

# IBM Research Report

## Buffer Scalability of Wireless Networks

**Predrag R. Jelenkovic**

Department of Electrical Engineering  
Columbia University  
New York, NY 10027

**Petar Momcilovic**

Department of Electrical Engineering and Computer Science  
University of Michigan  
Ann Arbor, MI 48109

**Mark S. Squillante**

IBM Research Division  
Thomas J. Watson Research Center  
P.O. Box 218  
Yorktown Heights, NY 10598



Research Division

Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

# Buffer Scalability of Wireless Networks

Predrag R. Jelenković

Petar Momčilović

Mark S. Squillante

**Abstract**—We investigate the existence of scalable protocols that can achieve the capacity limit of  $c/\sqrt{N}$  per source-destination pair in a large wireless network of  $N$  nodes when the buffer space of each node does not grow with the size of the network  $N$ . We show that there is no end-to-end protocol that is capable of carrying out the limiting throughput of  $c/\sqrt{N}$  with nodes that have constant buffer space. In other words, this limit is achievable only with devices whose buffers grow with the size of the network. On the other hand, we establish that there exists a protocol which realizes a slightly smaller throughput of  $c/\sqrt{N \log N}$  when devices have constant buffer space. Furthermore, we show that the required buffer space can be very small, capable of storing just a few packets. This is particularly important for wireless sensor networks where devices have limited resources. Finally, from a mathematical perspective, this paper furthers our understanding of the difficult problem of analyzing large queueing networks with finite buffers for which, in general, no explicit solutions are available.

**Index Terms**—Ad-hoc wireless network, wireless sensor network, large-scale network, scaling laws, finite buffer queueing network.

## I. INTRODUCTION

The growth of modern communication infrastructures, such as the Internet and various wireless networks, over the last decade has surpassed the expectations of everyone. Indeed, going back in time to the origins of these networks, it would have been hard even to imagine the importance and scale to which these networks have developed. Now, projecting into the future, we strongly believe that this growth trend will only continue, if not accelerate. Hence, the communication devices and protocols of today must be capable of operating with the same efficiency in the very large-scale networks of the future. In this regard, we investigate the behavior of a network as its size grows while the buffer space of each device, one of the primary communication resources, stays finite. Specifically, we term a protocol, or in general a network architecture, to be buffer scalable, or simply scalable, if the performance of the network does not degrade as its size grows due to the limited buffer space in each node, i.e., buffer space is not a source of bottlenecks.

The critical importance of scalability is widely recognized by practitioners and well documented in the systems engineering literature. However, the definitions of scalability are often either vague or simply absent. Hence, we believe that

there is a strong need for developing a precise mathematical foundation for investigating the notions of scalability. Furthermore, since building experimental large-scale networks is basically impractical, the theoretical approach has a unique advantage for providing insights into the scaling properties of communication networks.

One of the basic concerns in building large-scale networks is that minor inefficiencies which can be well tolerated in small networks can accumulate and become dominant factors in large networks. To further illustrate this point, consider a single link of capacity one packet/time slot and no buffer space. Suppose that two independent flows with identical statistical properties and average rate of 0.25 packets/time slot share this link. Now, even if we assume that the flows are quite “smooth”, just a small amount of jitter that can be represented with Bernoulli arrivals can result in the reduction of throughput to 0.4375 instead of 0.5 packets/time slots. Thus, in the presence of multi-hop routes such small inefficiencies could accumulate and yield a significant throughput decrease.

Using this simple example as motivation, we focus on how the throughput scales in large wireless networks with finite buffers. Before discussing our results we present a brief overview of the literature that investigates the capacity limits of wireless networks. In [1], a random network model of static wireless networks was introduced. For this model, the authors showed that throughput per source-destination pair is  $O(1/\sqrt{N})$  as  $N \rightarrow \infty$ , where  $N$  is the number of nodes in the network (refer to Appendix A for definitions of the standard asymptotic notation used throughout the paper). In the same paper a scheme that achieves  $\Theta(1/\sqrt{N \log N})$  per source-destination pair was presented. The scheme was generalized to a parametrized version [2] (see also [3], [4]) that results in the optimal throughput-delay trade-off for throughputs that are  $O(1/\sqrt{N \log N})$ . In [5] it was shown that throughput  $\Theta(1/\sqrt{N})$  in fact can be achieved. Results in [1] were rederived in [6] for the corresponding constant packet size model. Extensions of the original model and their analyses can be found in [7], [8], [9]. We note that the capacity of wireless networks relates to the capacity of lattices – refer to [10] and the references therein.

In this paper, we investigate the existence of scalable protocols that can achieve the capacity limit, derived in [1], of  $c/\sqrt{N}$  per source-destination pair in a large wireless network of  $N$  nodes. More specifically, focusing on scalability with respect to the buffer space, we establish that there is no protocol with end-to-end acknowledgments that is capable of carrying out the limiting throughput of  $c/\sqrt{N}$  when each node has constant buffer space. In other words, this limit is achievable only with devices whose buffer grows with the size of the network. On the other hand, we show that there exists a protocol which realizes a slightly smaller throughput

P.R. Jelenković is with the Department of Electrical Engineering, Columbia University, New York, NY 10027; e-mail: predrag@ee.columbia.edu. Work is supported by NSF Award No. 0435168.

P. Momčilović is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109; e-mail: petar@eecs.umich.edu.

M.S. Squillante is with the Mathematical Sciences Department, IBM T.J. Watson Research Center, Yorktown Heights, NY 10598; e-mail: mss@us.ibm.com.

of  $c/\sqrt{N \log N}$  with devices that have constant buffer space. Furthermore, we establish that the required buffer space can be very small, capable of storing just a few packets, which is particularly important for wireless sensor networks where devices have limited resources. In addition, we show that any protocol which is capable of achieving the throughput of  $c/\sqrt{N \log N}$  with finite buffers has to employ a local buffer coordination scheme. This insight may provide guidance for designing efficient scalable wireless protocols in practice.

The present study is related to the study of the throughput-delay trade-off [2], [3], [4] since the finiteness of buffers limits the delay incurred by packets, given that these packets are not dropped. However, apart from this heuristic connection, the expected delay considered in [2], [3], [4] does not provide results on buffer scalability because the dynamics of the network change under the constraints imposed by finite buffers.

Finally, from a mathematical perspective, this paper furthers our understanding of the difficult problem of analyzing large queueing networks with finite buffers for which no general explicit solutions, e.g., the product-form solution of Jackson networks, are available.

The paper is organized as follows. The next section contains some of the preliminary results on a single server queue model. In Section III we introduce a motivating example of a one-dimensional multi-hop route. A model of a wireless network is described and discussed in Section IV. The main results of the paper are presented in Section V. Concluding remarks can be found in Section VI. The Appendix provides additional mathematical details.

## II. PRELIMINARIES ON A SINGLE SERVER QUEUE

This section contains basic results on a single server queue that will be used in the remainder of the paper. Throughout the section we assume that the server works at unit rate with the packet (customer) service requirements being independent and identically distributed (i.i.d.) exponential random variables with mean  $\mu^{-1}$ , independent from the packet arrival times. The service discipline is first-come-first-served (FCFS).

We start by considering two queues with buffers of sizes  $b_1$  and  $b_2$  packets (including any one in service), respectively. Let  $\{A_i^{(j)}\}$  and  $\{D_i^{(j)}\}$ ,  $j = 1, 2$ , be positive, monotonically increasing sequences, representing customer arrival and departure times to these queues, respectively. In general, we use superscripts to distinguish quantities that relate to different queues. Let  $Q^{(b_j)}(t)$  denote the number of customers in queue  $j$  (of size  $b_j$ ) at time  $t$  and define

$$\Lambda_b^{(j)}(t) := \sum_{i: A_i^{(j)} \leq t} \mathbf{1}\{Q^{(b_j)}(A_i^{(j)}) \geq b\}.$$

Note that, when  $b_1 < \infty$ , the quantity  $\Lambda_{b_1}^{(1)}(t)$  represents the number of packets lost in queue 1 in the interval  $[0, t]$ .

The first lemma establishes a stochastic comparison of the two queues with different buffer sizes when the arrival sequences are monotonically related.

*Lemma 1:* If  $Q^{(b_1)}(0) = Q^{(b_2)}(0)$ ,  $b_1 \leq b_2 \leq \infty$  and  $\{A_i^{(1)}\} \subseteq \{A_i^{(2)}\}$ , then  $\{D_i^{(1)}\} \subseteq \{D_i^{(2)}\}$  and, for all  $b, t \geq 0$ ,

$$\mathbb{E}\Lambda_b^{(1)}(t) \leq \mathbb{E}\Lambda_b^{(2)}(t). \quad (1)$$

*Proof:* The first step of the proof is to establish that there is a coupling between the service times of the two queues such that, for all  $t > 0$ ,

$$Q^{(b_1)}(t) \leq Q^{(b_2)}(t). \quad (2)$$

To this end, we couple the service times in these queues such that the packets with overlapping time in service depart simultaneously from both systems. The coupling is feasible due the memoryless property of the exponential distribution of service times.

Then, the proof is by induction over the times  $\{t_i\}_{i \geq 0}$ ,  $t_0 \equiv 0$ , right after an arrival or departure in either queue. Suppose that  $Q^{(b_1)}(t_j) \leq Q^{(b_2)}(t_j)$ ,  $j \leq i$  for some  $i > 0$ . If an arrival occurred right before  $t_{i+1}$ , then (2) holds for  $t = t_{i+1}$  due to the assumption  $\{A_i^{(1)}\} \subseteq \{A_i^{(2)}\}$ . On the other hand, if a departure occurs right before  $t_{i+1}$  then either customers depart from both queues or only from the one with buffer  $b_2$ . However, in the latter case (due to the coupling) one has  $Q^{(b_1)}(t_{i+1}) = 0$  and, therefore, (2) holds. Note that the coupling also yields  $\{D_i^{(1)}\} \subseteq \{D_i^{(2)}\}$ .

Finally, we consider (1). The definition of functions  $\Lambda_b^{(j)}$  and (2) yield

$$\begin{aligned} \Lambda_b^{(1)}(t) &= \sum_{i: A_i^{(1)} \leq t} \mathbf{1}\{Q^{(b_1)}(A_i^{(1)}) \geq b\} \\ &\leq \sum_{i: A_i^{(1)} \leq t} \mathbf{1}\{Q^{(b_2)}(A_i^{(1)}) \geq b\}. \end{aligned}$$

From the preceding inequality, the non-negativity of the indicator function and the assumption of the lemma, it easily follows that

$$\begin{aligned} \Lambda_b^{(1)}(t) &\leq \sum_{i: A_i^{(2)} \leq t} \mathbf{1}\{Q^{(b_2)}(A_i^{(2)}) \geq b\} \\ &= \Lambda_b^{(2)}(t). \end{aligned}$$

This concludes the proof of the lemma.  $\blacksquare$

Our next goal is to establish a lower bound on the loss rate in a finite buffer queue with stationary and ergodic arrival times. Recall that the service times are assumed to be i.i.d. exponential and independent from the arrival times. Note that the minimum loss rate for arrival sequences with specified mean will be positive since the buffer is finite and service times have unbounded support.

More formally, consider a positive increasing sequence of arrival times  $\{A_i\}_{i \geq 0}$ ,  $A_0 = 0$ , with stationary and ergodic interarrival times  $\{A_{i+1} - A_i\}_{i \geq 0}$  that are equal in distribution to  $A$ . Let  $\rho := (\mu \mathbb{E}A)^{-1}$  denote the offered traffic load, where  $\mu^{-1}$  is the average service time. Note that since the buffer is finite, the offered load does not need to be below 1. Given that the buffer can accommodate  $b$  customers (including any one in service), the loss probability  $p(b)$ , i.e., the long-term fraction of lost packets, is defined as

$$p(b) := \lim_{k \rightarrow \infty} \frac{1}{k} \sum_{i=1}^k \mathbf{1}\{Q^{(b)}(A_i) = b\}, \quad (3)$$

where  $Q^{(b)}(t)$  is the number of packets in the queue at time  $t$  and the existence of the limit is ensured by stationarity and ergodicity; recall that  $Q^{(b)}(t)$  is right-continuous and, thus,  $Q^{(b)}(A_i)$  represents the number of customers in the queue just before the  $i$ th arrival. In addition, using the ergodicity and stationarity assumptions it is easy to show that the *loss rate*  $\kappa(b)$  satisfies

$$\kappa(b) = p(b)/\mathbb{E}A.$$

The loss probability can be lower bounded as in the following lemma.

*Lemma 2:* For any  $b > 0$ ,

$$p(b) \geq (b+1)^{-1}e^{-b/\rho}.$$

*Remark 1:* If the arrival sequence is assumed to be i.i.d., a tighter lower bound can be obtained by analyzing a finite buffer GI/M/1 queue.

*Proof:* Let  $N(t)$  be the number of arrivals in  $[0, t]$  in a Poisson process of rate  $\mu$ . Note that  $N(A_k)$  can be interpreted as the number of potential customers served between the arrival of  $(k+1)$  customers. Given that the system is in stationarity, for any  $k > 0$ ,

$$p(b) = k^{-1}\mathbb{E}\left[\sum_{i=1}^k \mathbf{1}\{Q^{(b)}(A_i) = b\}\right], \quad (4)$$

since

$$p(b) = \mathbb{P}[Q^{(b)}(A_i) = b].$$

Now note that the event  $\{k - N(A_{k-1}) > b\}$  implies that at least one packet is lost in the time interval  $[A_1, A_k]$ . Besides the non-negativity of the queue at  $t = 0$ , this follows from the observation that exactly  $k$  customers arrive in  $[A_1, A_k]$  and at most  $N(A_{k-1})$  customers depart. Hence, (4) renders

$$\begin{aligned} p(b) &\geq k^{-1}\mathbb{P}[k - N(A_{k-1}) > b] \\ &= k^{-1}\mathbb{P}[N(A_{k-1}) < k - b]. \end{aligned}$$

Setting  $k = b + 1$ , results in

$$\begin{aligned} p(b) &\geq (b+1)^{-1}\mathbb{P}[N(A_b) = 0] \\ &= (b+1)^{-1}\mathbb{E}e^{-\mu A_b} \end{aligned}$$

that, in conjunction with Jensen's inequality, yields

$$p(b) \geq (b+1)^{-1}e^{-\mu\mathbb{E}A_b}.$$

The statement follows from the preceding inequality and  $\mathbb{E}A_b = b\mathbb{E}A$ . ■

Next, we consider a buffer management policy under which the buffer is divided into  $s$  partitions, each being able to accommodate  $b$  customers (including possibly one in service). Upon an arrival to the system, a customer uniformly at random chooses a partition of the buffer. If the number of customers in the chosen partition is less than  $b$ , then the newly arrived customer is accommodated in the given partition. Otherwise, one packet is dropped from the partition (at this point we do not specify whether it is the newly arrived customer or one of the customers in the partition). Intuitively, since a packet can be lost even if some partitions are not full, it is clear

that the given buffer management policy results in a higher loss probability than the one with complete buffer sharing (single partition of size  $sb$ ). The service policy is FCFS. As in (3), denote the long-term fraction of lost customers with  $p(b, s)$ ; note that the stationarity and ergodicity of the system follows from the stationarity and ergodicity of the arrival sequence, monotonicity of this queueing system and Loynes' construction [11]. The following lemma estimates losses in this slightly more complicated system. By setting the number of partitions to 1 ( $s = 1$ ), Lemma 3 reduces to Lemma 2.

*Lemma 3:* For any  $b > 0$  and  $s > 0$ ,

$$p(b, s) \geq s^{-b}(b+1)^{-1}e^{-b/\rho}.$$

*Proof:* The proof is very similar to the proof of Lemma 2. Consider a sequence of  $(b+1)$  arrivals to the system. Then, the probability of losing a single customer (out of  $(b+1)$ ) due to an overflow of a particular partition is bounded from below by

$$s^{-(b+1)}(b+1)^{-1}\mathbb{E}e^{-\mu A_b} \geq s^{-(b+1)}(b+1)^{-1}e^{-b/\rho}.$$

Thus, the loss probability is lower bounded by

$$\begin{aligned} p(b) &\geq s s^{-(b+1)}(b+1)^{-1}e^{-b/\rho} \\ &= s^{-b}(b+1)^{-1}e^{-b/\rho} \end{aligned}$$

since a packet can be lost due to an overflow in any one of the  $s$  partitions. ■

This section is concluded with an estimate on a hitting time in a single server queue. In particular, we consider a multi-class FCFS M/M/1 queue with service rate  $\mu$  and aggregate arrival rate  $\lambda < \mu$ . Class  $i$  customers arrive to the system at Poisson rate  $\lambda_i$  (the number of classes  $C$  is finite),  $\sum \lambda_i = \lambda$ . Service times are class independent, and the system is in steady state at time  $t = 0$ . Let  $l_i(t)$  be the number of class  $i$  customers in the system at time  $t$  (including possibly one in service). For a set  $\Gamma \subseteq \{1, \dots, C\}$  define

$$l_\Gamma(t) := \sum_{i \in \Gamma} l_i(t), \quad \lambda_\Gamma := \sum_{i \in \Gamma} \lambda_i.$$

Let  $\tau_\Gamma(b)$  be the first time that the number of customers in classes belonging to  $\Gamma$  reaches level  $b$ , i.e.,

$$\tau_\Gamma(b) = \inf\{t > 0 : l_\Gamma(t) \geq b\}.$$

Given that the system is in stationarity at  $t = 0$ , it is possible to have  $\tau_\Gamma(b) = 0$ . The following lemma provides an estimate on the overflow time  $\tau_\Gamma(b)$ .

*Lemma 4:* Let  $\rho := \lambda/\mu$  and  $\gamma_\Gamma := \lambda_\Gamma/\lambda$ . For any  $T > 0$ , we have

$$\mathbb{P}[\tau_\Gamma(b) < T] \leq 2\lambda_\Gamma T \left( \frac{\rho\gamma_\Gamma}{1 - \rho(1 - \gamma_\Gamma)} \right)^{b-1} + e^{-\lambda_\Gamma T/4}.$$

*Proof:* Given in [12]. ■

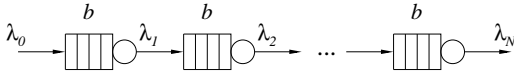


Fig. 1. A multi-hop path consisting of  $N$  nodes. Each node has limited buffer space to store packets.

### III. MOTIVATION: MULTI-HOP ROUTE

In this section we consider a multi-hop route consisting of  $N$  nodes labeled by the natural numbers, as illustrated in Fig. 1. Namely, we examine  $N$  queues of unit capacity in tandem with each queue having a finite buffer capable of storing  $b$  packets. Packets arrive to the first queue according to a Poisson process of rate  $\lambda_0 < 1$ . Service times are exponential with unit mean and independent from the Poisson arrival process. In addition, the service requirements of a single packet are independent in different queues and independent from the requirements of other packets.

We now lower and upper bound the rate  $\lambda_N$  at which packets leave the last node  $N$  as a function of  $N$  and the buffer size  $b$ . Note that  $\lambda_N$  represents the throughput of the route. Lemma 1 allows one to bound the rate at which packets are lost (dropped due to a full buffer) at each node. By letting all nodes prior to node  $i$  have an infinite buffer, it is clear, from Lemma 1, that the loss rate at the  $i$ th node is upper bounded by the stationary loss rate in a finite buffer M/M/1 queue with arrival rate  $\lambda_0$ , given by  $c_b \lambda_0^{b+1}$ , where

$$c_b^{-1} := \sum_{j=0}^b \lambda_0^j$$

is the normalization constant. This follows from the fact that the infinite buffer system is reversible, and, hence, the loss probability is given by  $c_b \lambda_0^b$  [13, Sec. 1.6]. Therefore, the total rate at which packets are lost in the considered finite buffer network is smaller than  $N c_b \lambda_0^{b+1}$ , rendering the following lower bound

$$\lambda_N \geq \lambda_0 - N c_b \lambda_0^{b+1}. \quad (5)$$

In order to establish an upper bound on the throughput, we define the following function

$$f_b(x) := x \left( 1 - (b+1)^{-1} e^{-b/x} \right).$$

According to Lemma 2, quantity  $f_b(x)$  can be interpreted as an upper bound on the maximum throughput of a queue having finite buffer  $b$ , exponential i.i.d. service requirements of unit mean, and offered load  $x < 1$ . Using the fact that the output of queue  $i$  represents the input for the next queue ( $i+1$ ), the monotonicity of  $f_b(x)$  in  $x$  for all  $b$  results in

$$\lambda_N \leq f_b(f_b(\dots f_b(\lambda_0) \dots)),$$

where the function  $f_b(\cdot)$  is iterated  $N$  times. Thus, by the definition of  $f_b(\cdot)$ , it is easy to obtain an upper bound

$$\begin{aligned} \lambda_N &\leq \lambda_0 \prod_{i=1}^N \left( 1 - (b+1)^{-1} e^{-b/\lambda_i} \right) \\ &\leq \lambda_0 \left( 1 - (b+1)^{-1} e^{-b/\lambda_N} \right)^N, \end{aligned} \quad (6)$$

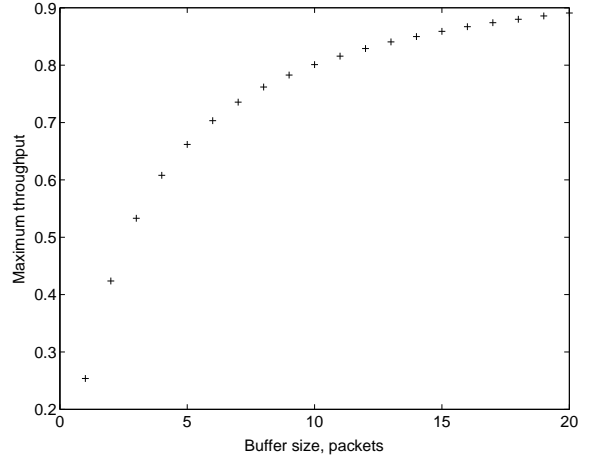


Fig. 2. The maximum throughput of a multi-hop path (see Fig. 1) as a function of the buffer space available at each hop under the hop-by-hop transmission scheme. In this particular example the path consists of  $N = 100$  nodes.

since  $\lambda_i \geq \lambda_{i+1}$  for  $i = 0, \dots, N-1$ . Inequality (6) implies that, in order to keep  $\lambda_N = \Theta(1)$  as  $N \rightarrow \infty$ , the buffer size  $b$  needs to grow at least logarithmically in the number of hops  $N$ . However, (5) indicates that having  $b = O(\log N)$  is enough to sustain constant throughput as the length of the network becomes large.

Next, we turn our attention to the performance of a hop-by-hop transmission scheme [14, p. 507]. The network topology is the same as in the preceding analysis (see Fig. 1). However, a node  $i$  starts packet transmissions only when the buffer at the downstream node can accommodate an incoming packet. The case  $b = 1$  corresponds to the scenario in which a node can hold only a single packet (possibly in service). Packets are added to the first node's buffer as soon as space becomes available. Analytical results for achievable throughput when  $b > 1$  are not available. Results of a numerical experiment provided in Fig. 2 demonstrate the dependency of throughput on the buffer size  $b$ . When  $b = 1$ , the problem of establishing the maximum throughput reduces to finding the current in an asymmetric exclusion process [15]. In particular, for  $b = 1$  it is known [15, Theorem 3.28, p. 272] that the maximum throughput converges to  $1/4$  as  $N \rightarrow \infty$ . However, it should be noted that hop-by-hop schemes in networks with more general topologies are susceptible to deadlocks and mechanisms for their avoidance need to be implemented; refer to [16] and the references therein.

### IV. WIRELESS NETWORK MODEL

We consider a variation of the standard random network topology model [1], [2], [6] where  $N$  nodes are randomly placed in a unit area square (see Fig. 3). More specifically, if  $X_i$  denotes the position of the  $i$ th node, then the variables  $\{X_i\}_{i=1}^N$  are independent and uniformly distributed inside the unit square. Each node is the source of one flow and can serve as the destination for some other flows. Namely, each source node chooses its destination independently and uniformly among the remaining  $(N-1)$  nodes. This results

in distances between sources and destinations that are  $\Theta(1)$  units on average (recall that the asymptotic notation is defined in Appendix A). By a flow, say from node  $i$  to  $j$ , we understand the (infinite) collection of packets generated at  $i$  with destination  $j$ . A node generates packets, independently of other nodes and the network state, according to a Poisson process with rate  $\lambda$  and attempts to inject those packets into the network. It is appropriate to think of packet generation times at node  $i$  as times at which some information becomes available at node  $i$  and needs to be communicated to node  $j$ . At any particular node, all packets have i.i.d. service requirements that are independent from their service requirements at other nodes, arrival times and, in general, the state of the network. The distribution of service is assumed to be exponential, which will be justified in the remainder of this section.

Two models that govern successful packet transmission between a pair of nodes have appeared in the literature [1]: physical and relaxed protocol models. It has been established [1] that these models are equivalent under the following two assumptions: (i) the fading factor in the physical model is greater than 2, and (ii) all nodes use the same power for transmission (i.e., they have a common range). Under the relaxed model, a transmission from node  $i$  to node  $j$  is successful if for some  $\Delta > 0$  and all  $k \neq i, j$

$$d(k, j) \geq (1 + \Delta) d(i, j), \quad (7)$$

where  $d(i, j)$  is the (Euclidian) distance between nodes  $i$  and  $j$ . Typically, models in the literature consider slotted systems. Such an assumption is justified in the presence of a centralized scheduler, and if (7) is satisfied then a packet is successfully transmitted from node  $i$  to node  $j$  with probability 1 in a single time slot. We deviate from this model by assuming that successful transmission from  $i$  to  $j$  is still possible under (7) with the understanding that the amount of time required to successfully transmit a packet is *random* with finite mean. Such an assumption on the model is justified by the following arguments:

- Centralized scheduling is unlikely to be feasible in large ad-hoc wireless networks. Hence, a slotted system is likely to be replaced with asynchronous transmission that is a result of distributed scheduling.
- The nodes in the network could have limited power supply and the physical channel might be highly variable. Thus, in certain time intervals only transmission at reduced rates (or no transmission at all) is feasible.
- The layered architecture of current networks contributes to the randomness of transmission times. For example, transmission errors on the data-link layer as well as possible collisions on the MAC layer contribute to such randomness.

As a first order approximation we assume that, under (7), transmission times are exponentially distributed with mean that is proportional to the packet size in bits. The exact parameter of the distribution depends on the details of the transmission model and is independent of  $N$ . Evaluation of this parameter is beyond the scope of this paper since we focus only on the qualitative network behavior as a function of its size.

Under the specified algorithm, we denote by  $\vartheta_N(b)$  the long-term throughput that each source-destination pair can achieve when there are  $N$  nodes in the network, each having storage space of size  $b$  packets. The units of  $\vartheta_N(b)$  are *bits/s*. Each source-destination pair is assigned a route and an end-to-end scheme is defined by packets being transmitted along this route without taking into account whether there is available buffer space at the next node. In other words, packets can be lost due to a full buffer at an intermediate node.

This section is concluded with a remark on packet sizes. We emphasize that  $\vartheta_N(b)$  is a function of the number of packets that can be stored at each node, not the number of bits. As the number of nodes in the network increases the packet size might need to increase in order to keep the payload size constant. This stems from the fact that one needs  $\log N$  bits to identify the destination node, e.g., if each packet carries its node's address then each packet has to be at least  $\log N$  bits in size. Our main result in the next section, Theorem 1, takes into account that the packet size might be an increasing function of the network size.

## V. SCALABILITY

This section contains the main results of the paper that characterize the memory scalability of large wireless networks in the *data plane*. In particular, we study the throughput dependency on the network size in the presence of limited buffer space at individual nodes. Let  $s$  denote the packet size in bits.

Our first result indicates that achieving  $\Theta(1/\sqrt{N})$  throughput per source-destination pair is not feasible when nodes have a fixed amount of buffer space. Informally, achieving  $\Theta(1/\sqrt{N})$  throughput requires that only short hops are used [1], [5] resulting in packets traversing a large ( $\Theta(\sqrt{N})$ ) number of hops. Hence, given that losses are inevitable (see Lemma 2), buffer sizes need to increase with route lengths, i.e., the network size. In the following proposition and Proposition 2 we only assume that the packet generation times are stationary and ergodic (see Lemma 2). However, in our main Theorem 1 we assume Poisson generation of packets, as specified by our model in Section IV.

*Proposition 1:* Assume that packet transmission times are exponential random variables with mean  $s/\mu$ . If each source generates packets at rate not higher than  $c/(s\sqrt{N})$  (packets/s), then, in order to achieve  $\varepsilon/\sqrt{N}$  throughput (bits/s) per source-destination pair ( $c \geq \varepsilon$ ), buffer space at  $\Theta(N)$  nodes need to scale as  $\Omega(\log N)$  in  $N$  almost surely (a.s.).

*Proof:* We start by establishing that, if all source-destination pairs achieve at least  $\varepsilon/\sqrt{N}$  bits/s throughput for some  $\varepsilon > 0$ , then there exists  $\delta > 0$  such that at least  $\delta N$  nodes have utilization higher than  $\delta$  (for all  $N$  large enough). To this end, if at most  $\delta N$ ,  $\delta > 0$ , nodes have utilization higher than  $\delta$ , then the long-term rate at which the network processes packets is bounded from above with

$$((1 - \delta)N \cdot \delta + \delta N \cdot 1) \mu/s = (2\delta - \delta^2) \mu N/s \quad \text{packets/s,}$$

or equivalently  $(2\delta - \delta^2) \mu N$  bits/s. On the other hand, if all source-destination pairs achieve at least  $\varepsilon/\sqrt{N}$  bits/s throughput for some  $\varepsilon > 0$ , then the total rate at which packets are

injected in the network must be at least  $\varepsilon\sqrt{N}/s$ . This, together with the fact that for a sufficiently small  $\zeta > 0$  at least a quarter of the flows traverse at least  $\zeta\sqrt{N}$  hops a.s. (see Lemma 7 in Appendix C), implies the total network offered load is at least

$$\frac{\varepsilon\sqrt{N}}{s} \cdot \frac{1}{4}\zeta\sqrt{N} = \frac{\varepsilon\zeta N}{4s} \text{ packets/s a.s.,}$$

or equivalently  $\varepsilon\zeta N/4$  bits/s. Since the amount of work processed in the network must be at least equal to the amount of work generated by packets delivered to their destinations, one has

$$(2\delta - \delta^2)\mu \geq \varepsilon\zeta/4.$$

Hence, the existence of  $\varepsilon > 0$  implies  $\delta > 0$  a.s..

Next, we argue about the buffer space. Let  $g_b(x)$  be the lower bound on the loss rate (packets/s) at a node with utilization  $x$  and buffer space  $b$  packets. By Lemma 2, one has

$$g_b(x) = \frac{\mu x}{s}(b+1)^{-1}e^{-b/x},$$

since utilization  $x$  implies the arrival rate of  $\mu x/s$ . Given the preceding discussion, the total network loss rate  $\kappa_N$  satisfies a.s.

$$\kappa_N \geq \delta N g_b(\delta) = \frac{\mu\delta^2}{s}N(b+1)^{-1}e^{-b/\delta} \text{ packets/s.} \quad (8)$$

The total number of packets delivered to destinations is equal to the number of packets injected into the network minus the number of lost packets, i.e.,

$$\frac{c}{s\sqrt{N}}N - \kappa_N \geq \frac{\varepsilon}{s\sqrt{N}}N.$$

The preceding equation and (8) yield an implicit estimate on  $b$

$$(b+1)^{-1}e^{-b/\delta} \leq \frac{c-\varepsilon}{\mu\delta^2\sqrt{N}}$$

from which it is clear that  $b = \Omega(\log N)$  a.s. ■

In the rest for the paper we construct a protocol that is capable of achieving a high throughput, proved in our main Theorem 1, with buffers of constant size. The key component of our protocol is a local buffer coordination that we motivate in the upcoming Proposition 2.

For constructing our protocol, it is sufficient to consider the parameterized scheme introduced in [2]. Namely, the unit square is divided into  $a_N^{-1}$  squares of size  $\sqrt{a_N} \times \sqrt{a_N}$  (for convenience we assume that  $1/\sqrt{a_N}$  is an integer so that the squares cover the unit square exactly). See Fig. 3 for an illustration. The created squares are termed cells. A packet is delivered from its source to the destination by relaying it between cells. A packet can be sent from a cell only to one of its four neighboring cells. No multi-cell hops are allowed. Given this cell based scheme, the next lemma states that each cell in the network can transmit packets to its neighbors at a fixed rate independent of the state of other cells. Recall that  $s$  denotes the packet size (in bits).

*Lemma 5:* All cells in the network can transmit packets simultaneously to one of their neighboring cells with transmission times being independent exponential random variables with mean  $s/\mu$  that are independent of  $N$ , for some  $\mu \in (0, \infty)$ .

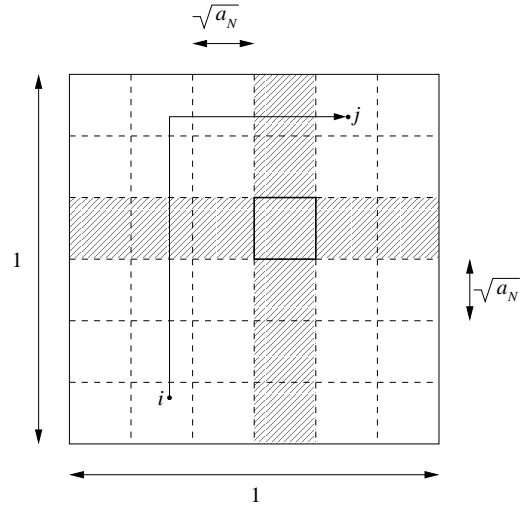


Fig. 3. The transmission algorithm is parameterized by  $a_N$  that defines the cell size. Only transmissions between neighboring cells are allowed.

*Remark 2:* The parameter  $\mu$  can be interpreted as a per-bit service rate.

*Proof:* The proof follows from the model described in Section IV and the fact that only a finite number of cells interfere with a given cell under the protocol model [2, Lemma 2]. ■

Next, we discuss an issue that arises only due to the finiteness of the buffer space in the network. The network performance depends on whether a buffer coordination algorithm is deployed. Such an algorithm operates in each cell and its goal is the efficient use of buffer space in cells. To motivate a potential need for such an algorithm, we consider the following example. Suppose that a node in an adjacent cell has a packet that requires relay through a given cell. The node that has the packet can, in principle, forward it to any of the nodes in the cell. However, if the next relay node is chosen without consideration of buffer contents, an unnecessary packet loss can occur. Namely, a packet could be forwarded to a node that can not accommodate any more packets due to a full buffer. A buffer coordination algorithm assists nodes in adjacent cells in forwarding packets to nodes that have available buffer space, whenever possible. In that case a packet loss occurs only if no node in a cell has buffer space available. The design of such an algorithm is beyond the scope of this paper, given our focus on the fundamental qualitative behavior of the network as a function of its size. However, we point out that it could be designed using a token approach (when the whole cell also implements the FCFS scheduling policy). In this case the amount of memory required to implement the buffer coordination protocol at each node can be limited to the size of a single packet. More sophisticated approaches are possible as well. Although we consider only a protocol that fully utilizes the buffer at each cell, we note that results from the load balancing literature suggest that protocols which examine the buffer contents of a small number of nodes before forwarding/storing a packet are expected to perform reasonably well.

The following proposition characterizes the network performance with no buffer coordination. In contrast to the case in which nodes have infinite buffer space, it is important to further specify the routing policy between two cells. In particular, we assume that when a packet is transmitted between neighboring cells, it is forwarded to a random node in the destination cell. In [1] (see also [2]) it was shown that under the straight line routing scheme and centralized scheduling it is possible to achieve  $\Theta(1/\sqrt{N \log N})$  throughput for every source-destination pair, i.e.,

$$\lim_{N \rightarrow \infty} \mathbb{P}[\vartheta(\infty) = c/\sqrt{N \log N} \text{ is feasible}] = 1.$$

In particular, one needs to choose  $a_N = \Theta(\log N/N)$  [2]. The following proposition motivates the need for a buffer coordination protocol within cells. Namely, without such a protocol, otherwise feasible  $\Theta(1/\sqrt{N \log N})$  throughput can not be achieved with limited buffer space at nodes.

*Proposition 2:* Assume no buffer coordination and  $a_N = \Theta(\log N/N)$ . If each source generates packets at rate not higher than  $c/(s\sqrt{N \log N})$ , then in order to achieve  $\varepsilon/\sqrt{N \log N}$  bits/s throughput per source-destination pair ( $c \geq \varepsilon$ ) the buffer spaces at  $\Theta(N)$  nodes need to scale as  $\Omega(\log N/\log \log N)$ , i.e., for  $b = o(\log N/\log \log N)$  as  $N \rightarrow \infty$  and any  $c > 0$

$$\lim_{N \rightarrow \infty} \mathbb{P}[\vartheta_N(b) = c/\sqrt{N \log N} \text{ is feasible}] = 0.$$

*Remark 3:* Comparing Propositions 1 and 2 yields that without a buffer coordination protocol a  $\sqrt{\log N}$  reduction on rate results only in  $\log \log N$  reduction in the required buffer size.

*Proof:* The proof is very similar to that of Proposition 1. However, the focus of the analysis is on cells rather than nodes. Using the same arguments as in the proof of Proposition 1, it is straightforward to argue that if all source-destination pairs achieve  $\varepsilon/\sqrt{N \log N}$  bits/s throughput for some  $\varepsilon > 0$ , then there exists  $\delta > 0$  such that the event  $\mathcal{G} := \{\text{at least } \delta a_N^{-1} = \delta N/\log N \text{ cells have utilization higher than } \delta\}$  obeys

$$\lim_{N \rightarrow \infty} \mathbb{P}[\mathcal{G}] = 1.$$

The choice of  $a_N$  yields (see Lemma 8 in Appendix C) that there exists  $\gamma < \infty$  such that

$$\lim_{N \rightarrow \infty} \mathbb{P}[\mathcal{B}] = 1,$$

where event  $\mathcal{B} := \{\text{each cell contains at most } \gamma \log N \text{ nodes}\}$ . Without a buffer coordination algorithm in place, the loss rate in a cell with at most  $\gamma \log N$  nodes and utilization greater than  $x$ , using Lemma 3, is lower bounded by

$$g_b(x) = \frac{\mu x}{s} (\gamma \log N)^{-b} (b+1)^{-1} e^{-b/x} \text{ packets/s,}$$

where  $b$  is the buffer space at each node and the number of partitions is in fact equal to the number of nodes. Furthermore, the following lower bound holds for the total loss rate (packets/s) in the network on event  $\mathcal{G} \cap \mathcal{B}$ :

$$\kappa_N \geq \delta a_N^{-1} g_b(\delta) = \frac{\mu \delta^2 N}{s \log N} (\gamma \log N)^{-b} (b+1)^{-1} e^{-b/\delta}. \quad (9)$$

The total number of packets delivered to their destinations is equal to the number of packets injected into the network minus the number of lost packets, i.e.,

$$\frac{c}{s\sqrt{N \log N}} N - \kappa_N \geq \frac{\varepsilon}{s\sqrt{N \log N}} N.$$

Hence, the preceding inequality and (9) imply that on event  $\mathcal{G} \cap \mathcal{B}$  we have

$$(\gamma \log N)^{-b} (b+1)^{-1} e^{-b/\delta} \leq \frac{c - \varepsilon}{\mu \delta^2 \sqrt{N \log N}}.$$

Rewriting the left-hand side of the preceding inequality as

$$e^{-b \log \log N - \log(b+1) - b(1/\delta + \log \gamma)}$$

and the right-hand side as

$$\frac{c - \varepsilon}{\mu \delta^2} e^{-\frac{1}{2} \log N - \frac{1}{2} \log \log N},$$

it is easy to conclude that  $b$  needs to scale at least as  $\Omega(\log N/\log \log N)$ . ■

The next theorem is the main result of the paper. It states that with a buffer coordination algorithm in place, a small reduction in the throughput from  $\Theta(1/\sqrt{N})$  to  $\Theta(1/\sqrt{N \log N})$  allows for a buffer scalable algorithm. The result follows from the fact that the cell size which allows for  $\Theta(1/\sqrt{N \log N})$  is sufficiently large to contain enough nodes, and, thus, enough buffer space that ensures low loss rates when the buffer is utilized efficiently. In particular, each cell contains buffer space capable of storing  $\Theta(\log N)$  packets. Recall that the units of  $\vartheta_N(b)$  are bits/s.

*Theorem 1:* There exists an end-to-end buffer-scalable algorithm, i.e., there exist  $b^*$  and  $c > 0$  such that

$$\lim_{N \rightarrow \infty} \mathbb{P}[\vartheta_N(b^*) = c/\sqrt{N \log N} \text{ is feasible}] = 1.$$

*Remark 4:* It is apparent from the proof that in order to achieve  $\Theta(1/\sqrt{N \log N})$  bit/s throughput it is sufficient if each node contributes a single packet buffer space for storing forwarded packets.

*Remark 5:* The model can be somewhat relaxed. In particular, each node does not need to communicate with only one other node. Other communication patterns are possible and the theorem remains valid as long as the load on each cell remains bounded.

*Proof:* To prove the theorem it suffices to construct a protocol that achieves  $\Theta(1/\sqrt{N \log N})$  throughput per source-destination pair given that each node has a buffer that can accommodate only  $b^*$  packets. We stress that the algorithm considered in the proof is not the most efficient one, and it is devised to allow for analytical tractability to demonstrate the desired qualitative network behavior.

The algorithm operates as follows:

- *Routing policy.* Packets are forwarded from their sources to destinations along cells using the column-first routing [17]. See Fig. 3 for an example of how packets from node  $i$  are routed to node  $j$ . Note that under such a routing scheme the number of cells each packet traverses is at most  $2/\sqrt{a_N}$ .



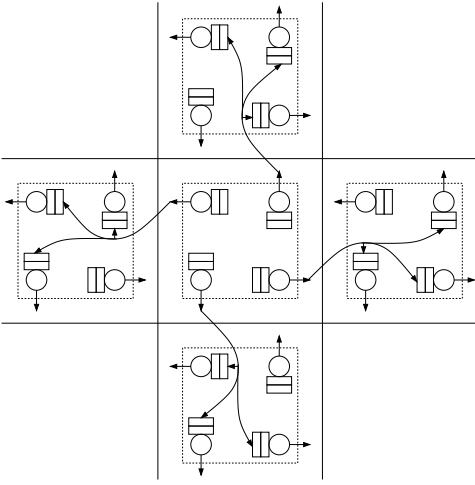


Fig. 4. Under the considered algorithm each cell can be modeled as a set of four finite buffer queues. Each queue forwards packets to only one of the neighboring cells. There is no bandwidth and buffer space sharing (multiplexing) between the four queues. The queues can forward packets to one of the three queues in the neighboring cell (or the final destination) depending on packet's route.

- *Buffer management policy.* Buffer space at each node is divided into two logical parts: one packet space is reserved for storing forwarded packets while the rest of the buffer ( $(b^* - 1)$  packet spaces) is designated for storing the node's own packets. The buffer space designated for forwarded packets at all nodes in a given cell is managed by that cell's buffer coordination algorithm. The buffer coordination algorithm creates four virtual buffers of equal size (up to a single packet space). In each one of these four virtual buffers, packets destined for the same cell are stored, e.g., in the first virtual buffer packets to be forwarded to the up cell are stored, in the second virtual buffer packets to be forwarded to the right cell are stored, etc. All packets to be forwarded to the same cell are transmitted in a FCFS manner.
- *Bandwidth sharing policy.* Bandwidth available for wireless communication is divided into four equal parts (using one of the available technologies, e.g., FDMA). Each quarter of the bandwidth can be used throughout the network (by all cells) to forward (transmit) packets only in one direction (up, right, down, or left). Such a policy corresponds to a scheme in which each cell consists of 4 virtual servers, each forwarding packets to one of the neighboring cells; see Fig. 4 for an illustration. Note that according to Lemma 5 each virtual server is able to transmit packets at a fixed non-zero rate.
- *Frame based operation.* Time is divided into frames of length  $T$  time units. At the beginning of each frame the network is initialized in a random state, as described below (see Fig. 5). During a frame packets are forwarded according to the above rules until the end of the frame is reached or a discard event occurs. If the end of the frame is reached, all packets that remain in the network at that moment are dropped. If the discard event occurs before the end of the frame, all packets in the network are

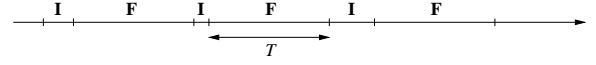


Fig. 5. The algorithm uses frames of fixed length  $T$  time units to transport packets. Each frame (F) is preceded by an initialization period (I).

dropped and no new packets are accepted to the network until the beginning of the next frame. The discard event occurs when in one of the cells a packet arrival occurs and the newly arrived packet can not be accommodated in the buffer according to the buffer management policy, i.e., the packet has to be dropped. For example, if a node has  $(b^* - 1)$  of its own packets in the buffer and it generates a new packet, then the discard event occurs.

- *Initialization.* The random state is the steady state of the network with the same topology and traffic flows but with each node having an infinite buffer. A probabilistic description of such a state is well known since the infinite buffer network admits a product-form solution. The packets with which the network is initialized contain dummy payloads; however, they are treated by the network in the same way as regular packets. Note that due to the initialization procedure the frame can be discarded right at its beginning. Full details of the initialization procedure are outlined in Appendix B. Here we just point out that the time  $T_I$  required to initialize the network satisfies  $\mathbb{E}T_I = o(sN^3)$  as  $N \rightarrow \infty$ .

Now, the remainder of the proof is divided into three parts.

*Part I.* We estimate the load and amount of buffer space available in each cell. To complete the description of the scheme, the size of cells  $a_N$  needs to be specified. We choose  $a_N$  to be the smallest function of  $N$  such that each cell contains at least  $\beta \log N$  nodes for some  $\beta > 0$ . Let  $k_i$  denote the number of nodes in the  $i$ th cell. By Lemma 8 of Appendix C, for every fixed  $\beta > 0$  there exists  $a(\beta) > 1$  such that if  $a_N = a(\beta) \log N/N$ , then

$$\lim_{N \rightarrow \infty} \mathbb{P}[k^\downarrow > \beta \log N] = 1, \quad (10)$$

where  $k^\downarrow := \min_{1 \leq i \leq a_N^{-1}} k_i$  is the minimum number of nodes contained in a cell. Lemma 8 also states that there exists a finite constant  $\gamma > \beta$  (independent of  $N$ ) such that

$$\lim_{N \rightarrow \infty} \mathbb{P}[k^\uparrow < \gamma \log N] = 1, \quad (11)$$

where  $k^\uparrow := \max_{1 \leq i \leq a_N^{-1}} k_i$  is the maximum of the number of nodes contained in a cell. Let each node generate packets at Poisson rate  $\lambda = c/(s\sqrt{N \log N})$ , where  $c > 0$  is a constant independent of  $N$ , and  $s$  is the packet size that can depend on  $N$ . Lemma 9 of Appendix C allows for an estimate of a potential load  $\rho_i$  of each cell, i.e., the rate at which packets with cell  $i$  being in their route are injected in the network. Formally,

$$\rho_i = \frac{\lambda s}{\mu} \sum_j \mathbf{1}_{\{i \in \mathbf{r}_j\}},$$

where  $\mathbf{r}_j$  is a collection of cells a packet originating in the  $j$ th cell needs to traverse to reach its destination, and  $\mu$  is the rate at which packets are served (transmitted to the next

cell). Equivalently,  $\rho_i = \lambda s K_i / \mu$ , where  $K_i$  is the number of lines that cross the  $i$ th cell when all sources are connected to their destinations using the column-first rule (see Fig. 3 for an example). Letting  $\rho^\dagger$  denote the maximum potential load

$$\rho^\dagger := \max_{1 \leq i \leq a_N^{-1}} \rho_i \leq \frac{\lambda s}{\mu} \max_{1 \leq i \leq a_N^{-1}} K_i,$$

and invoking Lemma 9 yield

$$\begin{aligned} \lim_{N \rightarrow \infty} \mathbb{P}[\rho^\dagger \leq 5\sqrt{a_N} N \lambda s / \mu] \\ = \lim_{N \rightarrow \infty} \mathbb{P}[\rho^\dagger \leq 5c\sqrt{a(\beta)}/\mu] = 1, \end{aligned} \quad (12)$$

since  $a_N = \alpha(\beta) \log N / N$  and  $\lambda = c/(s\sqrt{N \log N})$ . Now, define the following event

$$\mathcal{X} := \left\{ \beta \log N < k^\downarrow \leq k^\uparrow < \gamma \log N, \rho^\dagger \leq \frac{5c\sqrt{a(\beta)}}{\mu} \right\},$$

and the corresponding conditional probability measure

$$\mathbb{P}_{\mathcal{X}}[\cdot] := \mathbb{P}[\cdot | \mathcal{X}].$$

Informally, event  $\mathcal{X}$  indicates that each cell in the network is not overloaded and there exists enough buffer space in each cell. From limits (10), (11) and (12) it follows that

$$\lim_{N \rightarrow \infty} \mathbb{P}[\mathcal{X}] = 1. \quad (13)$$

Equation (13) allows one to consider only networks that are not overloaded and have ‘‘sufficient’’ buffer space.

*Part II.* Now we estimate the probability of an arbitrary frame being discarded. Let  $v_i(t)$ ,  $1 \leq i \leq N$ , be the number of packets at time  $t$  in the  $i$ th node buffer that have node  $i$  as their source. Clearly, we have  $v_i(t) \leq b^* - 1$  since each node can accommodate at most  $(b^* - 1)$  of its own packets. Furthermore, let  $u_{ij}(t)$ ,  $1 \leq i \leq a_N^{-1}$ ,  $1 \leq j \leq 4$ , be the number of packets at time  $t$  in the  $j$ th virtual buffer of the  $i$ th cell that originated in other cells. Without loss of generality consider the frame over the time interval  $[0, T]$  and define

$$\begin{aligned} \mathcal{E} := \{v_i(t) < b^* - 1, \forall i, \forall t \in [0, T]\} \\ \cap \{u_{ij}(t) < b, \forall i, \forall j, \forall t \in [0, T]\}, \end{aligned}$$

where  $b := \lfloor \beta \log N / 4 \rfloor$ . An important observation is that on event  $\mathcal{E}$  no packet losses occur in the original network during the time interval  $[0, T]$ , which yields that the frame over the interval  $[0, T]$  is not discarded. Equivalently,  $\mathbb{P}_{\mathcal{X}}[\mathcal{E}]$  is the probability of a frame not being discarded. The following lemma, whose proof can be found in Appendix D, provides an estimate of this probability of interest. The dependency of  $\mathbb{P}_{\mathcal{X}}[\mathcal{E}]$  on parameter  $\beta$  (see the definition of event  $\mathcal{X}$ ) is through quantity  $b$ .

*Lemma 6:* If  $\rho := 5c\sqrt{\alpha(\beta)}/\mu < 1$ , then

$$\begin{aligned} \mathbb{P}_{\mathcal{X}}[\mathcal{E}] \geq 1 - \left(1 + \frac{2\mu T}{s}\right) \left[ N \left(\frac{\lambda s / \mu}{1 - \rho}\right)^{b^* - 1} - 4a_N^{-1} \rho^b \right] \\ - (N + 4a_N^{-1}) e^{-\lambda T / 4}. \end{aligned}$$

*Part III.* Next we consider an arbitrary packet in the original finite buffer network and estimate the conditional probability

$p$ , given event  $\mathcal{X}$ , of the packet being delivered to its destination. Let  $D$  be the total amount of time the packet spends in the network given that it is eventually delivered to its destination. On event  $\mathcal{X}$ , the quantity  $D$  can be upper bounded as follows

$$D \leq \sum_{i=1}^{2b^* \gamma \log N / \sqrt{a_N}} S_i,$$

where  $\{S_i\}_{i \geq 1}$  is a sequence of packet service times, i.e., an i.i.d. sequence of exponential random variables with mean  $s/\mu$ . The bound is due to the finiteness of buffers throughout the network, i.e., the fact that only a limited number of packets can be served between the considered packet’s arrivals and departures. On event  $\mathcal{X}$  the amount of buffer space available at each cell is upper bounded by  $b^* \gamma \log N$  and each packet traverses at most  $2/\sqrt{a_N}$  cells. Then the conditional (on  $\mathcal{X}$ ) probability  $p$  that an arbitrary packet is delivered to its destination is lower bounded by

$$p \geq q \mathbb{P}_{\mathcal{X}}[\mathcal{E}, U > D] \geq q(\mathbb{P}_{\mathcal{X}}[\mathcal{E}] - \mathbb{P}_{\mathcal{X}}[U \leq D]),$$

where  $U$  is a uniformly distributed random variable on  $[0, T]$  and  $q := T/(T + \mathbb{E}T_I)$ . Effectively, the bound states that if the frame during which the packet is generated is not discarded and the packet generation time is not close to the end of the frame, then the packet is delivered successfully. The pre-factor  $q$  is due to the fact that packets do not reach their destinations if they are generated during the initialization period. The random variable  $U$  represents the amount of time between packet generation and the end of the current frame. The uniform distribution of  $U$  yields an easy bound on the second term in the preceding equation

$$\mathbb{P}_{\mathcal{X}}[U \leq D] \leq \frac{\mathbb{E}D}{T} = \frac{2sb^* \gamma \log N}{\mu T \sqrt{a_N}}.$$

The last inequality and Lemma 6 render

$$\begin{aligned} 1 - \frac{p}{q} \leq \left(1 + \frac{2\mu T}{s}\right) \left[ N \left(\frac{\lambda s / \mu}{1 - \rho}\right)^{b^* - 1} + 4a_N^{-1} \rho^b \right] \\ + (N + 4a_N^{-1}) e^{-\lambda T / 4} + \frac{2sb^* \gamma \log N}{\mu T \sqrt{a_N}}. \end{aligned} \quad (14)$$

The preceding equation highlights the trade-off present in selecting the frame length  $T$ . The first term on the right-hand side is linear in  $T$  while the last one is inversely proportion to  $T$ . In other word, the larger  $T$ , the higher probability of a frame being discarded; however, small  $T$  limits the ability of packets to reach their destinations by the end of the frame. Setting  $T = sN^3$  results in  $q \rightarrow 1$  as  $N \rightarrow \infty$ . By letting  $b^*$  be sufficiently large (yet finite) and  $c$  be sufficiently small, we have that, as  $N \rightarrow \infty$ ,

$$\frac{TN}{s} \left(\frac{\lambda / \mu}{1 - \rho}\right)^{b^* - 1} = \frac{TN}{s} \left(\frac{c/(\mu(1 - \rho))}{\sqrt{N \log N}}\right)^{b^* - 1} = o(1),$$

$$\frac{T \rho^b}{s a_N} = \frac{TN}{\alpha(\beta) s \log N} \left(\frac{5c\sqrt{\alpha(\beta)}}{\mu}\right)^{\lfloor \frac{\beta \log N}{4} \rfloor} = o(1),$$

$$(N + a_N^{-1}) e^{-\lambda T / 4} = \left(N + \frac{N}{\log N}\right) e^{-\frac{cT}{4s\sqrt{N \log N}}} = o(1),$$

$$\frac{s \log N}{T \sqrt{a_N}} = \frac{s \sqrt{N \log N}}{T \sqrt{\alpha(\beta)}} = o(1).$$

Combining the above estimates with (14) it is easy to conclude that the conditional probability of the packet not being delivered satisfies, as  $N \rightarrow \infty$ ,

$$1 - p = o(1).$$

Finally, the proof of the theorem is completed by combining the preceding estimate on  $p$ ,  $\lambda = c/(s\sqrt{N \log N})$  and (13). ■

We conclude this section with an observation that when  $a_N$  is chosen to be larger than  $\Theta(\log N/N)$  the scheme remains buffer scalable, albeit it achieves lower throughput [2]. The buffer scalability follows from the fact that the larger the cells, the more buffer space is available at each cell, and, hence, one can expect a lower probability of loss.

## VI. CONCLUDING REMARKS

Large and rapidly increasing networking infrastructures place network scalability as the central problem in designing next generation communication protocols and architectures. While this problem was addressed by practitioners and system engineers, there has been, in particular in the context of finite buffers, little attempt to establish its mathematical foundations. The need for theoretical investigation is even more necessary since conducting experimental studies on large-scale networks is very expensive or practically infeasible.

A straightforward, albeit very inefficient, way to alleviate the issue of scalability is to build nearly fully connected networks where each node is only a few hops away from any other node. Hence, the protocols designed for small networks are likely to extend well to this large highly connected network. However, this is not even an option for wireless ad-hoc networks since achieving theoretical capacity bounds requires that packets traverse long routes. Even the long routes would not be a problem in a world of perfect synchronization without randomness. However, as we argued in the paper, packet transmission times are inherently random due to the multitude of effects, ranging from the physical, data-link to MAC layer. This leads to the strong dependency between the loss probability on each hop and the network size. Namely, the longer the route, the smaller losses on each hop should be in order to maintain constant end-to-end performance.

In the context of static wireless ad-hoc networks, we demonstrate that the network can not operate at its capacity limit while maintaining a constant buffer space in each node. However, maybe somewhat surprisingly, we construct a provably scalable protocol that obtains just a slightly smaller throughput with fixed buffers, capable of storing only a few packets. This understanding is especially important for wireless sensor networks that operate with very limited resources.

Scalability of protocols can be studied in mobile wireless networks [18] as well. Such networks can potentially achieve much higher throughput of  $\Theta(1)$  by exploiting node mobility [18]. The key idea for increasing the throughput is to limit the number of hops packets traverse [18]; see also [2], [19] for studies of throughput-delay trade-off in networks with mobility. With finite buffers, overflows can occur not only

because of the random service times, as describe in this paper, but also because of the random mobility of nodes. Although details differ depending on the mobility model, the type of algorithm studied in [18] appears not to be buffer scalable since it is expected that each node needs to have buffer space of size  $\Theta(N)$  packets. By considering the family of algorithms devised in [2], [3], [4] it is possible to determine the throughput that can be sustained with limited resources at each node.

## ACKNOWLEDGEMENT

The authors would like to thank Nick Maxemchuk for pointing out reference [16].

## APPENDIX

### A. Asymptotic notation

Throughout the paper we use standard notational conventions. For two nonnegative functions  $g(\cdot)$  and  $f(\cdot)$ : (i)  $g(n) = \Theta(f(n))$  means that there exist positive constants  $c_1, c_2$  and  $n_0$  such that  $c_1 f(n) \leq g(n) \leq c_2 f(n)$  for all  $n \geq n_0$ , (ii)  $g(n) = O(f(n))$  means that there exist positive constants  $c$  and  $n_0$  such that  $g(n) \leq c f(n)$  for all  $n \geq n_0$ , (iii)  $g(n) = \Omega(f(n))$  means that there exist positive constants  $c$  and  $n_0$  such that  $c f(n) \leq g(n)$  for all  $n \geq n_0$ .

### B. Initialization procedure

In this section we discuss an initialization procedure for the transmission scheme described in Section V. We remark that there is no unique procedure that achieves the desired goal. Exact details of different procedures depend on the system architecture and nodes capabilities.

Since the network needs to be initialized in the steady-state of the infinite buffer network that admits a product-form solution, a sequential procedure that initializes one queue at a time suffices. Therefore, we focus on a single virtual queue (i.e., one of the four virtual queues in a cell – see Fig. 4).

We assume that each packet contains its destination address so that nodes (cells) do not need to waste memory on maintaining routing tables. However, we do assume that nodes have access to a source of randomness. Each node in the network can be identified with a number  $1 \leq m \leq N$ . In addition, each cell can be identified by a pair  $(i, j)$ ,  $1 \leq i, j \leq 1/\sqrt{a_N}$ , with  $i$  increasing from left to right and  $j$  increasing from bottom to top.

The first step is to establish the utilization of the queue, i.e., the number of flows served by the queue. A node in the cell of interest broadcasts its address  $(i, j)$  and a number between 1 and 4 that identifies the queue of interest. After that, all nodes sequentially indicate, using labels  $m$  to establish order, whether their flows utilize the queue. Note that for any particular flow, defined by its source and destination coordinates, it is easy for a queue to compute whether this flow utilizes it or not. Therefore, simple counting of the number of flows that traverse a particular node establishes the utilization since all nodes generate packets at the same Poisson rate. This step of the procedure takes  $O(N)$  time units to execute.

The second step is to generate a random number of dummy packets and place them at right parts of the virtual buffer. Dummy packets are generated sequentially. Using a source of randomness, a node in the cell generates Bernoulli (0-1) random variables with its parameter being the queue's utilization until 0 occurs. If a 1 is generated, one of the nodes that have packets forwarded by the queue is chosen at random and that queue's destination is made the destination of the dummy packets. The dummy packet is stored in the virtual buffer according to the buffer management policy. If the dummy packet can not be stored, then the procedure is completed with the frame being discarded. Using a random walk on the nodes, a random node that uses the queue for forwarding can be obtained in expected time  $O(\sqrt{N})$ , and, thus, the second step can be executed in  $O(sN^{3/2})$  time units.

Upon combining the two steps, it is apparent that the expected time to initialize the network is asymptotically  $O(sN^{5/2})$ .

### C. Auxiliary results

This subsection contains three results on the wireless network model described in Section IV. The first one provides a lower bound on the number of hops in the case when all source-destination pairs are able to achieve the maximal possible throughput  $\Theta(1/\sqrt{N})$ . Effectively, the lemma is a consequence of the relaxed protocol model defined by (7).

*Lemma 7:* If each source-destination pair achieves  $c/\sqrt{N}$  bits/s throughput (with packets of size  $s$ ), then at least 1/4 of all flows traverse at least  $\delta\sqrt{N}$  hops a.s. for some sufficiently small  $\delta > 0$ .

*Proof:* Given in [12]. ■

The next lemma estimates the number of nodes in cells as a function of the cell size. Consider a unit square partitioned into cells of equal area  $a_N$ , as in Figure 3, and recall that  $(\sqrt{a_N})^{-1}$  is assumed to be an integer. Nodes are uniformly distributed in the unit square with  $k_i$  being the number of nodes in the  $i$ th cell (where the labeling of cells is arbitrary).

*Lemma 8:* Let  $a_N > a \log N/N$ . If  $a > \delta$  and

$$a - \delta(1 + \log(a/\delta)) > 1,$$

then

$$\lim_{N \rightarrow \infty} \mathbb{P} \left[ \min_{1 \leq i \leq a_N^{-1}} k_i > \delta \log N \right] = 1.$$

On the other hand, if  $a_N < a \log N/N$ , then there exists  $\gamma < \infty$  such that

$$\lim_{N \rightarrow \infty} \mathbb{P} \left[ \max_{1 \leq i \leq a_N^{-1}} k_i < \gamma \log N \right] = 1.$$

*Proof:* Given in [12]. ■

The last result of this subsection provides an estimate of cell loads under the column-first routing. Consider the unit square partitioned into square cells of size  $a_N$  as in Fig. 3. Let each source-destination pair be connected with a line that stems from the column-first routing, and let  $K_i$  be the number of such lines that cross the  $i$ th cell. Namely, if  $X_i = (x_i, y_i)$  and  $X_j = (x_j, y_j)$  are the coordinates of a source and

destination, respectively, then the line connecting those two nodes is defined by three points:  $(x_i, y_i)$ ,  $(x_i, y_j)$ , and  $(x_j, y_j)$ ; e.g., see Fig. 3. The following lemma bounds the number of lines connecting source-destination pairs that cross a single cell. A similar result was established in [2] for a unit torus and the straight-line routing scheme.

*Lemma 9:* If  $a_N N > 1$  and  $a_N = o(1)$  as  $N \rightarrow \infty$ , then

$$\lim_{N \rightarrow \infty} \mathbb{P} \left[ \max_{1 \leq i \leq a_N^{-1}} K_i > 5\sqrt{a_N N} \right] = 0.$$

*Proof:* Given in [12]. ■

### D. Proof of Lemma 6

Consider a network of the same topology and traffic flows as the original one, however, with virtual buffers in each cell having infinite buffer space. As before, the buffer policy is FCFS within each virtual buffer. The network is in stationarity at time  $t = 0$ . Given this scheduling policy and the Poisson assumption on times when packets enter the network, we observe that the network of virtual buffers is in fact a multi-class Jackson network (also known as a Kelly network) that admits a product-form solution [20, p. 101], [21, p. 123]. A customer (packet) class corresponds to a particular flow in the network. In this new infinite buffer network, packets belonging to any flow and departing any cell form a Poisson stream.

Define quantities  $v_i^\infty(t)$  and  $u_{ij}^\infty(t)$  to correspond to variables  $v_i(t)$  and  $u_{ij}(t)$ , respectively, that are previously defined for the finite buffer network. Namely, let  $v_i^\infty(t)$ ,  $1 \leq i \leq N$ , be the number of packets originated at node  $i$  that are at time  $t$  in  $i$ th node buffer in the infinite buffer network. Likewise, define the collection of quantities  $u_{ij}^\infty(t)$ ,  $1 \leq i \leq a_N^{-1}$ ,  $1 \leq j \leq 4$ . According to the initialization procedure, we have

$$v_i(0) = v_i^\infty(0) \wedge (b^* - 1), \quad (15)$$

$$u_{ij}(0) = u_{ij}^\infty(0) \wedge b, \quad (16)$$

where  $b = \lfloor \beta \log N/4 \rfloor$ .

In order to estimate  $\mathbb{P}_{\mathcal{X}}[\mathcal{E}]$ , we introduce two events:

$$\mathcal{I} := \{v_i(0) < b^* - 1, \forall i\} \cap \{u_{ij}(0) < b, \forall i, \forall j\},$$

$$\begin{aligned} \mathcal{E}^\infty := & \{v_i^\infty(t) < b^* - 1, \forall i, \forall t \in [0, T]\} \\ & \cap \{u_{ij}^\infty(t) < b, \forall i, \forall j, \forall t \in [0, T]\}. \end{aligned}$$

The first event  $\mathcal{I}$ , defined for the finite buffer network, indicates whether the frame  $[0, T]$  is discarded right at  $t = 0$ . The second event  $\mathcal{E}^\infty$  indicates that quantities  $v_i^\infty(t)$  and  $u_{ij}^\infty(t)$  remain bounded in the infinite buffer network during the whole duration of the frame. Next, it is straightforward to relate the conditional probabilities  $\mathcal{E}$ ,  $\mathcal{E}^\infty$  and  $\mathcal{I}$  as follows:

$$\begin{aligned} \mathbb{P}_{\mathcal{X}}[\mathcal{E}] &= \mathbb{P}_{\mathcal{X}}[\mathcal{E}, \mathcal{I}] = \mathbb{P}_{\mathcal{X}}[\mathcal{E}^\infty, \mathcal{I}] \\ &\geq 1 - \mathbb{P}_{\mathcal{X}}[\overline{\mathcal{E}^\infty}] - \mathbb{P}_{\mathcal{X}}[\overline{\mathcal{I}}], \quad (17) \end{aligned}$$

where  $\overline{\mathcal{E}^\infty}$  and  $\overline{\mathcal{I}}$  indicate the complementary events of  $\mathcal{E}^\infty$  and  $\mathcal{I}$ , respectively. The second equality in the preceding equation is due to the fact that in the absence of losses in the finite buffer network the two networks behave identically, i.e., all

the packet transitions between cells are the same. Bounding the conditional probability of event  $\bar{\mathcal{I}}$  using (15) and (16) is easy since the infinite buffer network is in stationarity at time  $t = 0$ :

$$\begin{aligned} \mathbb{P}_{\mathcal{X}}[\bar{\mathcal{I}}] &\leq \sum_{i=1}^N \mathbb{P}_{\mathcal{X}}[v_i^\infty(0) \geq b^*] + \sum_{i=1}^{a_N^{-1}} \sum_{j=1}^4 \mathbb{P}_{\mathcal{X}}[u_{ij}^\infty(0) > b] \\ &\leq N \left( \frac{\lambda s / \mu}{1 - \rho} \right)^{b^*} + 4a_N^{-1} \rho^{b+1}. \end{aligned} \quad (18)$$

In the preceding relationship, it is important to consider only the conditional probabilities on event  $\mathcal{X}$  since on event  $\mathcal{X}$  we have that each queue in the network is not overloaded ( $\rho < 1$ ) and results for stable Kelly networks apply. The bound is due to the fact that the number of customers in each queue is geometrically distributed where the parameter of the distribution is its load. In addition, each packet is of a certain class with probability that is equal to the ratio of arrival rates of this particular class and all classes together.

To analyze the probability of event  $\bar{\mathcal{E}}^\infty$  we introduce two families of stopping times

$$\begin{aligned} \tau_i(b^*) &:= \inf\{t > 0 : v_i^\infty(t) \geq b^*\}, \\ \tau_{ij}(b) &:= \inf\{t > 0 : u_{ij}^\infty(t) > b\}, \end{aligned}$$

i.e.,  $\tau_i(b^*)$  is the first time the number of packets originated in node  $i$  reaches level  $b^*$  in the  $i$ th node buffer, given that the system is in stationarity at time  $t = 0$ . Then the union bound renders

$$\begin{aligned} \mathbb{P}_{\mathcal{X}}[\bar{\mathcal{E}}^\infty] &= \mathbb{P}_{\mathcal{X}} \left[ \min_{1 \leq i \leq N} \tau_i(b^*) \wedge \min_{1 \leq i \leq a_N^{-1}, j=1, \dots, 4} \tau_{ij}(b) < T \right] \\ &\leq \sum_{i=1}^N \mathbb{P}_{\mathcal{X}}[\tau_i(b^*) < T] + \sum_{i=1}^{a_N^{-1}} \sum_{j=1}^4 \mathbb{P}_{\mathcal{X}}[\tau_{ij}(b) < T]. \end{aligned}$$

The routing, buffer management and bandwidth sharing policies ensure that the infinite buffer network is feed-forward. Namely, queues in the network can be labeled in such a way that each packet traverses queues with an increasing order of labels (see Fig. 6). This fact allows one to estimate the hitting times since input processes to individual virtual buffers and the node's designated local buffers are independent of their states. In particular, applying Lemma 4 yields the following estimate

$$\begin{aligned} \mathbb{P}_{\mathcal{X}}[\bar{\mathcal{E}}^\infty] &\leq 2N\lambda T \left( \frac{\lambda s / \mu}{1 - \rho} \right)^{b^* - 1} + N e^{-\lambda T / 4} \\ &\quad + 8a_N^{-1} \rho \mu T s^{-1} \rho^b + 4a_N^{-1} e^{-\lambda T / 4}, \end{aligned} \quad (19)$$

where we used the fact that on event  $\mathcal{X}$  all queues have their load bounded from above by  $\rho$  and the fact that if the arrival rate of packets from outside the cell is non-zero then it is at least  $\lambda$ . Combining (17), (18) and (19) with  $\rho < 1$  and  $\lambda < \mu/s$  yields the desired result

$$\begin{aligned} \mathbb{P}_{\mathcal{X}}[\bar{\mathcal{E}}] &\geq 1 - (1 + 2\mu T / s) \left[ N \left( \frac{\lambda s / \mu}{1 - \rho} \right)^{b^* - 1} + 4a_N^{-1} \rho^b \right] \\ &\quad - (N + 4a_N^{-1}) e^{-\lambda T / 4}. \end{aligned}$$

This concludes the proof of the lemma.

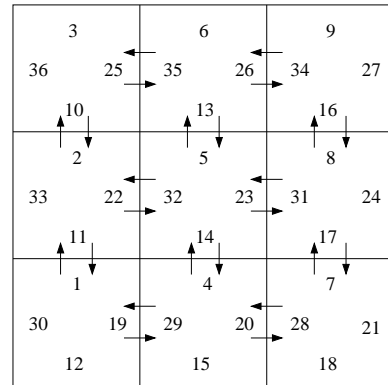


Fig. 6. Queues in the network (see Fig. 4) can be labeled with integers in such a way that each packet traverses queues with an increasing order of labels. The figure shows a labeling for a network that consists of 9 cells, i.e., 36 queues. Column-first routing is used. For example, the route from the lower-left cell to the upper-right cell is 1-2-25-26.

## REFERENCES

- [1] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inform. Theory*, vol. 46, no. 2, pp. 388–404, 2000.
- [2] A. El Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Throughput-delay trade-off in wireless networks," in *Proc. IEEE Infocom*, Hong Kong, March 2004.
- [3] —, "Optimal throughput-delay scaling in wireless networks - Part I: The fluid model," preprint.
- [4] —, "Optimal throughput-delay scaling in wireless networks - Part II: Constant-size packets," preprint.
- [5] M. Franceschetti, O. Dousse, D. Tse, and P. Thiran, "On the throughput capacity of random wireless networks," preprint.
- [6] S. Kulkarni and P. Viswanath, "A deterministic approach to throughput scaling in wireless networks," *IEEE Trans. Inform. Theory*, vol. 50, no. 6, pp. 1041–1049, 2004.
- [7] P. R. Kumar and L.-L. Xie, "A network information theory for wireless communications: Scaling laws and optimal operation," *IEEE Trans. Inform. Theory*, vol. 50, no. 5, pp. 748–767, 2004.
- [8] O. Leveque and E. Telatar, "Information theoretic upper bounds on the capacity of ad hoc networks," preprint.
- [9] A. Jovičić, P. Viswanath, and S. Kulkarni, "Upper bounds to transport capacity of wireless networks," preprint.
- [10] G. Barrenechea, B. Bekerull-Lozano, and M. Vetterli, "Lattice sensor networks: Capacity limits, optimal routing and robustness to failures," in *Proc. IPSN*, Berkeley, CA, April 2004.
- [11] R. M. Loynes, "The stability of a queue with non-independent inter-arrival and service times," *Proc. Cambridge Philos. Soc.*, vol. 58, pp. 497–520, 1962.
- [12] P. Jelenković, P. Momčilović, and M. Squillante, "Scalability of wireless networks," EECS, University of Michigan, Tech. Rep., July 2005, www.eecs.umich.edu/~petar.
- [13] F. Kelly, *Reversibility and Stochastic Networks*. New York: Wiley, 1979.
- [14] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Prentice-Hall, 1992.
- [15] T. Liggett, *Stochastic Interacting Systems: Contact, Voter and Exclusion Processes*. Berlin: Springer-Verlag, 1999.
- [16] M. Karol, S. J. Golestani, and D. Lee, "Prevention of deadlocks and livelocks in lossless backpressured packet networks," *IEEE/ACM Trans. Networking*, vol. 11, no. 6, pp. 923–934, 2003.
- [17] F. T. Leighton, *Introduction to parallel algorithms and architectures: Arrays, trees, and hypercubes*. San Mateo, CA: Morgan Kaufmann Publishers, 1992.
- [18] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad-hoc wireless networks," *IEEE/ACM Trans. Networking*, vol. 10, no. 4, pp. 477–486, 2002.
- [19] M. Neely and E. Modiano, "Capacity and delay tradeoffs for ad-hoc mobile networks," preprint.
- [20] P. Robert, *Stochastic Networks and Queues*. Berlin: Springer, 2003.
- [21] X. Chao, M. Miyazawa, and M. Pinedo, *Queueing Networks: Customers, Signals, and Product Form Solutions*. Chichester, England: Wiley, 1999.