Performance Analysis of k-Fold Multicast Networks

Zhenghao Zhang and Yuanyuan Yang

Dept. of Electrical & Computer Engineering, State University of New York, Stony Brook, NY 11794, USA

Abstract— **Multicast involves transmitting information from a single source to multiple destinations, and is an important operation in high-performance networks. A k-fold multicast network was recently proposed [7] as a cost-effective solution to provide better QoS functions in supporting multicast communication. To give a quantitative basis for network designers to determine the suitable value of system parameter k under different traffic loads, in this paper we propose an analytical model for the performance of k-fold multicast networks under Poisson traffic. We first give the stationary distribution of network states and then derive the throughput and blocking probability of the network. We also conduct simulations to validate the analytical model and the results show that the analytical model is very accurate under the assumptions we make. The analytical and simulation results reveal that by increasing the fold of the network, network throughput increases almost exponentially when the fanouts of multicast connections are relatively small compared to the network size.**

I. Introduction

Multicast involves transmitting information from a single source to multiple destinations, and is an important operation in high-performance networks. Multicast will be increasingly used to support various interactive applications such as multimedia, teleconferencing, web servers and electronic commerce on the Internet, as well as communication-intensive applications in parallel and distributed computing systems, such as distributed database updates and cache coherence protocols. Many of these applications require not only multicast capability but also predictable communication performance, called quality-of-service (QoS). The combination of the non-uniform nature of multicast traffic and the requirement of QoS guarantees makes the problem very challenging.

There has been much work in the literature on multicast communication in various networks, see, for example, [1]-[7]. Several researchers [1], [2], [3], [4] have considered supporting *multicast assignments* in switch-based networks in a nonblocking or rearrangeable fashion. A multicast assignment is a mapping from a subset of network source nodes to a subset of network destination nodes with no overlapping allowed among the destinations of different sources. However, in real-world multicast applications, multicast communication patterns are not necessarily multicast assignments and overlapping among destinations of different multicast connections may be possible. A simple example is that a destination node may be simultaneously involved in two multicast connections. Such connections will be blocked in a network which is designed to be nonblocking or rearrangeable for only multicast assignments.

To overcome this problem, recently [7] presented a design for a nonblocking k-fold multicast network, which can provide better QoS functions for arbitrary multicast communication. Specifically, the network can realize multiple, say, k multicast assignments, in a single pass with a guaranteed latency. A *k-fold multicast assignment* is defined as a mapping from a subset of network source nodes to a subset of network destination nodes with up to k-fold overlapping allowed among the destinations of different sources. In other words, any destination node can be involved in multicast connections from up to k different sources at a time. A network which can realize any k-fold multicast assignments in a single pass is referred to as a *k-fold multicast network*. Clearly, an ordinary multicast network is a 1-fold multicast network. Here k is an adjustable parameter in network design and an appropriate value of k should be determined by the multicast traffic in the target multicast applications, specially by the statistics of destination overlapping in multicast connections. Note that although k-fold multicast assignments can be realized by simply stacking k copies of 1-fold networks together, the k-fold network designed in [7] has much lower hardware cost. In fact, the cost of the former is about 3 to k times of a k-fold network for any k.

The research work was supported in part by the U.S. National Science Foundation under grant numbers CCR-0073085 and CCR-0207999.

To provide a quantitative basis for network designers to choose the suitable value of system parameter k under different traffic loads, in this paper we propose an analytical model for analyzing the performance of a k-fold multicast network. Although there has been a lot of research on performance analysis in various networks under multicast traffic, see, for example, [5], [6], [9], none of them has considered network performance in terms of k-fold multicast assignments. In this paper, we derive the *blocking probability* and the *throughput* of a k-fold multicast network under Poisson traffic and validate the model through simulations. Based on the assumptions we make, we first show that the number of ongoing multicast connections in the network is a continuous time Markov chain. The network throughput and blocking probability can then be obtained in terms of the stationary distribution of the Markov chain. We also conduct simulations to validate the analytical results.

II. Preliminaries

In this section, we give the definitions and assumptions we will use in this paper. First, we will need following definitions:

Definition 1: A multicast connection refers to that a source node is connected to multiple destination nodes simultaneously in the network, and is sending the same message to these destination nodes. In this paper, we sometimes simply refer to it as a connection.

Definition 2: We assume there are no buffers at source nodes. Input blocking refers to the case that a multicast connection request arrives at a busy source node and is blocked.

Definition 3: If a destination node is connected to m source nodes, we say that this destination node is of degree m.

Definition 4: Output blocking refers to the case that a mul-

0-7803-7632-3/02/$17.00 ©2002 IEEE

ticast connection request arrives at an idle source node but is blocked because it requests some destination node of degree k.

Definition 5: We define output blocking ratio as the ratio of the requests blocked due to output blocking over all the requests blocked.

Definition 6: If a group of multicast connections can be transmitted simultaneously through the network without any blocking, we say that they are mutually compatible and abbreviate it as $m.c.$ Clearly, in a k-fold multicast network, only those multicast connections that can fit into a k-fold multicast assignment are $m.c.$

Definition 7: We say that the network is in state i when there are i ongoing multicast connections in the network.

Definition 8: We define the average number of successful multicast connection requests carried by the network in a unit time as carried throughput, or simply throughput.

In addition, throughout this paper, we make following assumptions on the multicast traffic we consider: (1) The probability of a destination node being involved in an incoming multicast connection request is θ and is independent of other destination nodes. Thus, the fanout of a multicast connection follows Binomial distribution. (2) Multicast connection requests at different source nodes are independent of each other. (3) Service time of each multicast connection is exponentially distributed with parameter μ and is independent of each other. (4) Multicast connection requests arrive at each source node according to a Poisson process with intensity λ and are independent of each other.

III. Stationary Distribution of Network States

Based on the assumptions made in the previous section, we now derive the stationary distribution of network states.

Consider an $N \times N$ k-fold multicast network. Given that there are i multicast connections in the network, let $p_{deg}(i,m)$ be probability that a destination node has degree m. A destination node is of degree m if among the i multicast connections, this node is the destination of exactly m multicast connections. The probability that any multicast connection request chooses this destination node is θ and is independent of other multicast connections. Thus, we have

$$p_{deg}(i,m) = \binom{i}{m} \theta^m (1-\theta)^{i-m},\ m \in \{0,1,...,i\}, \quad (1)$$

which is a Binomial random variable. Since each destination node in a multicast connection is selected independently, the degrees of all N destination nodes in the network are independent of each other and have the same distribution given by equation (1).

Let $P_{mc}(i)$ be the probability that i multicast connection requests are $m.c.$ in a k-fold multicast network. Recall that a set of multicast connection requests are $m.c.$ when none of the destination nodes has a degree more than k when realized simultaneously in the network. From equation (1), we know the probability that a destination node having a degree less than or equal to k is $\sum_{m=0}^{k} p_{deg}(i,m)$ for $i > k$, and 1 for $i \le k$ because when $i \le k$, no destination node can have a degree more than k. Since the degrees of destination nodes are independent of each other, we have

$$P_{mc}(i) = \begin{cases} \left(\sum_{m=0}^{k} p_{deg}(i,m)\right)^N & i > k \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

Suppose a new multicast connection request arrives when there are i multicast connections in the network. If this new connection can be realized along with those ongoing connections, we say it can "join" them. Let $P_{jn}(i)$ be the probability that a new multicast connection can join i ongoing connections. We have

$$P_{jn}(i) = \frac{P_{mc}(i+1)}{P_{mc}(i)} \quad (3)$$

To see this, let E_1 be the event that a new multicast connection request and the existing i multicast connections are $m.c$, E_2 be the event that i multicast connections are $m.c$, and E_3 be the event that $i+1$ multicast connections are $m.c.$ We have

$$P_{jn}(i) = P(E_1|E_2) = \frac{P(E_1,E_2)}{P(E_2)} = \frac{P(E_3)}{P(E_2)} = \frac{P_{mc}(i+1)}{P_{mc}(i)}$$

Suppose the network is carrying i multicast connections, or in state i. The network state may change when some connection requests arrive at the network and can join the ongoing connections, or when some ongoing connections are terminated. Since we assume that the arrival process is a Poisson process and the connection holding time follows exponential distribution, the probabilities that more than one connection requests arrive at the same time, more than one ongoing connections terminate at the same time, and a connection request arrives and an ongoing connection terminates at the same time are very small and therefore can be neglected. Thus, state i may only transit to state $i+1$ or $i-1$ (for $0 < i < N$). A transition to state $i+1$ occurs when a new connection request arrives at an idle source node and can join the i ongoing connections. A transition to state $i-1$ occurs when one of the ongoing connections terminates.

Since the arrival process is Poisson and the service time is exponentially distributed, the network state is a continuous time Markov chain [8]. We now derive the state transition rate. First, consider the transition from state i to state $i-1$. In state i there are i ongoing multicast connections, and service times of these connections are exponentially distributed with parameter μ and independent of each other. Thus, the transition rate to state $i-1$ is $i\mu$. Now consider the transition from state i to state $i+1$. We know that there are $N-i$ idle source nodes in the network, and under assumption 4 in the previous section, the arrival processes at these nodes are Poisson with intensity λ and independent of each other. Hence the combined arrival process is also Poisson and with intensity $(N-i)\lambda$. A connection request is accepted with probability $P_{jn}(i)$ and is rejected with probability $1-P_{jn}(i)$. Thus, the combined Poisson process is then randomly split into two random processes: one is the process of the arrivals that can be successfully realized in the network, and another is the process of the arrivals that are blocked. Since a split Poisson process is still Poisson, the arrival of a connection request that can join the i ongoing connections is Poisson with intensity $(N-i)\lambda P_{jn}(i)$, which is the transition rate from state i to state $i+1$. We should also consider the boundary conditions for $i=0$ and $i=N$. A complete state transition diagram is illustrated in Fig. 1. Clearly, this is a birth-death process and let $\pi_i, i = 0,1,\ldots,N$, be the stationary distribution. We have

$$\pi_{i+1}(i+1)\mu = \pi_i(N-i)\lambda P_{jn}(i),\ i \in \{0,1,\ldots,N-1\}$$

Let $\rho = \lambda/\mu$,

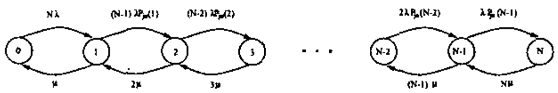

Fig. 1. State transition diagram of a k-fold multicast network.

$$\pi_i = \pi_0 \frac{N(N-1)\cdots(N-i+1)}{i!}\rho^i \prod_{j=0}^{i-1} P_{jn}(j) = \pi_0 \binom{N}{i} \rho^i P_{mc}(i), i \in \{0,1,\ldots,N\} \quad (4)$$

where π_0 is the probability that the network is in state 0 which can be determined by $\pi_0 = \frac{1}{\sum_{i=0}^{N} \binom{N}{i} \rho^i P_{mc}(i)}$.

IV. Throughput and Blocking Probability

Given the stationary distribution of network states, in this section we derive the throughput and blocking probability of a k-fold multicast network.

Consider a long time period $[0,T]$. First we have the following lemma.

Lemma 1: For a time period $[0,T]$, the average number of successful multicast connection requests carried by the network during $[0,T]$ is given by

$$N_{succ} = T\lambda \sum_{i=0}^{N} \pi_i (N-i) P_{jn}(i) \quad (5)$$

Proof. The average dwelling time of state i is the inversion of the rate that the network departs from state i:

$$\frac{1}{(N-i)\lambda P_{jn}(i) + i\mu}$$

It holds for all $i \in \{0,1,\ldots,N\}$. Since the time the system spent in state i during $[0,T]$ is $\pi_i T$, the average number of visits to state i during $[0,T]$ is $T\pi_i((N-i)\lambda P_{jn}(i) + i\mu)$. Notice that the number of visits is also the number of times the network departs from state i. A departure may be caused by the arrival of a successful connection request or the termination of an ongoing connection. The first case has the probability

$$\frac{(N-i)\lambda P_{jn}(i)}{(N-i)\lambda P_{jn}(i) + i\mu}$$

Thus, the average number of times the network departs from state i due to the arrival of a successful connection request is $T\pi_i(N-i)\lambda P_{jn}(i)$. This is also the average number of successful connection requests arrived at the network when the network is in state i ($i \in \{0,1,\ldots,N\}$) during $[0,T]$. Therefore, the average number of successful connection requests carried by the network during $[0,T]$ is obtained by summing over i: $N_{succ} = T\lambda \sum_{i=0}^{N} \pi_i (N-i) P_{jn}(i)$. ■

Proposition 1: The *throughput* of a k-fold multicast network is given by

$$TH = \lambda \sum_{i=0}^{N} \pi_i (N-i) P_{jn}(i) \quad (6)$$

which can be directly obtained from Lemma 1.

On the average, the total number of connection requests arrived at the network during $[0,T]$ is $N_{total} = N\lambda T$, among which only N_{succ} connection requests given in Lemma 1 are successful, and the rest

$$N_{bl} = N_{total} - N_{succ} = T\lambda \sum_{i=0}^{N} \pi_i (N(1-P_{jn}(i)) + iP_{jn}(i))$$

connection requests are blocked. The blocking probability is N_{bl}/N_{total}. Thus, we have

Proposition 2: The blocking probability of a k-fold multicast network is

$$PB = \frac{1}{N} \sum_{i=0}^{N} \pi_i (N(1-P_{jn}(i)) + iP_{jn}(i)) \quad (7)$$

To further study the performance of a k-fold multicast network, we consider input blocking and output blocking separately. Input blocking depends on the number of busy source nodes in the network. Since a larger fold means more busy source nodes, increasing k will generally increase the probability of input blocking. Input blocking can be reduced only by adding buffers at each source node, which is not considered in this paper. On the other hand, increasing k will reduce the chance of output blocking. In order to calculate the output blocking probability, first we have the following Lemma.

Lemma 2: Let $p_{bl}(i,m)$ be the probability that exactly m requests arrived at idle source nodes are blocked during a single visit to state i. We have

$$p_{bl}(i,m) = \alpha(1-\alpha)^m, \quad (8)$$

where $\alpha = \frac{(N-i)\lambda P_{jn}(i) + i\mu}{(N-i)\lambda + i\mu}$.

Proof. Define the following events: E_1: the first m connection requests arrive before any of the i ongoing connections terminates; E_2: none of the m connection requests can be realized along with the i ongoing connections; E_3: one of the ongoing connections terminates after the arrival of the m_{th} request and before the arrival of the $(m+1)_{th}$ request; E_4: the $(m+1)_{th}$ request arrives before any ongoing connection terminates and is $m.c.$ with the i ongoing connections. We have

$$p_{bl}(i,m) = P(E_1 \cap E_2) P(E_3 \cup E_4 | E_1 \cap E_2)$$

Events E_1 and E_2 are independent of each other because that knowing the arrival time of a connection request does not provide any information on whether it can join the i ongoing connections. Thus, $P(E_1 \cap E_2) = P(E_1)P(E_2)$. Since events E_3 and E_4 cannot occur at the same time,

$P(E_3 \cup E_4 | E_1 \cap E_2) = P(E_3 | E_1 \cap E_2) + P(E_4 | E_1 \cap E_2)$. Thus,

$$p_{bl}(i,m) = P(E_1)P(E_2)(P(E_3|E_1 \cap E_2) + P(E_4|E_1 \cap E_2))$$

Now let Y denote the time interval that the network first enters state i till one of the i ongoing connections terminates. Y is an exponentially distributed random variable with parameter $i\mu$. Let X_m denote the time interval that the network first enters state i till the arrival of the m_{th} connection request. Then X_m is the summation of m independent exponentially distributed random variables with parameter $(N-i)\lambda$. We can show that

$$P(E_1) = P(Y > X_m) = \left(\frac{(N-i)\lambda}{(N-i)\lambda + i\mu}\right)^m$$

Since connection requests are independent of each other, we have $P(E_2) = (1 - P_{jn}(i))^m$. Also,

$$\begin{aligned} P(E_3|E_1 \cap E_2) &= P(Y < Xm+1 | Y > Xm) \\ &= 1 - P(Y > Xm+1 | Y > Xm) = 1 - \frac{P(Y > Xm+1, Y > Xm)}{P(Y > Xm)} \\ &= 1 - \frac{P(Y > Xm+1)}{P(Y > Xm)} = \frac{i\mu}{(N-i)\lambda + i\mu} \end{aligned} \quad (9)$$

Similarly, we have

$$P(E_4|E_1 \cap E_2) = \frac{(N-i)\lambda}{(N-i)\lambda + i\mu} P_{jn}(i)$$

Let $\alpha = \frac{(N-i)\lambda P_{jn}(i)+i\mu}{(N-i)\lambda+i\mu}$. We have $p_{bl}(i,m) = \alpha(1-\alpha)^m$ ■

From Lemma 2 we know that the number of the requests blocked due to output blocking during staying in state i is a geometrically distributed random variable with parameter α. It has mean:

$$\frac{1-\alpha}{\alpha} = \frac{(N-i)\lambda(1-P_{jn}(i))}{(N-i)\lambda P_{jn}(i) + i\mu} \tag{10}$$

From the proof of Lemma 1, we know that the average number of visits to state i during $[0,T]$ is $T\pi_i((N-i)\lambda P_{jn}(i)+i\mu)$. Combining this fact and (10), we have that the average total number of requests blocked due to output blocking when network is in state i during $[0,T]$ is $T\pi_i(N-i)\lambda(1-P_{jn}(i))$. Then the average number of requests blocked by output blocking during $[0,T]$ is

$$N_{outbl} = T\lambda \sum_{i=0}^{N} \pi_i(N-i)(1-P_{jn}(i))$$

Thus, the average number of requests blocked due to input blocking during $[0,T]$ is the average number of the requests blocked minus N_{outbl}

$$N_{inbl} = N_{bl} - N_{outbl} = T\lambda \sum_{i=0}^{N} \pi_i i$$

We have

Proposition 3: The output blocking ratio which is defined as the ratio of blocked requests due to output blocking over all blocked requests is

$$\frac{\sum_{i=0}^{N} \pi_i(N-i)(1-P_{jn}(i))}{\sum_{i=0}^{N} \pi_i(N(1-P_{jn}(i)) + iP_{jn}(i))} \tag{11}$$

V. Some Generalizations

In this section, we discuss some possible generalizations of our results obtained in previous sections.

First, although we have mainly focused on an $N \times N$ k-fold multicast network, the results can be easily extended to an asymmetrical k-fold multicast network with N source nodes and M destination nodes. The only modification we need to make is equation (2). For an $N \times M$ k-fold multicast network, the probability that i multicast connection requests are $m.c.$ is

$$P_{mc}(i) = \begin{cases} \left(\sum_{m=0}^{k} p_{deg}(i,m)\right)^M & i > k \\ 1 & \text{otherwise} \end{cases} \tag{12}$$

All other results and discussions are still valid.

Secondly, if the fanout of a multicast connection follows other types of distributions other than binomial, such as geometric distribution or arbitrary distribution used in [9], we also only need to recalculate $P_{mc}(i)$. However, in this case, the degrees of destination nodes will no longer be independent of each other. When the network size is large, finding $P_{mc}(i)$ analytically becomes impractical. Thus we have to generate $P_{mc}(i)$ by simulations. After obtaining $P_{mc}(i)$, the throughput and blocking probability can be obtained immediately from our analytical model.

VI. Simulations and Observations

We have conducted simulations for network sizes 16 and 32 under different fanouts and arrival rates to validate our analytical results. Due to limited space, in Fig. 2, we illustrate only the results for network size 32. In the simulation, we use F to represent the fanout of a multicast connection request. F is a Binomial random variable with parameter (N,θ) and mean $E(F) = N\theta$. Without loss of generality, we let service rate $\mu = 1$.

In Fig. 2(a) and (b) we plot network throughput as a function of network fold k. In Fig. 2(a), we keep arrival rates at 0.5 and vary the fanout distribution. First, let's look at the growth of the throughput with respect to k for different values of mean fanout $E(F)$. We observe that when $E(F)$ is relatively small, the throughput grows almost exponentially, but when $E(F)$ becomes larger, throughput grows much slower. Next, we can see that for any $E(F)$, the throughput grows more rapidly when k is small, and tends to converge to some value when k further increases. For example, in Fig. 2(a) for the average fanout $E(F) = 3.2$, the throughput increases by 75% when network fold k increases from 1 to 2, but when $k > 5$, it stops increasing and remains at about 10.6. This is because when k becomes very large, almost every connection request arrived at an idle source node can go through. Thus, the network throughput is totally determined by the arrival rate. In this case, further increasing k will not increase the throughput.

In addition, the value of k to achieve the maximum throughput depends on the fanout distribution. In Fig. 2(a),we can see that to achieve the maximum throughput, we need to let $k = 5$ for $E(F) = 3.2$, $k = 8$ for $E(F) = 6.4$, $k = 13$ for $E(F) = 16$ and $k = 16$ for $E(F) = 28.8$, respectively. We can see that the increase is not linear to $E(F)$ because the larger the $E(F)$, the less k we need to add to the network to achieve the maximum throughput.

In Fig. 2(b), we fix the fanout distribution and study network performance under different arrival rates. Similar observations can be drawn for $\lambda < 1$. We also did experiments on $\lambda > 1$ which means arriving is faster than serving. In most networks this would lead to an unstable state and is not considered. However, since we assume no buffers at source nodes and any connection request arrived at a busy source node is immediately dropped, it is still valid in our case. As we can see, a larger λ leads to larger throughput, but needs a larger k to reach the maximum throughput.

In Fig. 2(c) and (d), we plot the network blocking probability as a function of network fold k. In Fig. 2(c),we keep arrival rates at 0.5 and vary the fanout distribution. First, we notice that when the average fanout $E(F)$ is relatively small, blocking probability decreases very fast when k increases. For example, in Fig. 2(c) where $N = 32$, for $E(F) = 3.2$, blocking probability drops by 30% when k increases from 1 to 2. However, when $E(F)$ becomes larger, blocking probability decreases much slower and almost linearly to k. Next, we can see that blocking probability does not reach 0 when k further

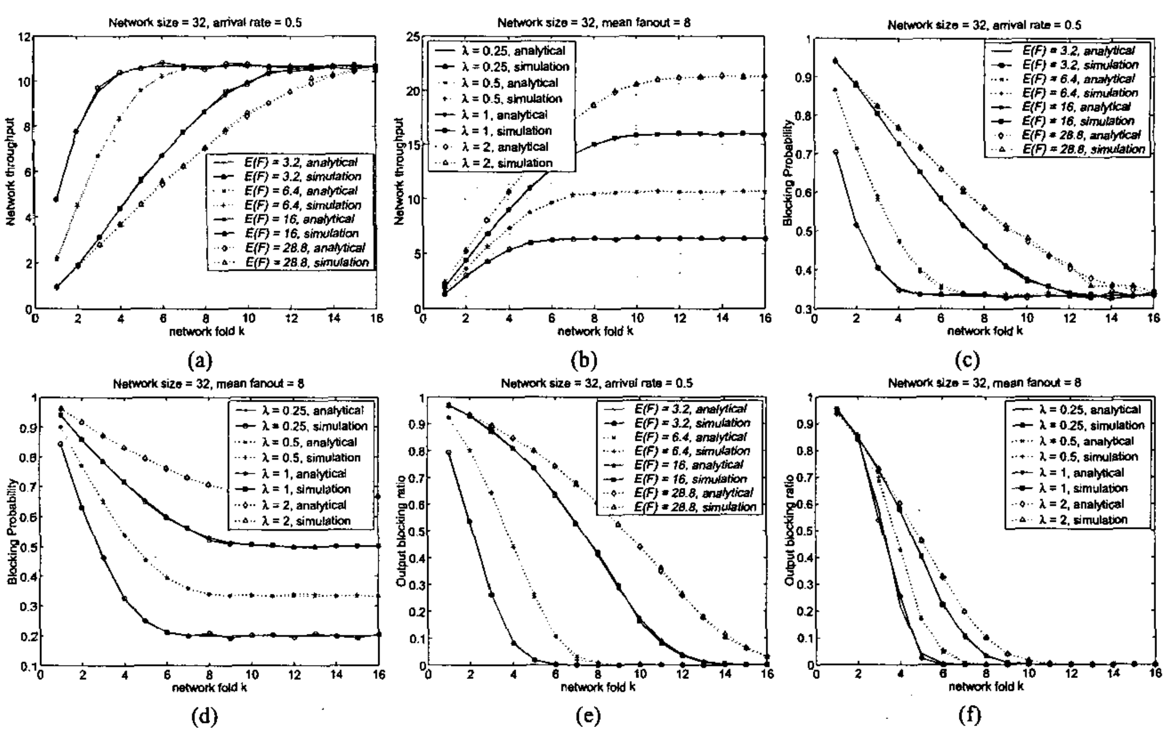

Fig. 2. Comparison of simulations and analytical results for a k-fold multicast network. (a) Throughput under different fanout distributions. (b) Throughput under different arrival rates. (c) Blocking probability under different fanout distributions. (d) Blocking probability under different arrival rates. (e) Output blocking ratio under different fanout distributions. (f) Output blocking ratio under different arrival rates.

increases. This is because of the presence of input blocking, which actually increases with k. In Fig. 2(d), we fix the fanout distribution and study network performance under different arrival rates. Again, we can see that blocking probability decreases quickly when arrival rates are small (say, less than 0.5) and much slower when arrival rates become large.

In Fig. 2(e) and (f), we plot the output blocking ratio as a function of network fold k. We can see that when k is larger than a certain value the output blocking ratio tends to 0, which means no connection request is blocked due to output blocking. As in Fig. 2(e) where $N = 32$ and $\lambda = 0.5$, to make output blocking ratio almost 0, we need only to let $k = 6$ for $E(F) = 3.2$, $k = 9$ for $E(F) = 6.4$ and $k = 14$ for $E(F) = 16$, respectively. Again, the value of k to eliminate output blocking is not a linear function of $E(F)$. The larger the $E(F)$, the less fold we need to add to the network to achieve zero output blocking ratio.

VII. Conclusions

A k-fold multicast network was recently proposed [7] as a cost-effective solution to provide better QoS functions in supporting multicast communication. To give a quantitative basis for network designers to determine the suitable value of system parameter k under different traffic loads, in this paper we have presented an analytical model for the performance of k-fold multicast networks under Poisson traffic. We first gave the stationary distribution of network states and then derived the throughput and blocking probability of the network. We have also conducted simulations to validate the analytical model and the results show that the analytical model is very accurate under the assumptions we made. From the analytical and simulation results we can see that by increasing the fold of the network, network throughput increases almost exponentially when the fanouts of multicast connections are relatively small compared to the network size. Our future work includes generalizing the model to other traffic distributions and other types of multicast networks.

References

[1] F.K. Hwang and A. Jajszczyk, "On nonblocking multiconnection networks," *IEEE Trans. Comm.*, vol. 34, pp. 1038-1041, 1986.

[2] Y. Yang and G.M. Masson, "Nonblocking broadcast switching networks," *IEEE Trans. Computers*, vol. 40, no. 9, pp. 1005-1015, 1991.

[3] C. Lee and A.Y. Oruç, "Design of efficient and easily routable generalized connectors," *IEEE Trans. Comm.*, vol. 43, pp. 646-650, 1995.

[4] Y. Yang and J. Wang, "A new self-routing multicast network," *IEEE Trans. Parallel and Distributed Systems*, vol. 10, no. 12, pp. 1299-1316, 1999.

[5] E. W. Zegura, "Evaluating blocking probability in generalized connectors," *IEEE/ACM Trans. Networking*, vol. 3, no. 4, pp. 387-398, 1995.

[6] Y. Yang and J. Wang, "On blocking probability of multicast networks," *IEEE Trans. Comm.*, vol. 46, no. 7, pp. 957-968, 1998.

[7] Y. Yang and J. Wang, "Nonblocking k-fold multicast networks," submitted to *IEEE Trans. Parallel and Distributed Systems*.

[8] G. Grimment and D. Stirzaker, *Probability and Random Processes*, 3rd Edition, Oxford University Press, 2001.

[9] A. Mokhtar and M. Azizoglu, "Performance analysis of a shared-medium ATM switch with multicast traffic," *IEEE GLOBECOM 1999*, pp. 1385-1390, 1999.