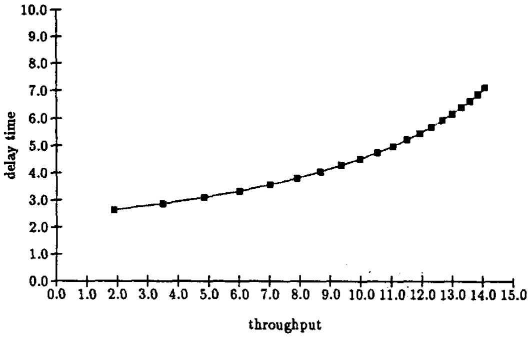

Figure 10. Throughput and Packet Delay Time

In Figure 10 the packet delay time is increasing exponentially with increasing throughput. This suggests that we need to limit the network throughput in order to maintain the packet delay time in a reasonable range.

Figure 11 shows the error and flow control mechanisms cause retransmission which increases the virtual circuit traffic. Since the increasing traffic only depends on the protocol parameters s and s' and these parameters are independent of the load we see that the traffic loads are approximately linear.

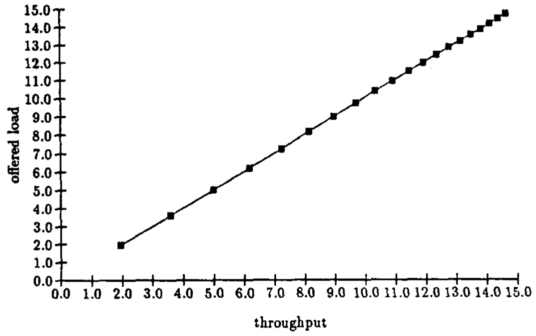

Figure 11. Throughput with Load

6. Conclusions

A queueing network model is presented to characterize the timing and blocking behavior of the ARQ protocols. An algorithm is developed for the mean value analysis of the queueing network model. In order to obtain the input parameters for the queueing network an Extended Finite State Machine Model is developed which captures the behavior of the ARQ protocols. By giving several examples we have demonstrated the general applicability of the model.

The contribution of this paper is threefold:

i) We characterize the general behavior of the ARQ protocols by a queueing network model with priorities representing the erroneous and recovery processes.

ii) We provide efficient mean value analysis algorithm for the performance analysis of the ARQ protocols in a high speed network environment.

iii) We propose a method to determine the input parameters for the general model. The parameters are the erroneous period and the recovery duration. They are obtained from the Extended Finite State Machine Model description of the protocols.

An open problem is to remove the assumption of exponentially distributed transition time. Instead we will consider the deterministic timeout period in the future work.

Further research will be to extend the model to the higher layer error control and recovery processes of the OSI architecture. Another possible future work is the modeling of the multiple connections where within each connection path a number of packets are transmitted by processes governed by the ARQ protocols. Other application includes the computation of the packet delay time in a network governed by certain protocols. The delay information makes it possible to design the optimal timeout interval for a connection setup request.

References

1. I.F. Akyildiz and W. Liu, "Performance Optimization of Distributed System Models with Unreliable Servers", *IEEE Transactions on Reliability, June 1990,* Vol. 39, No. 2, pp. 236-244.
2. M. Anagnoston and E.N. Protonotarios, "Performance Analysis of the Selective Repeat ARQ Protocol", *IEEE Trans. on Communications,* Vol. COM-34, No 2, Feb. 1986, pp. 127-135.
3. A. Bhargava, J.F. Kurose, D. Towsley and G. Van Leemput, "Performance Comparison of Error Control Schemes in High Speed Computer Communication Networks", *IEEE Transactions on Selected Areas on Communications,* Vol. 6, No. 9, Dec. 1988, pp. 1565-1575.
4. N. D. Birrel, "Pre-emptive Retransmission for Communication over Noisy Channels", *IEEE Proc.,* November 1981, pp. 393-400.
5. P.T. Brady, "Performance Evaluation of Multi-Reject, Selective Reject and Other Protocol Enhancements", *IEEE Trans. on Communications,* Vol. COM-35, No 6, June 1987, pp. 659-666.
6. T. T. Bradley and T. Suda, "Performance of Error Recovery Schemes in a Fast Packet Switching Network: Simulation Study", *IEEE ICC Conference,* Atlanta, April 1990, pp. 483-488.
7. H. Bruneel and M. Moeneclaey, "On the Throughput Performance of Some Continuous ARQ Strategies with Repeated Transmissions", *IEEE Trans. on Communications,* March 1986, pp. 244-249.
8. J.K. Choi and C.K. Un, "On Acknowledgement Schemes of Sliding Window Flow Control", *IEEE Trans. on Communications,* Vol. COM-37, No 11, Nov. 1989, pp. 1184-1191.
9. J. van Doremalen, J. Wessels, and R. Wijbrands, "Approximate Analysis of Priority Queueing Networks", *Teletraffic Analysis and Computer Performance Evaluation,* Elsevier Science Publishers B.V. (North-Holland), 1986, pp. 117-131.
10. S.W. Edge and A.J. Hinchley, "A Survey of End-to-End Retransmission Techniques", *ACM Computer Communication Review,* Vol. 8, No. 4, Oct. 1978, pp. 1-18.
11. J.S. Kaufman, "Approximation Methods for Networks of Queues with Priorities", *Performance Evaluation,* Vol. 4, 1984, pp. 183-198.
12. A. G. Konheim, "A Queueing Analysis of Two ARQ Protocols", *IEEE Trans. on Communications,* Vol. COM-28, No 7, July 1980, pp. 1004-1014.
13. P. S. Kritzinger, "A Performance Model of the OSI Communication Architecture", *IEEE Trans. on Communications,* Vol. COM-34, No 6, June 1986, pp. 554-563.
14. P. S. Kritzinger, "A General Analysis of ARQ Protocol Performance in a Multiprocess Environment", *Performance Evaluation 11,* 1990, pp. 31-34.
15. D. Mitra and I. Mitrani, "A Go-Back-n Protocol which is efficient in High Speed Data Networks with Small Buffers", *AT&T Bell Lab Report, Jan. 1990.*
16. D. Mitra and I. Mitrani, "Asymptotic Optimality of the Go_Back_n Protocol in High Speed Data Networks with Small Buffers", *Proc. of the Int. Conference on Data Communication Systems and their Performance,* North Holland, Barcelona, June 20-23, 1990, pp. 17-31.
17. M. Moeneclaey, H. Bruneel, I. Bruyland and D-Y Chung, "Throughput Optimization for a Generalized Stop-and-Wait ARQ Scheme", *IEEE Trans. on Communications,* Vol. COM-34, No 2, Feb. 1986, pp. 205-207.
18. N. Nassehi, "Window Flow Control in Frame-relay Networks", *GLOBECOM'88, pp. 1784-1790.*
19. D. P. Pozefsky and F. D. Smith, "A Meta-implementation for Systems Network Architecture", *IEEE Trans. on Communications,* Vol. COM-30, June 1984, pp. 1348-1355.
20. M. Reiser, "Communication-System Models Embedded in the OSI-Reference Model, A Survey", *Proc. of the Computer Networking and Performance Evaluation Conf.,* Tokyo, Sept. 1985, North Holland, pp. 85-111.
21. M. Reiser and S. Lavenberg, "Mean Value Analysis of Closed Multichain Queueing Networks", *Journal of the ACM,* April 1980, pp. 313-322.
22. H. Rudin, "From Formal Protocol Specification Towards Automated Performance Prediction", *Proc. 3rd Workshop Protocol Specification, Testing, Verification,* Rueschlikon, Switzerland, May 1983, pp. 257-269.
23. B.H. Saeki and I. Rubin, "An Analysis of a TDMA Channel Using Stop-and-Wait, Block, and Select-and-Repeat ARQ Error Control", *IEEE Trans. on Communications,* Vol. COM-30, No. 5, May 1982, pp. 1162-1173.
24. T. Suda and N. Watanabe, "Evaluation of Error Recovery Schemes for a High-Speed Packet Switched Network: Link-By-Link versus Edge-To-Edge Schemes", *Proc. of the IEEE INFOCOM'88,* pp. 0722-0731.
25. C.A. Sunshine, "Efficiency of Interprocess Communication Protocols for Computer Networks", *IEEE Trans. on Communications,* Vol. COM-25, No 2, Feb. 1977, pp. 287-293.
26. D. Towsley, "The Stutter Go-Back-N ARQ Protocol", *IEEE Trans. on Communications,* Vol. COM-27, 1979, pp.869-875.
27. D. Towsley, and J.K. Wolf, "On the Statistical Analysis of Queue Lengths and Waiting Times for Statistical Multiplexers with ARQ Retransmission Schemes", *IEEE Trans. on Communications,* Vol. COM-27, No 4, April 1979, pp. 693-702.