A Method for Delay Analysis of Interacting Queues in Multiple Access Systems

Eytan Modiano

Information Technology Division
Naval Research Laboratory
Washington, D.C. 20375

Anthony Ephremides

Electrical Engineering Department
University of Maryland
College Park, MD 20742

Abstract

We develop an approximate model for analyzing interacting queues. This new approximation models an N-dimensional infinite Markov chain by means of two Markov chains, one being one-dimensional and infinite and the other being N-dimensional and finite. The transition probabilities of each chain are expressed in terms of statistics of the other chain. The two chains are solved together iteratively to yield an approximation to the original N-dimensional infinite chain. We use this approximate model to analyze systems of dependent queues which often arise in multiple a ccess protocols. We show how this model can be used to analyze the ALOHA multiple access protocol as well as a broadcast algorithm for a mesh network which was proposed in [2]. The results of our approximation compare very well with simulation.

1 Introduction

In this paper we present a new approximate model for the analysis of systems of interacting queues which often occur in various multiple access network protocols. This new model is a refinement of an existing model developed in [1] for the ALOHA multiple access protocol. We begin by applying this model to the analysis of a multiple-node broadcast algorithm for a mesh network, which was developed in [3] and presented in [2]. We then show how our model can be used to study the performance of the ALOHA multiple access protocol.

A multiple-node broadcast is a common task in the execution of parallel algorithms in a network of processors where every processor may have a message to be broadcast to all the other processors. In [3] an algorithm was developed which performs periodic, synchronized, broadcast cycles; where during each cycle only a small number of nodes are allowed to broadcast their message. Consider an N by N mesh, where each node has exogenous packets arriving (to be broadcast) independently according to a Poisson random process and placed in infinite-capacity queues. The broadcast algorithm works as follows: We partition the mesh into N vertical rings, such that each node belongs to exactly one ring. At the beginning of every broadcast cycle each ring selects, at random, up to d packets to be broadcast throughout the mesh. The broadcast of the d packets from each ring is performed and has a fixed duration of $(d+1)(N-1)$ time slots. Clearly, the queues at the N nodes on each ring are highly dependent on each other. In fact, the queue sizes of the N nodes on each ring form an N-dimensional infinite Markov chain. Obtaining analytic expressions for the steady state behavior of such a system is very difficult. Even a numerical evaluation of such systems can be computationly prohibitive [3]. A similar difficulty arises in the analysis of the Aloha multiple access protocol and no exact analysis for packet delay is known for that case either. Several approximate models have been proposed for the analysis of ALOHA which may be useful in analyzing this broadcast algorithm.

In [1], Ephremides and Saadawi developed an approximate model for a system of interacting queues for analyzing the ALOHA protocol. In their model they approximate a system of N infinite queues as a single dimensional infinite Markov chain representing the state of one user together with an N-dimensional finite Markov chain representing the state of the rest of the system. They use parameters from the solution of one chain in analyzing the other and solve the two chains together using an iterative algorithm. This two chain approach tracks the interaction between the different users in a system model that can be analyzed. We develop a similar approximate model for the system of interacting queues in the mesh broadcast case. Our new model is a refinement of the model in [1] and is shown to perform much better when compared to simulation.

Since our refined model offers such an improvement to the original model with a minimal additional complexity, we were motivated to develop a similar model for the ALOHA multiple access protocol. In the ALOHA case the state of any single user can be specified by its queue size and by the indication of whether it is in the blocked or active states. A complete description of a N-terminal system requires the analysis of a 2N-dimensional infinite Markov chain. Again, such chains are known to be very difficult to analyze and require an approximation.

As was stated earlier, in [1] an approximation was developed which modeled an N-dimensional infinite Markov chain as a one-dimensional infinite chain representing the state of a single user together with a N-dimensional finite chain representing the number of blocked and active users in the entire system. In [5] an improvement to the above model was proposed which expanded the system chain to include the identity of all N users. That expanded model was shown to perform far better than the model in [1]; however, the expanded system chain contained 3^N states and was very difficult to analyze for all but very small values of N. We therefore develop a new system chain, similar to the one developed for the multiple-node broadcast algorithm, which improves the performance of the approximation while keeping the computation complexity of the approximate model low.

2 The Broadcast Algorithm

In this section we present an approximate model for analyzing the multiple node broadcast algorithm for a mesh presented in [2]. The algorithm is based on performing "periodic broadcast cycles" and works as follows: We partition the $N \times N$ mesh into N vertical rings and N horizontal rings so that each node is contained in exactly one vertical and one horizontal ring, and proceed according to the following set of steps which are repeated periodically:

Step 1) Every node broadcasts one packet along its vertical ring, so that all nodes on the same vertical ring possess N packet, one from each node on the ring.

Step 2) Every ring selects, at random, d packets to be further broadcast throughout the mesh. The un-selected packets rejoin their node's queues and re-attempt transmission during the next cycle. (If a ring has fewer than d packets then the remaining slots are filled with null packets). Clearly, all nodes on a given vertical ring have the same d packets, to be broadcast through the mesh.

step 3) To complete the broadcast cycle all nodes send the d "select" packets throughout their horizontal rings.

The first step takes N-1 slots and the third step takes an additional d(N-1) slots. Therefore, the broadcast cycle is repeated every S=(d+1)(N-1) slots. Since there are N nodes on the ring and up to d of them can receive service during a cycle of duration (d+1)(N-1), in order for the algorithm to be stable we must have $\lambda \leq \frac{d}{N(d+1)(N-1)}$.

We would like to compute the average delay in this system. Because of the dependence between the N queues, the queue sizes in a ring of N nodes form an N-dimensional infinite Markov chain which is very difficult to analyze. Some approximate models have been proposed for the analysis of the ALOHA system which compare very well with simulation results. Because of the similarity between the two problems we are lead to consider similar approximations for our system. The basic idea behind these models was to split the system into two Markov chains. One single dimensional, infinite, chain, termed the "user chain", tracking the state of a single user and the other an N-dimensional, finite, Markov chain, termed the "system chain", tracking the state of the rest of the system. In [1] Saadawi and Ephremides let the user chain represent the number of packet a particular user has in queue, and the system chain represent the number of active and idle nodes in the system. In [5] Zhu and Ephremides consider an extension to the first approximation by having the system state represent the identity and state of every node (Idle, Active, or Blocked). Simulation results show that the latter approximation is better than the former; however, having such an expanded system state is costly in that the system Markov chain becomes difficult to analyze. Numerical solution to the system state in the first model require the solution to a set of N linear equation where as for the second model the solution of 3^N linear equations. In fact, the solution to the second model is not much simpler than to that of the original model. In evaluating the performance of their model Zhu and Ephremides are restricted to very small values of N.

2.1 The Two-chain Model

The idea behind having two Markov chains, one for a single user and the other for the rest of the nodes, is that while a single infinite chain with N dimensions is very difficult to analyze, each of the two smaller chains are analyzable and when solved together provide an approximation to the original chain. The one dimensional infinite user chain represents the buffer size and state for a single user and can be solved by

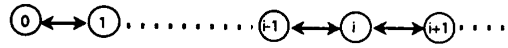

Figure 1: The User Chain.

conditioning on the state of the system chain. In turn, the system chain tracks the interaction between the different users.

2.1.1 The User's Markov Chain

The user's Markov chain represents the queue size for a single user. It is therefore an infinite chain. Arrivals are Poisson and departures are geometrically distributed with parameter P_s representing the probability that a node is "selected" during a given cycle. So the user chain is an M/G/1 with geometrically distributed service time. Figure 1 shows the user chain.

This, of course, constitutes an approximation to the real model because we assume that the probability of a successful transmission is independent of the number of packets in the queue. The average delay for this system can be easily computed using the well known formulas for an M/G/1 and the first and second moments of the geometric distribution.

$$delay = \frac{S}{2} + \frac{\lambda S(2 - \lambda S)}{2\lambda(P_s - \lambda S)}, \quad (1)$$

where the $\frac{S}{2}$ term accounts for the synchronization delay.

So, this completes the analysis of the user chain. The only missing ingredient, in order to compute the delay, is P_s. This is the one term that we will obtain from the system chain. Before we turn our attention to the system chain, there is one parameter from the user chain which will be needed later for the analysis of the system chain. We will need to know the probability that there is exactly one packet in the queue. This is easily derived as follows, let $\Pi(i)$ be the probability that the user's system is in state i. Then, $\Pi(0) = 1 - \bar{A}_x/P_s = 1 - \lambda S/P_s$. We can next express $\Pi(1)$ in terms of $\Pi(0)$ by writing the steady state flow equation out of state zero. So, $\Pi(0) = A(0)\Pi(0) + A(0)P_s\Pi(1)$, where A(0) is the probability of having no arrivals during a slot, and is equal to $\exp(-\lambda S)$. So, $\Pi(1) = (P_s - \lambda S(1 - A(0)))/(A(0)P_s^2)$.

2.1.2 The System Markov Chain

All of our approximations for this system involving two Markov chains have the same user chain analyzed above; they, therefore, differ only in the system chain model and how P_s is computed using that system chain. We consider three different models for the system chains.

The first model is based on the Saadawi model for approximating the ALOHA protocol and the system chain simply represents the number of non-empty nodes on one ring (the ring containing our node of interest). Clearly, this chain consists of N+1 states. The transition probabilities between these states can be expressed in terms of parameters from the user chain. If we let S_i denote the i^{th} state of the system with i non-empty and N-i empty nodes and P_i denote the steady state probability of S_i then P_s is expressed by $P_s = (\sum_{i=1}^{i=d} P_i + d\sum_{i=d+1}^{i=N} \frac{1}{i} P_i)/(1 - P_0)$.

Since the system chain equations depend on parameters from the user's chain and visa versa, the two chains are solved together using an iterative algorithm described in [3]. The results from our approximate model compare reasonably well with simulation, particularly when arrival rates are low.

Our second system model is based on the Zhu model for approximating ALOHA where the system chain includes the identity of the individual queues and their status (empty or non-empty). This adjustment to the system model proved to dramatically improve the performance of the approximation. However, with this change the system chain consists of 2^N states and is difficult to solve for all but very small values of N.

To overcome this shortcoming of the expended model we limit the system chain, in our third model, so that it merely represents the identity and state of one user along with the number of non-empty nodes on the ring. Having the system chain contain the state of a single user allows us to more accurately derive the probability of success for the user chain. This is because the probability of success is defined to be the probability that the user is chosen given that it is non-empty. Therefore, when the system chain contains the state of our user, we can compute the probability of success by conditioning on the user state being non-empty. It turns out that this new model is just as accurate as the previous model (containing the identities of all of the users) but since this new chain has only 2(N+1) states it is much easier to analyze.

The complete details of the first two system models are presented in [3]. Here we focus our attention only on the third model. Our new system chain will include information on whether or not node x is empty as well as the total number of empty nodes in the system. Let the pair (S, D_x) represent the state of the system, where S equals the number of non-empty nodes and D_x is equal to zero if node x is empty and one other-

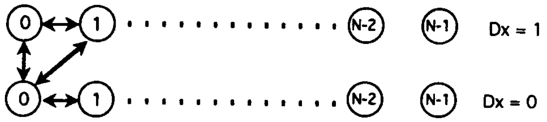

Figure 2: The system chain.

wise. Clearly, there are a total of 2N possible states. Figure 2 shows the system chain.

In order to express the transition probabilities for this chain we define the following two quantities.

Let $B_{(i,j)}(k)$ be the probability that k empty nodes become full given that we started in state (i,j) and,

$C_{i,j}(k,l)$ be the probability that k full nodes become empty, and that D_x goes from j to l.

We can now write the probability of going from state (i,j) to (i',j') as follows,

$$P((i,j),(i',j')) = \sum_{k=0}^{k=N} B_{(i,j)}(k) C_{i,j}(k-(i'-i),j').$$

The $B_{(i,j)}(k)$ term does not depend on the state of node x and is simply the probability that k of the N-i empty nodes receive packets. So,

$$B(i,j)(k) = \begin{cases} \binom{N-i}{k} (1-A(0))^k A(0)^{N-i-k} & \text{if } k \leq N-i \\ 0 & \text{otherwise} \end{cases}$$

The $C_{i,j}(k,l)$ term does depend on the state of node x. Also, it requires P_e, the probability that a node empties upon receiving service. That probability can be expressed in terms of the user chain steady-state probabilities by $P_e = \Pi(1)A(0)/(1-\Pi(0))$, where the numerator is the probability that the node had one packet in the queue and received no new packets while the denominator represents the probability that the queue was not empty (a necessary condition in order for the node to receive service). Now the computation of $C_{i,j}(k,l)$ is trivial, though cumbersome because it depends on the values of the parameters and involves many cases. The probabilities for all the different cases are presented next without proof. Their derivation is presented in [3].

$C_{(i,0)}(k,0) =$

$$= \begin{cases} A(0) \binom{d}{k} P_e^k (1-P_e)^{d-k} & \text{if } k \leq d \leq i \\ A(0) \binom{i}{k} P_e^k (1-P_e)^{i-k} & \text{if } k \leq i \leq d \\ 0 & \text{if } j > d \text{ or } j > i \end{cases}$$

$C_{(i,0)}(k,1) =$

$$= \begin{cases} (1-A(0)) \binom{d}{k} P_e^k (1-P_e)^{d-k} & \text{if } k \leq d \leq i \\ (1-A(0)) \binom{i}{k} P_e^k (1-P_e)^{i-k} & \text{if } k \leq i \leq d \\ 0 & \text{if } k > d \text{ or } k > i \end{cases}$$

$C_{(i,1)}(k,0) =$

$$= \begin{cases} \frac{d}{i+1} P_e \binom{d-1}{k} P_e^k (1-P_e)^{d-1-k} & \text{if } k < d \leq i \\ P_e \binom{i}{k} P_e^k (1-P_e)^{i-k} & \text{if } k \leq i < d \\ 0 & \text{if } k \geq d \text{ or } k > i \end{cases}$$

$C_{(i,1)}(k,1) =$

$$= \begin{cases} \frac{i+1-d}{i+1} P_e \binom{d}{k} P_e^k (1-P_e)^{d-k} + \\ +(1-P_e)\frac{d}{i+1} \binom{d-1}{k} P_e^k (1-P_e)^{d-1-k} & \text{if } k \leq d \leq i \\ (1-P_e) \binom{i}{k} P_e^k (1-P_e)^{i-k} & \text{if } k \leq i < d \\ 0 & k > d \text{ or } k > i \end{cases}$$

These equations completely specify the transition probabilities for the system chain. Next we need to compute P_s, the probability of node x having a successful transmission with this system chain. $P_s = (\sum_{i=0}^{i=d-1} P_{i,1} + \sum_{i=d}^{i=N-1}(d/(i+1))P_{i,1})/(\sum_{i=0}^{i=N-1} P_{i,1})$. The term $P_{i,j}$ is the probability of being in state (i,j). The top part of this fraction represents the probability of success when the system chain is in one of the states with node x full. The bottom represents the probability of the system being in one of these states.

We have now expressed the system chain in terms of the user chain parameter P_e and the user chain in terms of the system chain parameter P_s. The two chains can now be solved together using an iterative algorithm to obtain the results for the approximation.

2.2 Numerical Results

We have tested our approximate model, using the three different system chains for many values of N, d and λ. In general the approximation works well for low arrival rates and deteriorates when the arrival rate approaches saturation. Of the three system chains, the first chain which included no information about the user's identity was the least accurate. The second system model, tracking the state of every user produced exactly the same results as the third model tracking only the state of one user. This result was somewhat surprising because it implies that having the state of all the users in the chain provides no additional information over having the state of only the one user. In figure 3 we plot delay vs. λ for N=4 and d=1. The line labeled approx1 represents the system model which contained no information about the individual user's

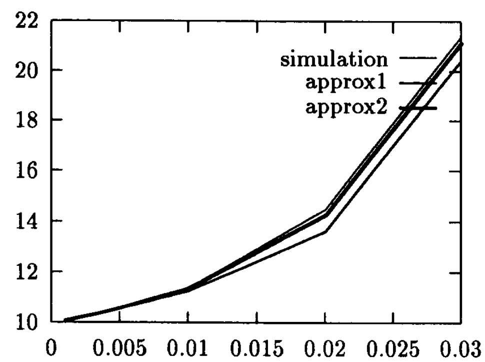

Figure 3: Delay vs. λ with N=4 and d=1.

identity and the line labeled approx2 represents the other two system models which performed the same. Plots for other values of N, d and λ are available in [3].

3 The Multiple Access Protocol

The two Markov chain approximate model proposed for the broadcast algorithm of the previous section is based on a similar model originally used in [1] to approximate the performance of the ALOHA multiple access protocol. We showed in the previous section that by altering the original model so that the system chain also represents the identity and state of one user the approximation improved substantially. In fact the performance of our expended model was identical to that of the model proposed in [5] where the system chain included the identity of all the users. Of course, our model had the advantage of being much easier to solve for; while the model in [5] required the solution of $O(2^N)$ linear simultaneous equations our model required only $O(N)$. In fact the model in [5] which was first proposed to study the ALOHA protocol had as its main drawback a high computation complexity and was solvable only for a system with a few users while our expanded model had a much lower computation complexity and was easily solvable for systems with many users. It is therefore of interest to develop a similar model for ALOHA.

In an ALOHA system, a number of users share a single channel over which they wish to transmit information. The channel is available to any terminal whenever it has a packet ready for transmission. A transmission is successful if and only if no other terminal attempts transmission during that same time slot. Sometimes, however, two or more terminals attempt transmission during the same slot. In such cases all of the colliding packets will have to be retransmitted during a later slot. Clearly, if all of these terminals were to attempt retransmission in the next slot they will collide again. We therefore need a method to schedule retransmissions so that collisions can be avoided. In the ALOHA system, after a collision a retransmission takes place in each of the subsequent slots with some probability, until the transmission is successful.

The terminal for a single user consists of an infinite buffer and a retransmission unit. Each user has packets arriving independently according to a Bernoulli random process. If a terminal is empty (has no packets), a newly arrived packet is transmitted immediately. The transmission is successful if and only if no other user attempts transmission during the same slot, otherwise a collision occurs and the terminal enters the blocked state, the colliding packets do not rejoin their queues but rather enter the retransmission unit where they await retransmission. When in the blocked state, the terminal attempts retransmission with probability p. In case of success the terminal becomes unblocked. An unblocked terminal can be in one of two states; idle (when its queue is empty), or active (when its queue is not empty). An active terminal transmits a packet with probability one.

The state of any single user can be specified by its queue size and by the indication of whether it is in the blocked or active states. A complete description of a N-terminal system requires the analysis of a 2N-dimensional infinite Markov chain. Again, such chains are known to be very difficult to analyze.

3.1 The Two-chain Model

As was stated earlier, in [1] an approximation was developed which modeled an N-dimensional infinite Markov chain as a one-dimensional infinite chain representing the state of a single user together with a N-dimensional finite chain representing the number of blocked and active users in the entire system. In [5] an improvement to the above model was proposed which expanded the system chain to include the identity of all N users. That expanded model was shown to perform far better than the model in [1]; however, the expanded system chain contained 3^N states and was very difficult to analyze for all but very small values of N. We therefore develop a new system chain, similar to the one developed for the multiple-node broadcast algorithm, which includes the state of only one user

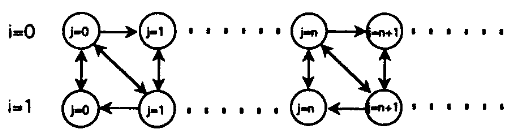

Figure 4: The User Chain

together with the number of active and blocked users in the entire system.

3.1.1 The User Markov Chain

The state of a single user can be characterized by the size of its buffer content and by the indication of whether it is blocked or unblocked. We denote the user state by the pair (i,j) where i is 0 if the terminal is blocked and 1 if it is unblocked (idle or active) and j is the number of packets in the buffer. Note that when both i and j are specified one can also determine whether the user is idle or active. Figure 4 shows the user chain.

Let π_{ij} be the steady-state probability of the state (i,j) and let

$$G_i(z) = \sum_{n=0}^{\infty} \pi_{i,n} z^{-n}, \quad i = 0, 1.$$

In order to express the transition probabilities of the user Markov chain we define the following quantities:

r = Pr[successful transmission/user is blocked]

q_1 = Pr[collision/user is idle]

q_2 = Pr[collision/user is active].

In the next section we will derive expressions for these quantities in terms of the system Markov chain steady-state probabilities. We can now express the following quantities relating the statistics of the user chain, and some of which will later be used to determine the transition probabilities of the system Markov chain.

$$\pi_{1,0} = \frac{r(1-\sigma) - \sigma q_2}{r(1-\sigma) - \sigma(q_2 - q_1)} \tag{2}$$

$$G_0(1) = \frac{(1-\sigma)\sigma q_1}{r(1-\sigma) - \sigma(q_2 - q_1)} \tag{3}$$

$$G_1(1) = \sigma + (1-\sigma)\frac{r(1-\sigma) - \sigma q_2}{r(1-\sigma) - \sigma(q_2 - q_1)} \tag{4}$$

$$\pi_{1,1} = \frac{\sigma}{1-\sigma}\pi_{0,0} \tag{5}$$

$$\pi_{0,0} = \frac{\sigma q_1}{\sigma(1-q_2) + r(1-\sigma)}\pi_{1,0} \tag{6}$$

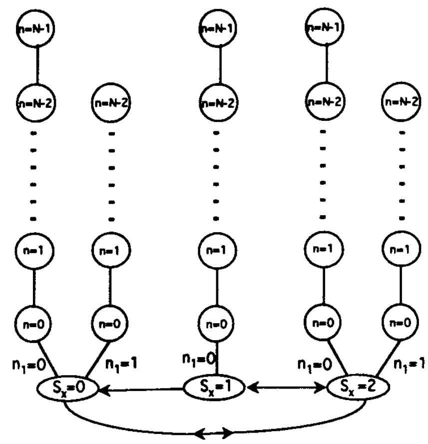

Figure 5: The System Chain.

and Q, the average queue size in the buffer is expressed by

$$Q = \frac{\sigma^2(1-\sigma)q_1}{[r(1-\sigma) - \sigma q_2][[r(1-\sigma) - \sigma(q_2 - q_1)]}. \tag{7}$$

Finally, the average delay can be expressed by

$$D = W + S \tag{8}$$

where W is the average waiting time in the queue and S is the average service time. The average waiting time in the queue can be expressed using littles result and is equal to Q/σ. The average service time is expressed according to the following equation

$$\frac{G_1(1) - \pi_{1,0}}{G_1(1)}\left[1 - q_2(1 - \frac{1}{r})\right] + \frac{\pi_{1,0}}{G_1(1)}\left[1 - q_1(1 - \frac{1}{r})\right] \tag{9}$$

All of the above equations were computed in [1] and for bravity are omitted here.

3.1.2 The System Chain

The system chain in [1] was described by the number of terminals that are in each of the three states, blocked, active, and idle. Here we extend the system chain to include the identity and state of a single arbitrary user, x. So we let the state of the system, S, be the tuple (S_x, n, n_1), where S_x is the state of node x and is either blocked (2), active (1), or idle (0) ; n is the number of blocked terminals and takes values between 0 and N-1; and n_1 is the number of active terminals and that can be at most 1, since we can not have more than one active terminals at a time. Clearly, the number of idle terminals is equal to N-1-(n+n_1). Figure 5 shows the system chain.

We can now express the quantities r,q_1, q_2, in terms of the system chain's steady-state probabilities. For simplicity we denote the steady-state probability of the system chain being in state S by Pr[S]. The following quantities are trivial to arrive at and therefore their derivations are omitted.

$$r = \frac{p\sum_{i=0}^{N-1} Pr[(2,i,0)](1-p)^i(1-\sigma)^{(N-1-i)}}{\sum_{i=0}^{N-1} Pr[(2,i,0)] + \sum_{i=0}^{M-2} Pr[(2,i,1)]} \quad (10)$$

where the numerator represents the probability that the blocked user (user x) attempts transmission and no other user transmits and the denominator is the probability that user x is blocked.

Similarly,

$$q_1 = 1 - \frac{\sum_{i=0}^{N-1} Pr[(0,i,0)](1-p)^i(1-\sigma)^{(N-1-i)}}{\sum_{i=0}^{N-1} Pr[(0,i,0)] + \sum_{i=0}^{N-2} Pr[(0,i,1)]} \quad (11)$$

and

$$q_2 = 1 - \frac{\sum_{i=0}^{N-1} Pr[(1,i,0)](1-p)^i(1-\sigma)^{(N-1-i)}}{\sum_{i=0}^{N-1} Pr[(1,i,0)]}. \quad (12)$$

We are now ready to derive the state transition probabilities for the system chain. For simplicity we break down our expressions according to the state of node x. We denote the transition probability from state S1 to S2 by P(S1,S2).

Let E be the probability that the buffer is empty given that the user is blocked. It can be shown that E is also equal to the probability that the buffer size equals one given that the user is active [1]. Therefore, $E = \pi_{00}/G_0(1) = \pi_{11}/(G_1(1) - \pi_{10})$. E is the one quantity from the user chain that will be used in expressing the transition probabilities for the system chain.

Let $B_j(n)$ be the probability that j of n blocked nodes attempt transmission and $I_j(n)$ the probability that j of the N-1-n idle nodes attempt transmission. Then, clearly, $B_j(n) = \binom{n}{j} p^j(1-p)^{n-j}$ and $I_j(n) = \binom{N-1-n}{j} \sigma^j(1-\sigma)^{N-1-n-j}$. These two quantities will be very useful in expressing the transition probabilities for the system chain. These transition probabilities are presented next without proof. Their derivation is trivial and can be verified by the reader.

$P((0,n,n_1),(1,n',n_1')) = 0$

$P((0,n,n_1),(2,n',n_1')) =$

$$= \begin{cases} \sigma(1-B_0(n))I_0(n) & \text{if } n = n', n_1 = n_1' = 0 \\ \sigma(I_{n'-n}(n)) & \text{if } n < n', n_1 = n_1' = 0 \\ \sigma(I_{n'-n-1}(n+1) & \text{if } n+1 < n', n_1 = 1, n_1' = 0 \\ \sigma I_0(n+1) & \text{if } n' = n+1, n_1 = 1, n_1' = 0 \\ 0 & \text{otherwise} \end{cases}$$

$P((0,n,n_1),(0,n',n_1')) =$

$$= \begin{cases} (1-\sigma)I_{n'-n}(n) & \text{if } n' > n+1, n_1 = n_1' = 0 \\ (1-\sigma)I_1(n)(1-B_0(n)) & n' = n+1, n_1 = n_1' = 0 \\ (1-\sigma)[I_0(n)(1-B_1(n))+ & \\ +I_1(n)B_0(n)]+ & \\ +\sigma I_0(n)B_0(n) & n' = n, n_1 = n_1' = 0 \\ (1-\sigma)I_0(n)B_1(n)E & n' = n-1, n_1 = n_1' = 0 \\ (1-\sigma)I_0(n)B_1(n)(1-E) & n' = n-1, n_1 = 0, n_1' = 1 \\ (1-\sigma)(1-B_0(n))I_0(n+1) & n' = n+1, n_1 = 1, n_1' = 0 \\ (1-\sigma)I_{n'-n-1}(n+1) & n' > n+1, n_1 = 1, n_1' = 0 \\ (1-\sigma)B_0(n)I_0(n+1)E & n' = n, n_1 = 1, n_1' = 0 \\ (1-\sigma)B_0(n)I_0(n+1)(1-E) & n' = n, n_1 = 1, n_1' = 1 \\ 0 & \text{otherwise} \end{cases}$$

$P((1,n,n_1),(0,n',n_1')) =$

$$= \begin{cases} B_0(n)I_0(n)E & \text{if } n = n', n_1 = n_1' = 0 \\ 0 & \text{otherwise} \end{cases}$$

$P((1,n,n_1),(1,n',n_1')) =$

$$= \begin{cases} B_0(n)I_0(n)(1-E) & \text{if } n = n', n_1 = n_1' = 0 \\ 0 & \text{otherwise} \end{cases}$$

$P((1,n,n_1),(2,n',n_1')) =$

$$= \begin{cases} I_0(n)(1-B_0(n)) & \text{if } n = n', n_1 = n_1' = 0 \\ I_{n'-n}(n) & n < n', n_1 = n_1' = 0 \\ 0 & \text{otherwise} \end{cases}$$

$P((2,n,n_1),(0,n',n_1')) =$

$$= \begin{cases} pEI_0(n)B_0(n) & \text{if } n = n', n_1 = n_1' = 0 \\ 0 & \text{otherwise} \end{cases}$$

$P((2,n,n_1),(1,n',n_1')) =$

$$= \begin{cases} p(1-E)I_0(n)B_0(n) & \text{if } n = n', n_1 = n_1' = 0 \\ 0 & \text{otherwise} \end{cases}$$

$P((2,n,n_1),(2,n',n_1')) =$

$$= \begin{cases} I_{n'-n}(n) & \text{if } n' > n+1, n_1 = n_1' = 0 \\ (1-p)I_1(n)(1-B_0(n))+ & \\ +pI_1(n) & n' = n+1, n_1 = n_1' = 0 \\ (1-p)[I_0(n)(1-B_1(n))+ & \\ +I_1(n)B_0(n)]+ & \\ +pI_0(n)(1-B_0(n)) & n' = n, n_1 = n_1' = 0 \\ (1-p)I_0(n)B_1(n)E & n' = n-1, n_1 = n_1' = 0 \\ (1-p)I_0(n)B_1(n)(1-E) & n' = n-1, n_1 = 0, n_1' = 1 \\ (1-p)(1-B_0(n))I_0(n+1)+ & \\ +pI_0(n+1) & n' = n+1, n_1 = 1, n_1' = 0 \\ (1-p)I_{n'-n-1}(n+1) & n' > n+1, n_1 = 1, n_1' = 0 \\ (1-p)B_0(n)I_0(n+1)E & n' = n, n_1 = 1, n_1' = 0 \\ (1-p)B_0(n)I_0(n+1)(1-E) & n' = n, n_1 = 1, n_1' = 1 \\ 0 & \text{otherwise} \end{cases}$$

As can be seen from figure 5 the system chain consists of 5N-2 states. The state transition matrix for this chain can be expressed in terms of E, a user chain parameter, using the above equations. Similarly the user chain can be solved in terms of system chain parameters according to equations (3)-(7). As before, the two chains can now be solved together using an iterative algorithm to obtain results for the approximate model.

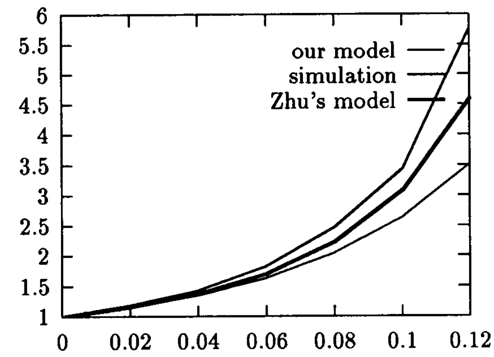

Figure 6: Results for approx model with N=3 and p=0.3.

3.2 Numerical Results

Again we evaluated our model for many values of N and p, and compared our results to those obtained via the Zhu expended model as well as simulation. In general both approximations worked well when the arrival probability was low. When that probability approached saturation the performance of both models deteriorated significantly. Consistent throughout our results is that our refined model did not perform as well as the Zhu model this time. This, of course, is disappointing because it was our hope that the two model would give the same results, as was the case in the broadcast algorithm model. In figure 6 we plot Delay vs. σ with N=3 and p=0.3 for this model as well as for the Zhu model and simulation. Plots for other values of N, p and σ are available in [3].

4 Conclusion

We present a new model for evaluating the performance of dependent queues. Our model is a refinement of existing models used in the evaluation of the ALOHA multiple access protocol and proved to be extremely useful in evaluating the performance of a broadcast algorithm for a mesh network. We believe that this model may prove to be useful in many other systems of interacting queues and in particular for multiple access schemes such as the two presented in this paper. It is of interest to find other systems with dependent queues, not necessarily involving multiple access, for whose analysis this model, or a two chain model in general, may prove to be useful.

Additionally, further modifications to this two chain model may also prove useful; although they may increase its computational complexity. One possible modification will expand the system chain to include the buffer size for one of the users. Of course, in order to keep the chain finite, this quantity will have to be truncated. With the additional information in the system chain, it would then be possible to derive transition probabilities for the user chain which take its buffer size into account. This additional refinement should further improve the performance of the approximation.

References

[1] T. N. Saadawi and A. Ephremides, "Analysis, Stability, and Optimization of Slotted ALOHA with a Finite Number of Buffered Users," *IEEE Transactions on Automatic Control*, June, 1981.

[2] E. Modiano and A. Ephremides, "Efficient Routing Schemes for Multiple Broadcasts in a Mesh" *Twenty-Sixth annual Conference on Information Sciences and Systems,* Princeton, NJ, March 1992.

[3] E. Modiano, "Security and Performance Issues in Distributed Protocols," Ph.D. Dissertation, Department of Electrical Engineering, The University of Maryland, College Park, 1992.

[4] T. Nakasis and A. Ephremides, "Steady-State Behavior of Interacting Queues - A Numerical Approach," *IEEE Transactions on Information Theory*, March, 1990.

[5] A. Ephremides and R. Z. Zhu, "Delay Analysis of Interacting Queues with an Approximate Model" *IEEE Transactions on Communications*, Feb., 1987.