

Routing Strategies for Maximizing Throughput in LEO Satellite Networks

Jun Sun and Eytan Modiano, *Senior Member, IEEE*

Abstract—This paper develops routing and scheduling algorithms for packet transmission in a low earth orbit satellite network with a limited number of transmitters and buffer space. We consider a packet switching satellite network, where time is slotted and the transmission time of each packet is fixed and equal to one time slot. Packets arrive at each satellite independently with a some probability during each time slot; their destination satellite is uniformly distributed. With a limited number of transmitters and buffer space on-board each satellite, contention for transmission inevitably occurs as multiple packets arrive at a satellite. First, we establish the stability region of the system in terms of the maximum admissible packet arrival rate that can possibly be supported. We then consider three transmission scheduling schemes for resolving these contentions: *random packet win*, where the winning packet is chosen at random; *oldest packet win*, where the packet that has traveled the longest distance wins the contention; and *shortest hops win (SHW)*, where the packet closest to its destination wins the contention. We evaluate the performance of each of the schemes in terms of throughput. For a system without a buffer, the SHW scheme attains the highest throughput. However, when even limited buffer space is available, all three schemes achieve about the same throughput performance. Moreover, even with a buffer size of just a few packets the achieved throughput is close to that of the infinite buffer case.

Index Terms—Low earth orbit (LEO) satellite, mesh, routing, throughput.

I. INTRODUCTION

IN ORDER TO efficiently support bursty packet data traffic, future satellite systems may be designed using packet switching techniques. In this paper, we consider a low earth orbit (LEO) satellite network with packet transmission capabilities. Each satellite is equipped with a limited number of transmitters for communicating with its neighboring satellites and some limited buffer space for storing packets while they await their transmission. Due to the random nature of packet traffic, contention for transmission inevitably occurs as multiple packets arrive at a satellite. How to best resolve these contentions in order to achieve high throughput is the focus of this paper. Specifically, we consider three scheduling schemes for node-to-node communication in satellite networks and we analyze their throughput under a stochastic traffic environment.

We model the satellite network as an $N \times N$ mesh (shown in Fig. 1), where each satellite has k transmitters and m receivers.

Manuscript received December 16, 2002; revised April 17, 2003. This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) under the Next-Generation Internet Initiative.

The authors are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139-4307 USA (e-mail: junsun@mit.edu; modiano@mit.edu).

Digital Object Identifier 10.1109/JSAC.2003.819975

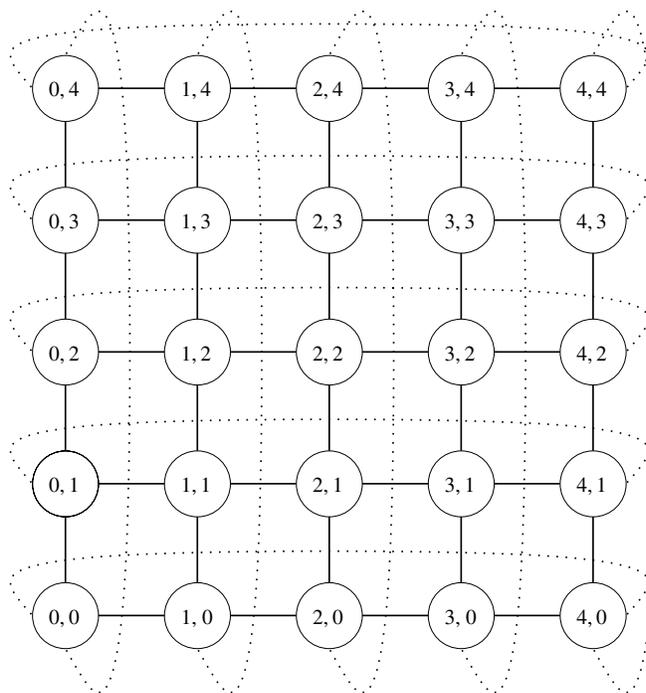


Fig. 1. A 5×5 mesh.

An $N \times N$ mesh is a two-dimensional (2-D) N -ary hypercube. This topology is similar in structure to that used in the Iridium satellite systems [3]–[5] and differs from a binary hypercube in that each node has a constant number of neighbors (4), regardless of N . Of course, satellite constellations, such as Iridium may not form a symmetric $N \times N$ mesh topology. Here, we focus on the symmetric case for simplicity, but our results can be very easily applied to a nonsymmetric topology. For the remainder of the paper, we will refer to this topology simply as a mesh. We focus on the case of $k = 1$ and $m = 4$ (i.e., the satellite can transmit to only one of its neighbors and receive from all of its neighbors simultaneously). A limited number of transmitters may be used in order to reduce the size and cost of the satellite payload and reduce the processing requirements on the satellite. For example, a single solid-state amplifier can be shared among the cross-links and be used to activate one cross-link at a time. Again, depending on the system design, the number of transmitters and receivers may be different than the case considered in this paper. However, the analysis presented here can be easily applied to any other number of receivers and transmitters (e.g., $k = 4$ and $m = 1$, etc.). In this paper, we focus only on the problem of intersatellite communication. That is, we are only concerned with the routing of packets from a source satellite to a destination satellite. The communication of data from

the satellite to the ground and from the ground to the satellite is a topic that has received much attention in the past and is not considered in this work.

The network operation is similar to one discussed in [12]. That is, the nodes operate synchronously: the time axis is divided into slots and each node can relay one packet per time slot. A *new* packet is generated independently at each node locally with probability p_0 during each time slot. Thus, the arrival process of new packets is modeled as a Bernoulli process with rate p_0 packets per time slot. Each packet then follows a fixed shortest path to its destination node. At the end of a slot there are up to four *continuing* packets (received from neighboring satellites during that time slot) at each satellite node. As more than one continuing packets arrive at a particular node, contention for transmission in the next time slot will occur. Note that even if we used as many as four transmitters at each node, contention would still occur, as multiple packets may need to be routed along the same satellite cross-link. Hence, we need to develop a transmission scheduling scheme for resolving these conflicts. After a packet arrives at its destination node at the end of a time slot, it is removed from the system.

Routing schemes for resolving packets' contention in a regular topology have been investigated by numerous researchers. In [10], Greenberg and Hajek provided an approximate analysis of the transient and steady-state behavior of deflection routing in hypercube network. Stamoulis and Tsitsiklis [11] studied the efficiency of greedy routing in hypercube network. In [12], the authors propose two different hypercube routing schemes and evaluate the throughput of both the buffered and the unbuffered version of these schemes. Their results are also approximate. In all the aforementioned papers, the topology that they used is hypercube.

In this paper, we propose several scheduling schemes and compare, for each scheme, the average throughput of the network when it reaches steady-state. Specifically, we study the throughput of *random packet win* (RPW) scheme, where the winning packet is chosen at random; *oldest packet win* scheme (OPW), where the packet that has traveled the longest distance toward its destination receives priority; and *shortest hop win* (SHW) scheme, where the packet that has the shortest distance to its destination receives priority over other continuing packets. Both our analytic and simulated results show that, in the case of no buffer at each node, SHW scheme attains the best throughput performance, OPW scheme the second, and RPW the worst. These results can be intuitively explained by noting that the SHW scheme gives preference to packets that are about to exit the system and, hence, leads to improved system throughput. Similarly, the OPW scheme gives preference to packets that have traveled a longer distance and, hence, may be closer to their destination than other packets. Surprisingly, we found that when there is a buffer at each node, the performance of the three schemes are very similar in terms of throughput. Moreover, we found that even a small buffer size can achieve throughput close to that of an infinite buffer size. Finally, our schemes give priority to continuing packets over new packets; hence, packets are typically not dropped inside the network, but are prevented from entering the network at their origin. When packets are dropped inside the network, due to contention or buffer overflow it is as-

sumed that those packets will be retransmitted by higher layer protocols (for example, by a link layer protocol that is designed to operate over the satellite system). The operation of such retransmission protocols is a well studied topic that is outside the scope of this paper. Instead, we assume that the packets arrive at each node with a rate p_0 , that includes both new packets and retransmissions of old packets. Such an assumption is often used in the analysis of multiple-access protocols [14].

In Section II, we derive the stability region of the system. That is the region of arrival rates for which the satellite systems can be used to deliver traffic. In Section III, we analyze the throughput of the various transmission schemes and compare their performance with and without buffers. In Section IV, we summarize the paper.

II. STABILITY ANALYSIS

We consider now an $N \times N$ mesh with N^2 nodes, each of which generates packets independently according to a random process of rate λ packets per unit time to be sent to a random and uniformly chosen destination node. The packet takes exactly one unit of time to be transmitted. Each node can only transmit to one of its neighbors during a given slot, but can receive packets from all neighbors simultaneously. We first derive a necessary condition for stability.

Theorem 1: A necessary condition for the system to be stable is

$$\lambda \cdot E[d] < 1 \quad (1)$$

where $E[d]$ is the expected number of hops from source node to destination node.

Proof: The total number of new packets generated in the network per unit time is λm , where $m = N^2$. During each time unit, an average total demand of $\lambda \cdot m \cdot E[d]$ packet transmissions are generated in the system, where $E[d]$ is the expected number of hops from source node to destination node. Since at most m transmissions may take place per unit of time, we have $\lambda m \cdot E[d] < m$, or $\lambda \cdot E[d] < 1$. ■

Next, we will show that the network is stable if $\lambda \cdot E[d] < 1$ by employing a fixed shortest path routing scheme. That is, all packets that are generated at node i with destination j go through the same shortest path to reach their destination nodes (for example see [8]). We first present several lemmas that will be useful. The following lemma is from [1].

Lemma 1: Consider any start node x and let $n_x(i)$ be the number of nodes exactly i hops away from node x . For N even, we have

$$n_x(i) = \begin{cases} 1, & i = 0 \\ 4i, & 0 < i < \frac{N}{2} \\ 4i - 2, & i = \frac{N}{2} \\ 4(N - i), & \frac{N}{2} < i < N \\ 1, & i = N \end{cases} .$$

Proof: See [1]. ■

Next, consider a scenario in which every node of the network sends out one unit of traffic to every other node (also known as *complete exchange* or *all-to-all communication* [2]) by using a fixed shortest path routing algorithm. Each source-destination pair uniquely defines one class of traffic. The load of a particular

link is defined to be the number of different classes of traffic that pass through that link. We are interested in the average load of a link under all-to-all traffic.

For an $N \times N$ mesh, the total number of nodes in the network is N^2 ; the total number of unordered node pairs is $(N(N-1))/2$; and the total number of links is $2N^2$. The following lemma gives us the average load of a link under all-to-all traffic.

Lemma 2: For an $N \times N$ mesh under all-to-all traffic, the average load of a link is $(1/4)(N^3 - N)$ for N odd, and $(1/4)N^3$ for N even by using a fixed shortest path routing algorithm.

Proof: We first consider the case where N is odd. From Lemma 1, we see that there are a total of $(1/2)(4i)N^2$ unordered pairs that are i hops away from each other for $0 < i \leq ((N-1)/2)$, and a total of $(1/2)(4(N-i))N^2$ unordered pair for $(N-1)/2 < i \leq N-1$. Here, the maximum shortest hop length between two nodes in the network is $N-1$.

Let D denote the average path length between two nodes. We then have

$$D = \frac{\sum_{i=1}^{\frac{N-1}{2}} i(2iN^2) + \sum_{i=\frac{N-1}{2}+1}^{N-1} i(2(N-i)N^2)}{\left(\frac{N(N-1)}{2}\right)} = \frac{N^3 - N}{2(N^2 - 1)} = \frac{N}{2}.$$

The total traffic in this network is $N^2(N^2 - 1)(N/2) = (1/2)(N^5 - N^3)$. Thus, since all links have the same load due to the symmetry of the network and the fixed shortest path routing, the average load on a link is the total traffic divided by the number of links $(1/2)(N^5 - N^3)/2N^2 = (1/4)(N^3 - N)$.

The case for N even can be shown similarly. ■

Now, assuming there is a separate buffer for each class of traffic that is going to be served at a node, we consider a round-robin service policy whereby the transmitter serves each queue with an equal amount of time. In the case of an empty queue, the transmitter will be idle for a period of time that is allocated to that queue. Then, we have the following sufficient condition for stability.

Theorem 2: With packet's arrival rate λ and the destination node uniformly chosen, the $N \times N$ mesh network is stable for all $\lambda < (1/(E[d]))$, where $E[d] = N/2$ is the expected path length between two nodes, under a fixed shortest path routing scheme and using the round-robin service policy.

Proof: Consider an $N \times N$ mesh for N odd. There is a total of $N^2(N^2 - 1)$ classes of traffic in this queuing network, each corresponding to a unique source-destination pair (i, j) . For an arbitrary node k in the network, since the packets are arriving at an external rate of λ and destinations are uniformly chosen, packets of class (k, j) arrive at the rate of $(\lambda/(N^2 - 1))$ for all nodes $j \neq k$. Due to the use of the fixed shortest path routing scheme, we know exactly how many classes of traffic need to be served at node k . Specifically, from Lemma 2, each of the four links connecting node k has $(1/2) \cdot (1/4)(N^3 - N)$ classes of traffic that need to go to or through node k (the term $1/2$ is there because the links are bidirectional and we only consider the traffic in one direction *coming into* node k). More precisely, for each link connecting node k , $(1/4)(N^2 - 1)$ classes of traffic need to *go to* node k , while $(1/2) \cdot (1/4)(N^3 - N) - (1/4)(N^2 - 1)$ classes of traffic need to *go through* node k . We

call these classes of traffic internal arrivals. That is, a total of $(1/2) \cdot (1/4)(N^3 - N) - (1/4)(N^2 - 1)$ internal arrivals must be transmitted through node k . In addition, the $N^2 - 1$ classes of traffic that are generated locally at node k are called external arrivals. Since the round-robin service policy serves each class in a round robin fashion, a constant fraction of service time is allocated to each class of traffic. Under the round-robin policy, node k can be viewed as having many dedicated servers (one for each class of traffic) with identical service rate. Hence, all queues at node k are independent, and they are stable as long as the service rate is greater than the arrival rate for each class of traffic. A particular class of traffic typically will travel through several nodes to reach its destination. If all nodes on its path to the destination are serving this class of traffic at a rate greater than the arrival rate, the series of queues are also stable ([15, Th. 7.4.12]). In order to complete the proof of the theorem, we next derive the traffic rate that travel through node k . First, the internal arrivals from all four of node k 's neighbors have a total rate of

$$\begin{aligned} & 4 \cdot \left[\frac{1}{8}(N^3 - N) - \frac{1}{4}(N^2 - 1) \right] \\ & \cdot (\text{arrival rate of a single class}) \\ & = 4 \cdot \left[\frac{1}{8}(N^3 - N) - \frac{1}{4}(N^2 - 1) \right] \cdot \left[\frac{\lambda}{N^2 - 1} \right] \\ & = \frac{N}{2}\lambda - \lambda. \end{aligned}$$

In addition, there are $N^2 - 1$ external arrivals at node k , so the external arrival rate at node k is equal to

$$(N^2 - 1) \cdot (\text{arrival rate of a single class}) = \lambda.$$

Therefore, the total arrival rate to node k (the sum of the external and internal arrival rates) is

$$\left(\frac{N}{2} \cdot \lambda - \lambda \right) + \lambda = \frac{N}{2} \cdot \lambda.$$

Now, since the total service rate is equal to 1, in order for the queue to be stable, we must have

$$\lambda \cdot \frac{N}{2} < 1 \Rightarrow \lambda < \frac{1}{E[d]}.$$

■

Note that in the above, we used the fact that under shortest path routing $E[d] = N/2$. Moreover, for routing schemes that choose a random shortest path between the source and destination node, it can be similarly shown that the stability region is still $\lambda < 1/(E[d])$.

III. ANALYSIS AND SIMULATION OF THROUGHPUT

In this section, we present the main results of this paper. Several scheduling schemes to resolve contention for transmission are discussed. Detailed theoretical analysis and simulation results of throughput are provided. First, we give a general overview of these transmission schemes which will be analyzed in the later sections. We assume that, at each node, there is buffer which can hold up to m packets, in addition to the packet under transmission. Because only one transmitter is available at each node, conflicts result from simultaneous arrivals of more than

one packet from the neighboring nodes or a new packet generated in the current node. Contention may be resolved by assigning different priority to the incoming packets (both the continuing packets and the new packet). Below, we describe several schemes to resolve the contention. In all schemes, packets follow fixed shortest paths to their destination nodes.

- 1) *Random Packet Win (RPW)*: If more than one continuing packets arrive at a node, RPW randomly chooses the one to be transmitted in the next time slot. The other packets are stored in the buffer if there is space available. When the buffer space cannot accommodate all of the continuing packets that need to be stored in the buffer, RPW randomly selects packets among these continuing packets to fill up the buffer (the other packets are dropped). In case of no continuing packet arriving, RPW picks the head of buffer packet to be transmitted in the next time slot. If the buffer is empty, RPW sends the newly generated packet (if there is one) in the next time slot. In case of contention, the new packet is discarded.
- 2) *Oldest Packet Win (OPW)*: If more than one continuing packets arrive at a node, OPW chooses the one that has traveled the most hops toward its destination to be transmitted in the next time slot. The other packets are stored in the buffer if there is space available. When the buffer space cannot accommodate all of the continuing packets that need to be stored in the buffer, OPW randomly selects packets among these continuing packets to fill up the buffer (the other packets are dropped). In case of no continuing packet arriving, OPW selects the head of buffer packet to be transmitted in the next time slot. If the buffer is empty, OPW transmits the newly generated packet (if there is one) in the next time slot. In case of contention, the new packet is discarded. The OPW scheme attempts to give priority to packets that have traveled a longer distance toward their destination and, hence, reduce the waste in capacity that may result from dropping packets that have already traveled a long way.
- 3) *Shortest Hop Win (SHW)*: If more than one continuing packets arrive at a node, SHW chooses the one with the shortest hop distance to its destination node to be transmitted in the next time slot. The other packets are stored in the buffer if there is space available. When the buffer space cannot accommodate all of the continuing packets that need to be stored in the buffer, SHW randomly picks packets among these continuing packets to fill up the buffer (the other packets are dropped). In case of no continuing packet arriving, SHW picks the head of buffer packet to be transmitted in the next time slot. If the buffer is empty, SHW sends the newly generated packet (if there is one) in the next time slot. In case of contention, the new packet is discarded. The SHW scheme attempts to give priority to packets that are closer to their destination and, hence, reduce the load in the system by getting packets out of the system as soon as possible.

We will give a detailed analysis of the throughput of SHW scheme and OPW scheme in the subsequent sections. The analysis of RPW scheme is similar to the SHW scheme and, therefore, omitted for brevity.

We also introduce the following notation which will be useful in the later sections. For an arbitrary packet \mathcal{P} , let $s_{\mathcal{P}}$ denote its source node; $t_{\mathcal{P}}$ denote its destination node; and $d_H(s_{\mathcal{P}}, t_{\mathcal{P}})$ denote the shortest hop distance between $s_{\mathcal{P}}$ and $t_{\mathcal{P}}$. We also let d denote the maximum shortest hop distance between a source and destination pair. Note that $d = N - 1$ for the $N \times N$ mesh.

A. Throughput Analysis for SHW Scheme With Buffer

In Theorems 1 and 2, the capacity region of the $N \times N$ mesh network was established. However, the throughput of any particular routing scheme may not necessarily achieve this capacity. We wish to compare the above routing schemes in terms of throughput performance. Analyzing throughput performance in a network with random traffic is nontrivial because the arrivals of packets on different links to a particular node are not independent. As a result, the network forms a system of interacting queues, whose exact analysis involves the solution of an n -dimensional Markov chain (where n is the number of nodes in the network and equals N^2 for the $N \times N$ mesh) [1]. In order to reduce the dimensionality of the system, we make an independence approximations to the system. Such approximations are often used to analyze network performance [1], [10], [14]. In particular, under our uniform traffic and random destination assumption, it is reasonable to expect that the independence approximation is well justified. In fact, we will show later through simulation that the analysis using the independence approximation performs extremely well when compared with simulations. We make two approximating assumptions that are similar to the assumptions made in [10] to analyze deflection routing in hypercubes. First, we assume that packet arrivals on each different incoming link to a particular node are independent during a time slot. Second, we assume that the arrivals of packets to a node in one slot are independent of the arrivals to the node during previous slots.

At the beginning of a time slot, the transmitter at an arbitrary node, say node a , sends a packet \mathcal{P} to one of its neighbors, say node k . Before the start of the transmission at node a , if the packet \mathcal{P} is i hops away from its destination node, we say the packet is of type i ; more precisely, $d_H(a, t_{\mathcal{P}}) = i$. When the packet \mathcal{P} arrives at node k , it competes with other arriving packets for transmission during the next time slot. A packet is said to be the winning packet if it will be transmitted in the next time slot. The SHW scheme selects amongst continuing packets at node k the one with the shortest hop distance to its destination to be the winning packet. If there are $j, j > 1$, packets having the same shortest hop distance to their respective destination nodes, SHW randomly selects one packet to be the winning packet among these j packets. If \mathcal{P} has the shortest hop distance to its destination node among the continuing packets at node k , it is said to be a winning packet of type $(i - 1)$ at node k . If no continuing packet arrives at node k during a time slot, the winning packet is the head of buffer packet if the buffer is nonempty. Similarly, the newly generated packet is the winning packet if there are no continuing packets and buffered packets at node k . Once the packet arrived its destination node, it departs the system.

In the steady-state, due to the same external arrival rate p_0 and uniform destination for each newly generated packet, by

symmetry each node has the same statistics (i.e., the probability that a winning packet is of type i , $1 \leq i \leq d$, is the same for all nodes) without the approximation assumption. However, to get the exact value of these statistics, we have to utilize the two approximations made above. Specifically, by considering only one node k in the network, let A_i , $1 \leq i \leq d$, denote the event that node k has a winning packet of type i . Similarly, let E denote the event that node k is idle, which means no packet in the buffer, no continuing packet, and no new packet generated at node k . We can then write the probabilities $P(A_i)$'s and $P(E)$ recursively, in terms of the same probabilities at neighboring nodes, by using the property that each node has the same statistics and by considering the interactions between node k and its neighboring nodes. Throughout this section, we focus on finding $P(A_i)$'s. The throughput is thus obtained as $P(A_1)$ in the SHW scheme.

Again, considering an arbitrary node k , we define the following.

- B_i , $0 \leq i \leq 4$, to be the event that node k received packets, with their destination nodes other than node k , from i out of the four neighboring nodes.
- H_i , $1 \leq i \leq d$, to be the event that the head of the buffer packet is of type i .
- U_i , $1 \leq i \leq d$, to be the event that a new packet that is i hops away from the destination node is generated at node k .
- BE to be the event that the buffer at node k is empty.
- BE^c to be the event that the buffer at node k is nonempty.

With the relevant events defined, we now write the equations for $P(A_i)$ in terms of these events. For $1 \leq i \leq d-1$, we have

$$\begin{aligned} \gamma(i) = P(A_i) &= P(A_i | B_1)P(B_1) + P(A_i | B_2)P(B_2) \\ &+ P(A_i | B_3)P(B_3) + P(A_i | B_4)P(B_4) \\ &+ P(H_i)P(BE^c)P(B_0) \\ &+ P(U_i)P(BE)P(B_0) \end{aligned} \quad (2)$$

and for $i = d$

$$\gamma(d) = P(A_d) = P(U_d)P(BE)P(B_0).$$

To derive the above equations, consider the events that take place at node k . Since we give the first priority to the continuing packets, next priority to the buffered packet, and the lowest priority to the new packet, event A_i occurs if and only if one of the following events occur.

- A continuing packet of type i arrives at node k and wins the contention.
- The head of buffer packet is of type i and no continuing packet arrives.
- A new packet of type i is generated at node k , no continuing packet arrives, and the buffer is empty.

Equation (2) enumerates all of the above events. For event A_d , it occurs only when a new packet of type d is generated; there are no continuing packets; and no buffered packet (any continuing packet cannot be type d packet since d is the maximum hop distance between any source and destination node). Now, we write the individual terms out. The probability that a new packet with i hops to its destination is generated is the following:

$$P(U_i) = \frac{n_x(i)}{N^2 - 1} \cdot p_0$$

where $n_x(i)$ denotes the number of nodes that are i hops away (see Lemma 1) and $N^2 - 1$ is the total number of possible destination node. We also get for $0 \leq n \leq 4$

$$\begin{aligned} \beta_n &= P(B_n) \\ &= \binom{4}{n} \left[\frac{1}{4} \sum_{j=2}^d \gamma(j) \right]^n \left[1 - \frac{1}{4} \sum_{j=2}^d \gamma(j) \right]^{4-n}. \end{aligned} \quad (3)$$

The term $[(1/4) \sum_{j=2}^d \gamma(j)]$ denotes the probability that a neighboring node of k sends a continuing packet to node k . Similarly, $(\gamma(i+1))/(\sum_{j=2}^d \gamma(j))$ is the probability that a node is sending a packet of type $i+1$ given that node is sending a packet; and $(\sum_{j=i+2}^d \gamma(j))/(\sum_{j=2}^d \gamma(j))$ is the probability that a node is sending a packet whose type is greater than $i+1$, given that node is sending a packet.

Then, letting $a_i = (\gamma(i+1))/(\sum_{j=2}^d \gamma(j))$ and $c_i = (\sum_{j=i+2}^d \gamma(j))/(\sum_{j=2}^d \gamma(j))$, we have for $1 \leq i \leq d-1$

$$P(A_i | B_1) = a_i \quad (4)$$

$$P(A_i | B_2) = a_i^2 + \binom{2}{1} a_i c_i \quad (5)$$

$$P(A_i | B_3) = a_i^3 + \binom{3}{2} a_i^2 c_i + \binom{3}{1} a_i c_i^2 \quad (6)$$

$$P(A_i | B_4) = a_i^4 + \binom{4}{3} a_i^3 c_i + \binom{4}{2} a_i^2 c_i^2 + \binom{4}{1} a_i c_i^3. \quad (7)$$

To interpret the above equations, consider (5) for $P(A_i | B_2)$. Recall that packet with shorter distance to its destination has priority. Given that exactly two packets arrived from two of the four neighboring nodes of node k , the event that the winning packet is i hops away from its destination, or type i packet, is the union of the following two disjoint events.

- At node k , both of these two arriving packets are type i packets [for example, the first term in (5), a_i^2].
- At node k , one of these two arriving packets is a type i packet and the other one is of type j , where $i+1 \leq j \leq d$ [for example, the second term in (5), $\binom{2}{1} a_i c_i$].

The rest of the above equations can be explained similarly.

Next, we will investigate the probability that a head of buffer packet is of type i . The event that a *head of buffer* packet is of type i is the same as the event that a *buffered* packet is of type i since each packet in the buffer will be transmitted sometime in the future by our routing algorithm. In other words, a buffered packet will surely become a head of buffer packet since it will not be dropped. Let G_{i+1} denote the event that an arbitrary packet, say \mathcal{P} , that is $i+1$ hops away from its destination node before the start of its transmission to node k , subsequently loses the contention with other packets at node k

$$\begin{aligned} P(H_i) &= P(\text{a packet in the buffer is of type } i) \\ &= \frac{P(\text{type } i \text{ packet gets sent to the buffer})}{P(\text{packet gets sent to buffer})} \\ &= \frac{\gamma(i+1)P(G_{i+1})}{\sum_{j=2}^d \gamma(j)P(G_j)}. \end{aligned} \quad (8)$$

Note also that a packet of type d (newly generated packet with d hops to its destination node) will never be stored in the

buffer by the priority rule. Packet \mathcal{P} , which just became a type i packet after reaching node k , may lose the contention if one of the following events occur.

- Event $E_0(i)$: Out of the three remaining neighboring nodes of node k , there is at least one of them which is sending a packet of type j , where $j < i + 1$, to node k .
- Event $E_1(i)$: Out of the three remaining neighboring nodes of node k , there is exactly one of them which is also sending a packet of type $i + 1$ to node k , while the others are either not sending a packet to node k or sending packets of type j ($j > i + 1$) to node k .
- Event $E_2(i)$: Out of the three remaining neighbor of node k , there are exactly two of them which are also sending packets of type $i + 1$ to node k , while the other neighboring node is either not sending a packet to node k or sending packet of type j , $j > i + 1$, to node k .
- Event $E_3(i)$: Out of the three remaining neighbor of node k , each one of them is sending a packet of type $i + 1$ to node k .

From the above description, it is clear that $P(E_0(1)) = 0$, because, by definition, a type 1 packet has arrived at its destination and departed the system. For $2 \leq i \leq d - 1$, we have

$$P(E_0(i)) = 1 - \left[1 - \frac{1}{4} \sum_{j=2}^i \gamma(j) \right]^3 \quad (9)$$

and for $1 \leq i \leq d - 1$

$$P(E_1(i)) = \binom{3}{1} \left[\frac{\gamma(i+1)}{4} \right] \left[1 - \frac{1}{4} \sum_{j=2}^{i+1} \gamma(j) \right]^2 \quad (10)$$

$$P(E_2(i)) = \binom{3}{2} \left[\frac{\gamma(i+1)}{4} \right]^2 \left[1 - \frac{1}{4} \sum_{j=2}^{i+1} \gamma(j) \right] \quad (11)$$

$$P(E_3(i)) = \left[\frac{\gamma(i+1)}{4} \right]^3. \quad (12)$$

When event $E_0(i)$ occurs, packet \mathcal{P} will be sent to the buffer with probability one (although it may be dropped due to buffer overflow). Likewise, when event $E_1(i)$, or $E_2(i)$, or $E_3(i)$ occurs, packet \mathcal{P} will be sent to the buffer with probability $1/2$, $2/3$, $3/4$, respectively. Now, $P(G_i)$ can be obtained as

$$P(G_i) = P(E_0(i)) + \frac{1}{2}P(E_1(i)) + \frac{2}{3}P(E_2(i)) + \frac{3}{4}P(E_3(i)).$$

To get $P(\text{BE})$, we denote by b_i , $i = 0, 1, \dots, m$, the probability that there are i packets at a node's buffer at the beginning of slot. Since there are four receivers at a node, at most three continuing packets may arrive at the buffer during a time slot. Fig. 2 is a finite-state Markov chain which describes the evolution of the number of packets in a buffer of size m . The state represents the number of packets in the buffer. Thus, we have

$$P(\text{BE}) = b_0 = b_0\beta_1 + (b_0 + b_1)\beta_0 \quad (13)$$

$$b_1 = b_0\beta_2 + b_1\beta_1 + b_2\beta_0 \quad (14)$$

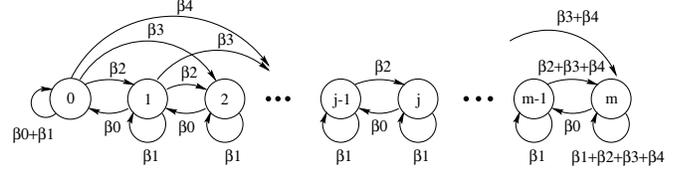


Fig. 2. Markov chain of the number of packets in a buffer of size m .

TABLE I
COMPARISON OF SIMULATION RESULT AND THEORETICAL RESULT FOR
11 × 11 MESH USING SHORTEST HOP WIN SCHEME WITH BUFFER

p_0	Theoretical	Simulation	99.99% Confidence
	Throughput	Throughput	Interval
0.1	0.0690	0.0690	$0.0690 \pm 1.003 \times 10^{-4}$
0.2	0.1052	0.1053	$0.1053 \pm 1.566 \times 10^{-4}$
0.3	0.1274	0.1274	$0.1274 \pm 1.446 \times 10^{-4}$
0.4	0.1423	0.1422	$0.1422 \pm 1.591 \times 10^{-4}$
0.5	0.1528	0.1527	$0.1527 \pm 1.678 \times 10^{-4}$
0.6	0.1607	0.1603	$0.1603 \pm 1.629 \times 10^{-4}$
0.7	0.1667	0.1663	$0.1663 \pm 1.736 \times 10^{-4}$
0.8	0.1715	0.1708	$0.1708 \pm 1.748 \times 10^{-4}$
0.9	0.1754	0.1747	$0.1747 \pm 1.713 \times 10^{-4}$
0.95	0.1771	0.1762	$0.1762 \pm 1.653 \times 10^{-4}$
0.99	0.1784	0.1773	$0.1773 \pm 1.701 \times 10^{-4}$

$$b_2 = b_0\beta_3 + b_1\beta_2 + b_2\beta_1 + b_3\beta_0 \quad (15)$$

$$b_j = b_{j-3}\beta_4 + b_{j-2}\beta_3 + b_{j-1}\beta_2 + b_j\beta_1 + b_{j+1}\beta_0 \quad (16)$$

$$b_m = b_m(\beta_1 + \beta_2 + \beta_3 + \beta_4) + b_{m-1}(\beta_2 + \beta_3 + \beta_4) + b_{m-2}(\beta_3 + \beta_4) + b_{m-3}\beta_4. \quad (17)$$

Recall that β_i is defined in (3).

With the above equations, we can solve for $\gamma(i)$ numerically. The throughput is obtained as $\gamma(1) = P(A_1)$. For our simulation, a 11×11 mesh with a buffer size of four at each node is used. As Table I shows, our numerical result is very accurate compare with the simulation result. The 99.99% confidence interval (within four standard deviations) is also shown in the table. Moreover, from Theorem 1, we know that the maximum achievable throughput is limited to $1/(E[d]) = 2/N = 2/11 = 0.182$. It is, therefore, interesting to note that the algorithm achieves throughput levels that are very close to the theoretical upper-bound. Finally, it is important to note that the value of p_0 represents the total offered load that includes both new arrivals, as well as retransmitted packets. In order for the system to be stable, the external arrival rate of new packets into the system must not exceed the throughput of the system, which in this case is limited to $2/11$.

As we mentioned earlier, our method can be generalized to asymmetrical satellite constellations (i.e., a $m \times n$ mesh

topology). Here, we briefly discuss how we can extend our analysis presented so far to the asymmetrical case. In the above analysis of SHW scheme, we calculate the probability that a particular node has a winning packet of type i , $\gamma(i)$, to get the throughput. Now, consider a winning packet of type i at node k . Since the topology is symmetrical, the packet will be transmitted to any of the four neighboring node of k with probability $1/4$. However, when the topology is asymmetric, the amount of traffic that goes through a vertical link is more than that goes through a horizontal link. For a winning packet of type i at node k , it is not clear how to get the probability that this packet will travel vertically or horizontally in the next hop. In fact, keeping track of the number of hops to destination is not enough for the asymmetrical topology. Instead, we need to know both the number of horizontal hops left and the number of vertical hops left to the destination for a packet. This way, once we specify a shortest path routing algorithm, the probability of a packet goes vertically or horizontally for the next hop can be obtained. Similar to the terms defined previously, a packet of type (i, j) indicates that packet has i horizontal hops left and j vertical hops left to its destination. To get the throughput, we need to get the probability that a particular node, say k , has a winning packet of type (i, j) , $\gamma(i, j)$ to be specific. The rest of the analysis is similar to the analysis of symmetrical case in the paper. That is, we still need to calculate $P(B_1)$ and $P(A_{ij} | B_1)$, where A_{ij} denotes the event that a particular node has a winning packet of type (i, j) .

B. Throughput Analysis for SHW Scheme Without Buffer

In this section, we consider the throughput of the SHW scheme without buffer at each node. The analysis is similar to the one in the previous section. The notation, if not specified, will be the same as the one defined previously. Again, we give priority to the continuing packet over the newly generated packet (i.e., the new packet can be transmitted only if no continuing packet arrives at that node). Thus, we have for $1 \leq i \leq d-1$

$$\begin{aligned} \gamma(i) = P(A_i) &= P(A_i | B_1)P(B_1) + P(A_i | B_2)P(B_2) \\ &+ P(A_i | B_3)P(B_3) + P(A_i | B_4)P(B_4) \\ &+ P(U_i)P(B_0) \end{aligned} \quad (18)$$

and for $i = d$

$$\gamma(d) = P(U_d)P(B_0). \quad (19)$$

$P(A_i | B_1), \dots, P(A_i | B_3)$ and $P(U_i)$ can be calculated by using the exact same formulas given in the previous section.

Fig. 3 is a plot of the throughput of a system with buffer and a system without buffer under SHW. When the load is low, the system with buffer outperforms the system without buffer. This can be explained by noting that packets can be put in the buffer temporarily if it lost the competition instead of just dropping them in the case of no buffer. Dropping a packet which has already traveled a certain number of hops wastes the previous transmissions of that packet, thus decreases the throughput of system. Intuitively, we would like to minimize the wasted work and hope that every transmission is going to contribute to the increase in throughput. As the load p_0 gets very high, more

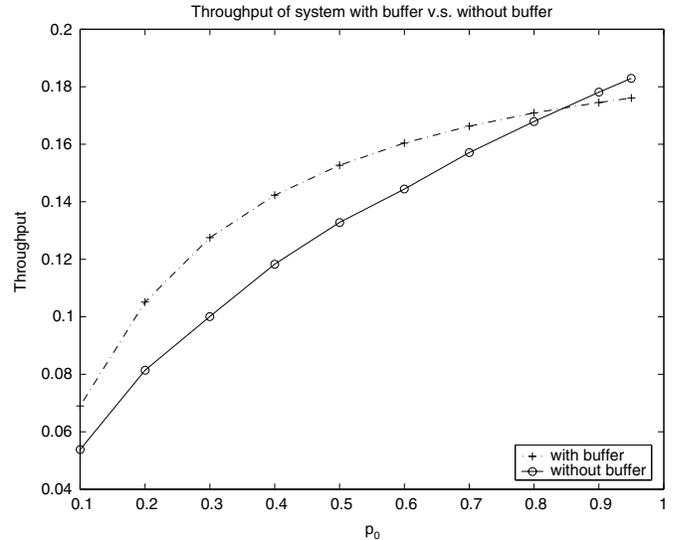


Fig. 3. Comparison of throughput of system with buffer size of 4 and without buffer using SHW.

packets will be in the system. As a result, the system without buffer can choose among these packets the ones with shorter hop distance to their destination nodes and drop the others with relatively longer hop distance to their destination, leading to an increase in throughput. For the system with buffer, even though it chooses the packet with shorter hop distance to the destination, it also stores packets with relatively longer hop distance to the destination in the buffer. In essence, for high p_0 , the system with buffer does not bias toward packets with shorter distance to their destination nodes as much as the system without buffer does.

C. Throughput Analysis for OPW Scheme With Buffer

In this section, we analyze the throughput of the OPW Scheme with buffer. Again, we use the independence approximations made in the analysis of the throughput of SHW scheme. At the beginning of a time slot, the transmitter at an arbitrary node, say node a , sends a packet \mathcal{P} to one of its neighbor, say node k . Before the start of the transmission at node a , if packet \mathcal{P} has already traveled i hops from its starting node ($d_H(a, s_{\mathcal{P}}) = i$), we say the packet is of type i . Notice that the definition of the type of a packet here is different from that in the previous section. In the analysis of the SHW, a packet of type i implies that it is i hops away from its destination node. During the transmission, we say that the packet is travelling on its $(i+1)$ th hop from its starting node. When \mathcal{P} arrives at node k , it becomes a type $(i+1)$ packet and has to compete with other arriving packets for the transmission right of the next time slot. Among all of the continuing packets at node k , a packet is said to be the winning packet if it traveled the longest hop distance from its origin node. In case of a tie, the winning packet is selected at random from the packets that have traveled the longest distance. When no continuing packets arrive at node k during a time slot, the winning packet is the head of buffer packet if the buffer is nonempty. Similarly, the newly generated packet is the winning packet if there are no continuing packets and no buffered packets at node k . Once a packet arrived its

destination node, it is immediately removed from the system. Also, when we say that node k receive a packet of type i , we implicit assume this packet has to travel at least one more hop to its destination (i.e., node k is not the packet's destination).

In the steady-state, due to symmetry, all nodes have identical statistics. Let A_i denote the event that an arbitrary node has a winning packet of type i (i.e., this packet will travel at least one more hop to its destination), and E denote the event that node k is idle. Again, we use the two approximations made previously to get a set of $P(A_i)$'s and $P(E)$, which solve the equations below and sum to one. The throughput can thus be obtained from $P(A_i), 0 \leq j \leq d-1$.

Let $\alpha(i) = P(A_i)$ and $C_{\text{pass}}(i) = \text{Pr}(\text{a packet must travel at least } i+2 \text{ hops on its way to the destination node} \mid \text{it must travel at least } i+1 \text{ hops to its destination})$

$$C_{\text{pass}}(i) = \frac{\sum_{j=i+2}^d n_x(j)}{\sum_{j=i+1}^d n_x(j)}.$$

To get $\alpha(i)$, note that a node has a winning packet of type i if and only if one of the following events occur during a time slot.

- Event O_1 : No continuing packet is transmitted to node k , and the head of buffer packet at node k is of type i .
- Event O_2 : Of the four receivers at node k , at least one received a packet of type i , while the others either did not receive any packet or received packet of type j ($j < i$).

We then have for $1 \leq i \leq d-1$

$$\begin{aligned} \alpha(i) = & \left[1 - \frac{1}{4} \sum_{j=0}^{d-2} \alpha(j) C_{\text{pass}}(j) \right]^4 (1 - b_0) \\ & \cdot \text{Pr}(H_i \mid \text{buffer nonempty}) \\ & + \binom{4}{1} \left[1 - \frac{1}{4} \sum_{j=i-1}^{d-2} \alpha(j) C_{\text{pass}}(j) \right]^3 \\ & \cdot \left[\frac{1}{4} \alpha(i-1) C_{\text{pass}}(i-1) \right] \\ & + \binom{4}{2} \left[1 - \frac{1}{4} \sum_{j=i-1}^{d-2} \alpha(j) C_{\text{pass}}(j) \right]^2 \\ & \cdot \left[\frac{1}{4} \alpha(i-1) C_{\text{pass}}(i-1) \right]^2 \\ & + \binom{4}{3} \left[1 - \frac{1}{4} \sum_{j=i-1}^{d-2} \alpha(j) C_{\text{pass}}(j) \right] \\ & \cdot \left[\frac{1}{4} \alpha(i-1) C_{\text{pass}}(i-1) \right]^3 \\ & + \left[\frac{1}{4} \alpha(i-1) C_{\text{pass}}(i-1) \right]^4 \end{aligned} \quad (20)$$

and

$$\alpha(0) = p_0 b_0 \left[1 - \frac{1}{4} \sum_{j=0}^{d-2} \alpha(j) C_{\text{pass}}(j) \right]^4. \quad (21)$$

The first term in (21) represents the probability of event O_1 , and the rest terms denotes the probability of event O_2 . The term $[1 - (1/4) \sum_{j=i-1}^{d-2} \alpha(j) C_{\text{pass}}(j)]$ represents the probability that there is no type j ($i-1 \leq j \leq d-2$) packet traveling on a particular link.

Let the events B_i and H_i be similarly defined as in the previous section. Then, we have for $0 \leq n \leq 4$

$$P(B_n) = \binom{4}{n} \left[\frac{1}{4} \sum_{j=0}^{d-2} \alpha(j) C_{\text{pass}}(j) \right]^n \cdot \left[1 - \frac{1}{4} \sum_{j=0}^{d-2} \alpha(j) C_{\text{pass}}(j) \right]^{4-n}.$$

Similar to the previous analysis on the throughput of SHW scheme, the probability that a buffered packet is of type $i, 1 \leq i \leq d-1$, is

$$P(H_i) = \frac{\alpha(i-1) C_{\text{pass}}(i-1) P(G_i)}{\sum_{j=0}^{d-2} \alpha(j) C_{\text{pass}}(j) P(G_{j+1})}. \quad (22)$$

Again, G_i is the event that an arbitrary packet has already traveled i hops from its starting node after reaching node k , it subsequently lost the contention with other packets at node k .

A packet, transmitted from one of node k 's neighbors (say node a), just finished traveling its i th hop may lose the contention at node k if one of the following events occur.

- Event $E_0(i)$: Out of the three remaining neighboring nodes of node k , there is at least one of them which is sending a packet of type j , where $j \geq i$, to node k .
— Comment: Since $j \geq i$, after reaching node k , that packet will be type $j+1$. The packet from node a will definitely lose in the competition since its hop distance to its source node, i , is strictly shorter.
- Event $E_1(i)$: Out of the three remaining neighboring node of node k , there is exactly one neighboring node, say b , which is also sending a packet of type $i-1$ to node k , while the other neighboring nodes are either not sending a packet to node k or sending packets of type j ($j < i-1$) to node k .
— Comment: The packet from node a will compete with the packet from node b for the transmission right of next slot. Since both of packets are of the same type, each one wins the competition with probability one half.
- Event $E_2(i)$: Out of the three remaining neighbor of node k , there are exactly two of them which are also sending a packet of type $i-1$ to node k , while the other neighboring node is either not sending a packet to node k or sending packet of type j ($j < i-1$) to node k .
- Event $E_3(i)$: For the three remaining neighbor of node k , each of them is sending a packet of type $i-1$ to node k .

We have for $1 \leq i \leq d-1$

$$P(E_0(i)) = 1 - \left[1 - \frac{1}{4} \sum_{j=i}^{d-2} \alpha(j) C_{\text{pass}}(j) \right]^3$$

and

$$P(E_0(d-1)) = 0 \quad (23)$$

$$P(E_1(i)) = \binom{3}{1} \left[\frac{1}{4} \alpha(i-1) C_{\text{pass}}(i-1) \right] \cdot \left[1 - \frac{1}{4} \sum_{j=i-1}^{d-2} \alpha(j) C_{\text{pass}}(j) \right]^2 \quad (24)$$

$$P(E_2(i)) = \binom{3}{2} \left[\frac{1}{4} \alpha(i-1) C_{\text{pass}}(i-1) \right]^2 \cdot \left[1 - \frac{1}{4} \sum_{j=i-1}^{d-2} \alpha(j) C_{\text{pass}}(j) \right] \quad (25)$$

$$P(E_3(i)) = \left[\frac{1}{4} \alpha(i-1) C_{\text{pass}}(i-1) \right]^3. \quad (26)$$

Now, $P(G_i)$ is obtained from the following equation:

$$P(G_i) = P(E_0(i)) + \frac{1}{2} P(E_1(i)) + \frac{2}{3} P(E_2(i)) + \frac{3}{4} P(E_3(i)).$$

The probability that there are i packets at a node's buffer b_i ($i = 0, 1, \dots, m$) has exactly the same form as in the previous section [see (13)–(17)]. Finally, we introduce the event that a packet reached its destination node given it has already traveled i hops. More precisely

$$C_{\text{end}}(i) = \Pr(\text{packet } \mathcal{P} \text{ reached its destination node in the next hop} \mid \text{it already traveled } i \text{ hops}) = \frac{\Pr(d_H(s_{\mathcal{P}}, t_{\mathcal{P}}) = i)}{\Pr(d_H(s_{\mathcal{P}}, t_{\mathcal{P}}) \geq i)} = \frac{n_x(i+1)}{\sum_{j=i}^d n_x(j)}. \quad (27)$$

With all equations available, the throughput of OPW scheme can be computed as follows:

$$\text{Throughput} = \sum_{j=0}^{d-1} \alpha(j) C_{\text{end}}(j).$$

Solving for the values of $\alpha(i)$ and subsequently the throughput numerically for a 11×11 mesh with a buffer size of four at each node, we again obtain accurate results as compared with simulations, as shown in Table II.

D. Throughput Analysis for OPW Scheme Without Buffer

The case of no buffer at each node is very similar to the case with buffer in terms of throughput analysis. With a few minor

TABLE II
COMPARISON OF SIMULATION RESULT AND THEORETICAL RESULT FOR 11×11 MESH USING OLDEST PACKET WIN SCHEME

p_0	Theoretical	Simulation	99.99% Confidence
	Throughput	Throughput	Interval
0.1	0.0690	0.0689	$0.0689 \pm 1.061 \times 10^{-4}$
0.2	0.1052	0.1051	$0.1051 \pm 1.446 \times 10^{-4}$
0.3	0.1274	0.1249	$0.1249 \pm 1.330 \times 10^{-4}$
0.4	0.1422	0.1421	$0.1421 \pm 1.641 \times 10^{-4}$
0.5	0.1526	0.1524	$0.1524 \pm 1.629 \times 10^{-4}$
0.6	0.1603	0.1579	$0.1579 \pm 1.665 \times 10^{-4}$
0.7	0.1661	0.1656	$0.1656 \pm 1.566 \times 10^{-4}$
0.8	0.1707	0.1700	$0.1700 \pm 1.616 \times 10^{-4}$
0.9	0.1744	0.1735	$0.1735 \pm 1.689 \times 10^{-4}$
0.95	0.1759	0.1748	$0.1748 \pm 1.616 \times 10^{-4}$
0.99	0.1771	0.1760	$0.1760 \pm 1.678 \times 10^{-4}$

modifications to (21), we get the following equations ($1 \leq i \leq d-1$):

$$\alpha(i) = \binom{4}{1} \left[1 - \frac{1}{4} \sum_{j=i-1}^{d-2} \alpha(j) C_{\text{pass}}(j) \right]^3 \cdot \left[\frac{1}{4} \alpha(i-1) C_{\text{pass}}(i-1) \right] + \binom{4}{2} \left[1 - \frac{1}{4} \sum_{j=i-1}^{d-2} \alpha(j) C_{\text{pass}}(j) \right]^2 \cdot \left[\frac{1}{4} \alpha(i-1) C_{\text{pass}}(i-1) \right]^2 + \binom{4}{3} \left[1 - \frac{1}{4} \sum_{j=i-1}^{d-2} \alpha(j) C_{\text{pass}}(j) \right] \cdot \left[\frac{1}{4} \alpha(i-1) C_{\text{pass}}(i-1) \right]^3 + \left[\frac{1}{4} \alpha(i-1) C_{\text{pass}}(i-1) \right]^4 \quad (28)$$

and

$$\alpha(0) = p_0 \left[1 - \frac{1}{4} \sum_{j=0}^{d-2} \alpha(j) C_{\text{pass}}(j) \right]^4. \quad (29)$$

For the 11×11 mesh, we again calculate the theoretical throughput of the system without buffer and compare it with the simulation results. We see that the throughput for a system with buffer is significantly greater than the throughput of a system without buffer, shown in Fig. 4.

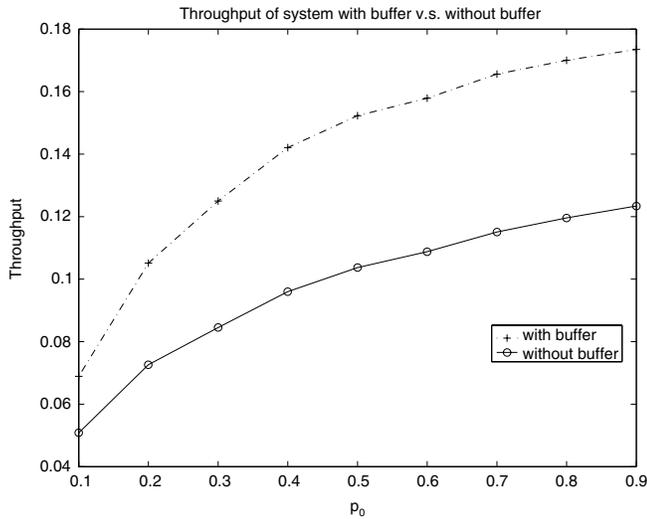


Fig. 4. Comparison of throughput of system with buffer size of 4 and without buffer using OPW.

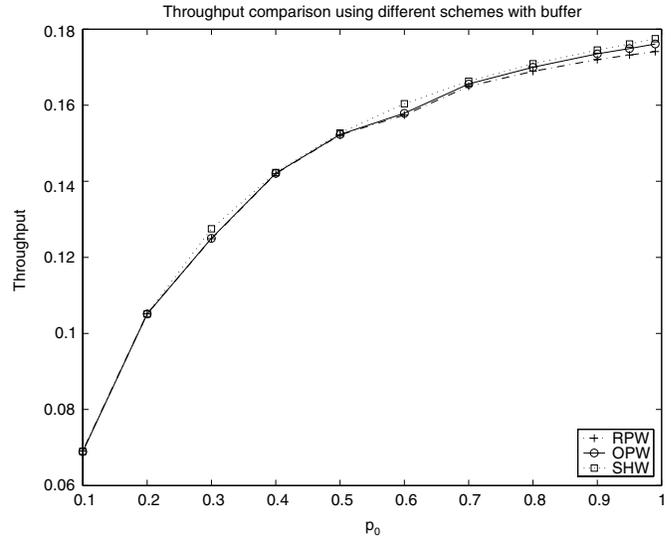


Fig. 6. Comparison of throughput of system with buffer size of 4 using different schemes.

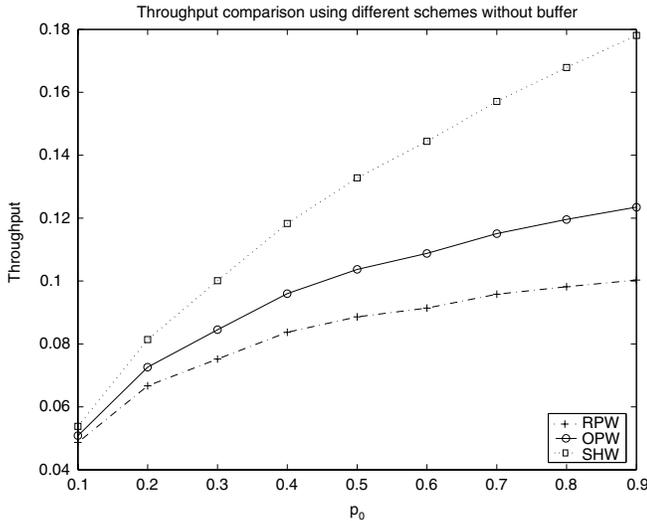


Fig. 5. A comparison of throughput of system without buffer using different schemes.

E. Comparison of Different Schemes in the No Buffer Case

Following the analysis of previous two sections, we can also get the throughput for the RPW scheme (with buffer or without buffer). For system without buffer, of the three schemes discussed so far (SHW, OPW, RPW), we expect that SHW to perform better than the other two schemes in terms of throughput since the continuing packets in the system are likely to have a shorter distance to the destination node. Also, the OPW scheme should perform better than the RPW scheme since it tries to minimize the amount of wasted work done for a continuing packet. Fig. 5 below substantiates these expectations. Again, notice that the SHW scheme achieves throughput that are very close to the theoretical limit of Theorem 1, which in the case of an 11×11 -mesh is limited to 0.182.

F. Simulation of Other Schemes

The theoretical analysis of the RPW scheme can be carried out by following the analysis in the previous two sections.

Here, we provide simulation results of the RPW schemes and compare its performance with the other two schemes. First, we want to compare the throughput of a system with buffer under the RPW scheme, OPW scheme, and SHW scheme. For all three schemes, the continuing packet is given the highest priority. Buffered packet will be transmitted only if there are no continuing packets. Likewise, the newly generated packet will be transmitted only if there are no continuing packets and no buffered packets. Fig. 6 plots the throughput for the three scheme. We expected that SHW scheme would perform significantly better than the other two schemes just as it did in the no buffer case. Surprisingly, however, we see that the throughput for three schemes are about the same, although SHW performs slightly better than the other two schemes in the high p_0 region. It seems that the buffer has a neutralizing effects on the system's throughput (i.e., the choice of which scheme to use becomes less important). An explanation to the rather counterintuitive result is the following. After a packet arrived at the receiving node, the packet which lost the contention is stored in the buffer if there is any space available. Notice that we do not decide which packet to put in the buffer. If there is enough space for all of the packets which did not win the contention, all of them will be stored in the buffer. In the event that there is not enough buffer space for all losing packets, we randomly pick amongst them to be placed in the remaining spots of the buffer. The packets in the buffer will eventually be transmitted. Unlike the system without buffer, these packets are not dropped immediately, although they did not win the contention. It is in this sense that the contention is not a strict competition (since they are still in the system). Therefore, the difference in throughput using different schemes is not significant. To increase the throughput, one may want to develop an additional scheme in choosing which packet to be sent to the buffer instead of choosing it randomly.

To verify the above explanation, we also investigate the throughput of a rather "bad" scheme called furthest hop win scheme. This scheme is identical to the SHW scheme except

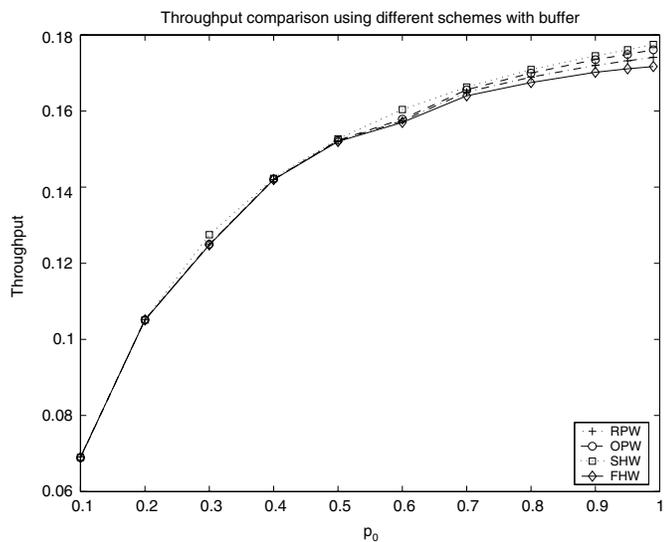


Fig. 7. Comparison of throughput of system with buffer size of 4 using FHW.

that during a contention the latter scheme chooses the packet with shortest hop distance to the destination to win while the former scheme chooses the packet with longest hop distance to the destination to win. We expect that the furthest hop win (FHW) scheme would perform much worse than all of the schemes mentioned so far. However, as Fig. 7 shows, the throughput of FHW scheme performs only slightly worse than other schemes. This further substantiates our expectation that the buffer has a neutralizing effect on system throughput for the various schemes.

For all of the schemes mentioned so far, the highest priority is assigned to the continuing packets and the new packets can only enter the system if there are no continuing packets or buffered packets. This prompts us to think that the throughput can be improved if we allow the continuing packets, the head of buffer packet, and the newly generated packet to compete for the transmission right in the next slot instead of just letting the continuing packets compete. A modified version of SHW scheme chooses the packet (among all available packets) with shortest hop distance to its destination node to win. Packets which loss the contention will be stored in the buffer if there is space available. We call this scheme shortest hops win (SHW) scheme 2. Simulation result (see Fig. 8) shows that throughput is higher than the four schemes discussed so far. However, this scheme has one drawback. A closer look at the distribution of number of hops to the destination node for the head of buffer packet reveals that, with high probability, the head of buffer packet has a long hop distance to its destination node. This can be explained by noting that the head of buffer packet is transmitted only if it has the shortest hop distance to its destination. Consequently, packets with longer hop distance to their destination than the head of buffer packet will be placed in the buffer. Eventually, the buffer will be filled with packets with d hops (the maximum hop distance between a source and destination pair) to their destination nodes. As a result, the buffered packets do not get sent by the transmitter. Thus, we develop next a scheme which will solve this problem while still maintaining high level of throughput.

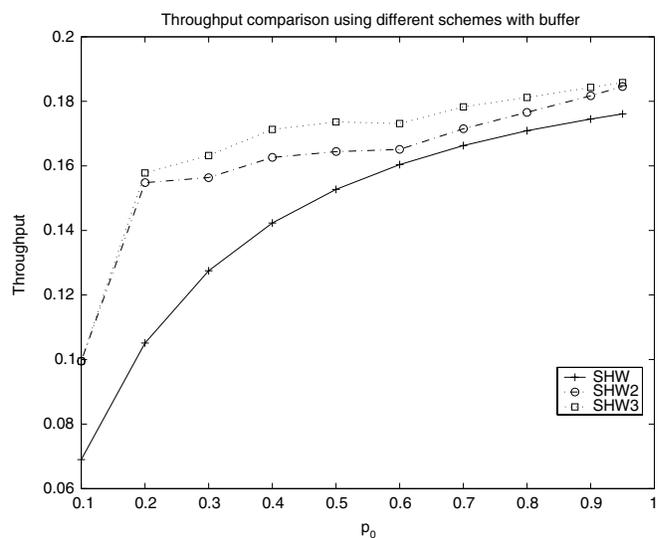


Fig. 8. Comparison of throughput of system with buffer size of 4 using SHW2 and SHW3.

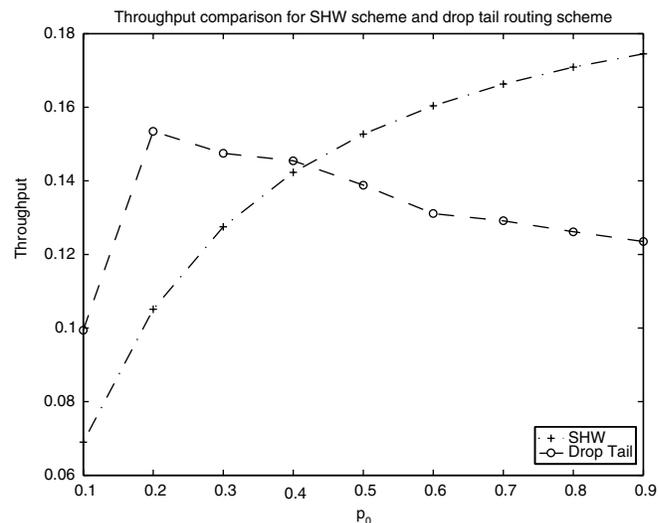


Fig. 9. Comparison of drop tail routing and SHW routing.

We call this scheme SHW scheme 3. It achieves high throughput when p_0 is relatively small. SHW3 works as follows. The new packet and the arriving packets are sent to the buffer if there is space available. If there is not enough space available for all the incoming packets (including the arriving packets and the new packet), we put packets with shorter hop distance to the destination node in the buffer first. At the beginning of a time slot, the packet at the head of buffer is going to be transmitted. Simulation shows that SHW3 achieves high throughput level when p_0 is small ($p_0 \approx 0.2$). However, unlike SHW scheme where packets are prevented from entering the network but seldom dropped while inside the network; SHW2 and SHW3 drop most packets once they have already entered the network. This is a significant disadvantage as it puts an increased burden on higher layer protocols to recover the dropped packets.

Finally, we compare the commonly used Internet queuing method, FIFO, and drop tail routing policy, with SHW scheme. In FIFO drop tail policy, arriving packets are dropped when the

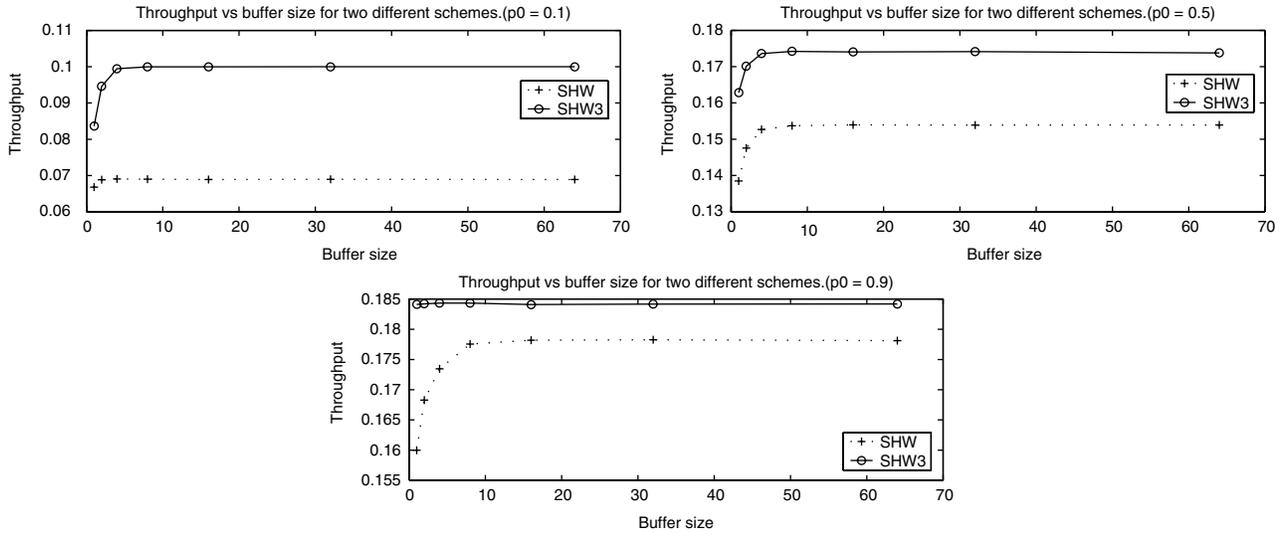


Fig. 10. Relation of throughput and size of buffer using SHW and SHW3.

buffer is full. The head of buffer packet always gets served first, while the arriving packets are placed at the end of the queue if there is space available. The resulting throughput of the drop tail method is shown in Fig. 9. Using the drop tail method, the buffer will be filled very quickly as the new packet's arrival rate increases. When a continuing packet arrives at a particular node, it has a rather high probability to be dropped due to buffer overflow. Thus, throughput of the drop tail policy decreases as p_0 becomes large.

G. Throughput and Buffer Size

To investigate the relationship between the buffer size and the throughput, we evaluate the throughput for SHW and SHW3 at $p_0 = 0.1, p_0 = 0.5$, and $p_0 = 0.9$ by using various buffer sizes. Fig. 10 illustrates that a network with moderate buffer size such as four or eight can achieve the same level of throughput as a network with significantly larger buffer size. In other words, the throughput of system does not increase with the increase of buffer size.

H. Nonuniform Traffic Model

So far the analysis and simulation are based on the uniform traffic model. However, uniform traffic, while it gives us insight on the performance of various routing schemes, is unlikely in a real system. Here, we investigate the possible impact of nonuniform traffic on the throughput of routing scheme discussed in this paper through simulation. Specifically, we consider two forms of nonuniform traffic in this section: *distance-dependent traffic* and *hot spot traffic*.

For distance-dependent traffic, the probability that a newly generated packet is i hops to its destination depends on the number of hops i . Specifically, a destination node that is i hops away from the source node is generated with probability $(N - i)/C$ for an $N \times N$ mesh. The constant C is a normalization constant satisfying $\sum_{i=1}^{N-1} n(i)(N - i)/C = 1$, where $n(i)$ denotes the number of nodes that is i hops away. The distance dependent traffic model defined above generates

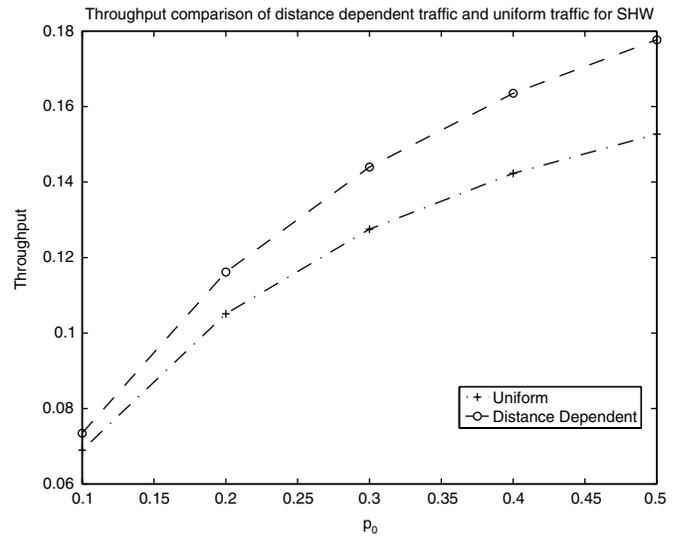


Fig. 11. Throughput comparison of distance dependent traffic and uniform traffic.

packets with probability that decreases proportionally with the distance to their destination.

For hot spot traffic, we pick a node arbitrarily from the constellation to be the hot spot node. Then, we add a hot spot traffic on top of the uniform all-to-all traffic, in which new packets are still generated with probability p_0 during each slot except the hot spot node. In addition, every node in the constellation generates new packets to the hot spot node with probability $p_{h_{in}}$. The hot spot node also generates a new packet to every other node with probability $p_{h_{out}}$.

Since the distance dependent traffic generates new packet that has short distance to its destination with high probability, we expect the throughput of the system would increase comparing with the throughput of the system with a uniform traffic. Using SHW scheme, the simulation result in Fig. 11 shows the expected increase. For hot spot traffic, since all nodes want to send packet to the hot spot node, the links around the hot spot will be highly congested. As a result, much of the hot spot

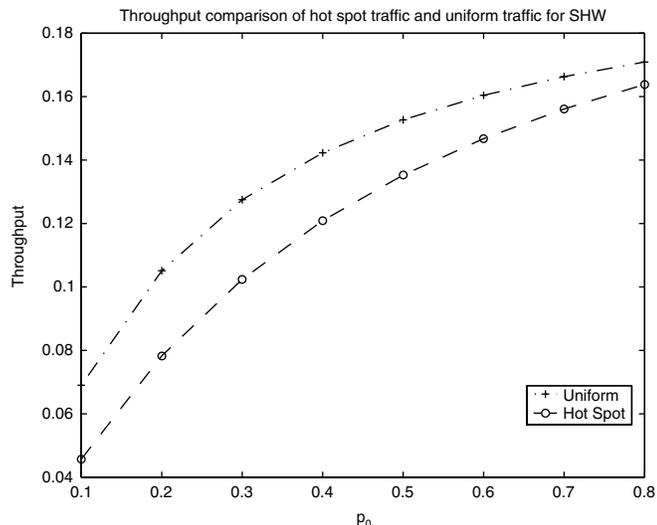


Fig. 12. Throughput comparison of hot spot traffic and uniform traffic.

traffic is dropped thus resulting in a decrease of throughput. Fig. 12 shows the throughput of the SHW scheme using hot spot traffic and uniform traffic. In the simulation, $p_{h_{in}} = 0.1$ and $p_{h_{out}} = 0.8$.

I. Simulation Accuracy

To establish credible simulation results, we used the methods of independent replications in [16] to get the sample mean, sample variance, and the confidence interval. Specifically, for each data point collected (e.g., the throughput result of SHW scheme with $p_0 = 0.2$), the simulation is independently repeated 200 times. For sample mean and sample variance, we use the following formulas: sample mean $= \bar{X} = 1/n \sum_{i=1}^n X_i$ and sample variance $= S^2 = 1/(n(n-1)) \sum_{i=1}^n (X_i - \bar{X})^2$. We tabulate the 99.99% confidence interval (within four standard deviations) with our original data in Tables I and II. For the data points that appeared in Figs. 6 to 12, since the variance of each data point is very small (in the range of $\pm 2 \times 10^{-4}$), it is difficult to depict it in the figure. We thus omit it. From the confidence interval, we see that our simulation results are consistent and accurate.

IV. SUMMARY

This paper considers packet routing and transmission scheduling in packet switched LEO satellite network. In particular, we examined the impact of limited transmitters and buffer space on the throughput of such networks. We modeled the network as an $N \times N$ -mesh network and analyzed throughput performance of the system. First, we showed the stability region of such network. Then, we consider three transmission schemes (SHW, RPW, and OPW) and compare their throughput performance using fixed-point analysis for the system's Markov chain. As multiple packets arrive at a particular node in a time slot, SHW chooses the one with shortest hop distance to its destination to be transmitted in the next slot, RPW randomly picks one to be transmitted, and OPW selects the one with longest hop distance to the destination. When the satellite nodes have no buffers, both the analytic and simulated results show that SHW

scheme attains the best throughput performance, followed by the OPW scheme and the RPW. In the presence of even limited buffers, the three schemes obtain similar throughput performance. More importantly, we observed that a small buffer size can achieve throughput close to that of the infinite buffer size. Of course, smaller buffers may lead to an increase in buffer overflow probability; however, we observed that most of the dropped packets occurred at the source node. That is, using our routing schemes, new packets had lower priority than continuing packets and, hence, when contention occurred at a node, new packets were prevented from entering the network. Once a packet entered the network, it reached its destination with very high probability.

Our paper has primarily focused on throughput analysis for a network with a single transmitter and multiple receivers. Similar techniques can be easily followed to analyze the throughput under a different combination of receivers and transmitters. The analysis of throughput provides us with insight to the design of such packet switched systems. In a practical system, the arrival rate of new packets into the network must not exceed the throughput value in order for the system to be stable. Of course, due to contention, some packets may be prevented from entering the network, or dropped and subsequently retransmitted. Such retransmissions can be done by higher layer protocols, such as a link layer protocol that operates over the satellite system.

REFERENCES

- [1] E. Modiano and A. Ephremides, "Efficient algorithms for performing packet broadcasts in a mesh network," *IEEE/ACM Trans. Networking*, vol. 4, pp. 639–648, Aug. 1996.
- [2] M. C. Azizoglu and O. Egecioglu, "Lower bounds on communication loads and optimal placements in torus networks," *IEEE Trans. Comput.*, vol. 49, pp. 259–266, Mar. 2000.
- [3] P. W. Lemme, S. M. Glenister, and A. W. Miller, "Iridium aeronautical satellite communications," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 14, pp. 11–16, Nov. 1999.
- [4] D. P. Patterson, "Teledesic: A global broadband network," in *Proc. 1998 IEEE Aerospace Conf.*, vol. 4, 1998, pp. 547–552.
- [5] E. Kicici, I. F. Akyildiz, and M. D. Bender, "A distributed routing algorithm for datagram traffic in LEO satellite networks," *IEEE/ACM Trans. Networking*, vol. 9, pp. 137–147, Apr. 2001.
- [6] G. D. Stamoulis and J. N. Tsitsiklis, "Efficient routing schemes for multiple broadcasts in hypercubes," *IEEE Trans. Parallel Distrib. Syst.*, vol. 4, pp. 725–739, July 1993.
- [7] E. Varvarigos, "Efficient routing algorithms for folded-cube networks," *Proc. IEEE 14th Annu. Int. Phoenix Conf. Computers Communications*, pp. 143–151, 1995.
- [8] J. Sun and E. Modiano, "Capacity provisioning and failure recovery in mesh-torus networks with application to leo satellite constellations," presented at the IEEE Int. Symp. Computer Communications, Taormina, Italy, July 2002.
- [9] D. P. Bertsekas, *Network Optimization: Continuous and Discrete Models*. Belmont, MA: Athena Scientific, 1998.
- [10] A. G. Greenberg and B. Hajek, "Deflection routing in hypercube networks," *IEEE Trans. Commun.*, vol. 40, pp. 1070–1081, June 1992.
- [11] G. D. Stamoulis and J. N. Tsitsiklis, "The efficiency of greedy routing in hypercubes and butterflies," *IEEE Trans. Commun.*, vol. 42, pp. 3051–3061, Nov. 1994.
- [12] E. A. Varvarigos and D. P. Bertsekas, "Performance of hypercube routing schemes with or without buffering," *IEEE/ACM Trans. Networking*, vol. 2, pp. 299–311, June 1994.
- [13] D. Bertsimas, Stability, Performance, and Optimization of Queueing Systems. Lecture notes.
- [14] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1991.

- [15] J. Walrand, *An Introduction to Queueing Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
- [16] K. Pawlikowski, "Steady-state simulation of queueing processes: A survey of problems and solutions," *ACM Comput. Surv.*, vol. 22, pp. 123–170, June 1990.



Jun Sun received the B.S. degree in computer engineering from University of Florida, Gainesville, in 1997 and the M.S. degree in electrical engineering from Massachusetts Institute of Technology (MIT), Cambridge, in 2002. He is currently working toward the Ph.D. degree in the Laboratory for Information and Decision Systems, MIT.

His research interest is on communication networks with emphasis on satellite and wireless networks.



Eytan Modiano (S'90–M'93–SM'00) received the B.S. degree in electrical engineering and computer science from the University of Connecticut, Storrs, in 1986, and the M.S. and Ph.D. degrees, both in electrical engineering, from the University of Maryland, College Park, in 1989 and 1992, respectively.

He was a Naval Research Laboratory Fellow from 1987 to 1992 and a National Research Council Post Doctoral Fellow from 1992 to 1993, while he was conducting research on security and performance issues in distributed network protocols. From 1993 to

1999, he was with the Communications Division, Massachusetts Institute of Technology (MIT), Lincoln Laboratory, Cambridge, where he designed communication protocols for satellite, wireless, and optical networks and was the Project Leader for MIT Lincoln Laboratory's Next-Generation Internet (NGI) project. He joined the MIT faculty in 1999, where he is presently an Associate Professor in the Department of Aeronautics and Astronautics and the Laboratory for Information and Decision Systems (LIDS). His research is on communication networks and protocols with emphasis on satellite, wireless, and optical networks.