Improving Delay in Ad-Hoc Mobile Networks Via Redundant Packet Transfers

Michael J. Neely
MIT—LIDS: mjneely@mit.edu
http://web.mit.edu/mjneely/www/

Eytan Modiano
MIT—LIDS: modiano@mit.edu
http://web.mit.edu/modiano/www/

***Abstract*— We consider the throughput/delay tradeoffs for scheduling data transmissions in a mobile ad-hoc network. To reduce delays in the network, each user sends redundant packet information along multiple paths to the destination. Such redundancy improves delay at the cost of increasing network congestion. Assuming the network has a cell partitioned structure and users move according to a simplified iid mobility model, we compute the exact network capacity and delay when no redundancy is used. The capacity achieving algorithm is a modified version of the Grossglauser-Tse 2-hop relay algorithm and provides $O(N)$ delay (where N is the number of users). We then show that redundancy cannot increase capacity, but can significantly improve delay. A lower bound on delay of $O(\sqrt{N})$ is computed for any algorithm (with or without redundancy) which restricts packets to 2-hop paths. A scheduling protocol which uses redundancy is presented and shown to achieve this delay bound when data rates of all sessions are reduced to $O(1/\sqrt{N})$.**

I. INTRODUCTION

We consider the effects of transmitting redundant packet information along independent paths of an ad-hoc wireless network with mobility. Such redundancy improves delay at the cost of increasing overall network congestion. We show that redundancy cannot increase network capacity, but can significantly improve delay performance, yielding delay reductions by several orders of magnitude when data rates are sufficiently less than capacity.

We use the following *cell partitioned* network model: The network is partitioned into C non-overlapping cells of equal size (see Fig. 1). There are N mobile users independently roaming from cell to cell over the network, and time is slotted so that users remain in their current cells for a timeslot, and potentially move to a new cell at the end of the slot. If two users are within the same cell during a timeslot, one can transfer a single packet to the other. Each cell can support exactly one packet transfer per timeslot, and users within different cells cannot communicate during the slot. Multi-hop packet transfer proceeds as users change cells and exchange data. The cell partitioning reduces scheduling complexity and facilitates analysis. Similar cell partitioning has recently been considered by Cruz et. al in [1].

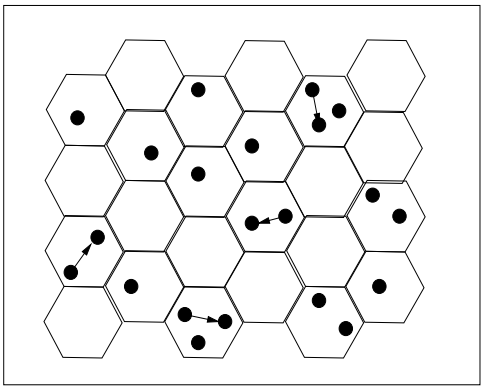

Fig. 1. A cell-partitioned ad-hoc wireless network with C cells and N mobile users.

We consider the following simplified mobility model: Every timeslot, users choose a new cell location independently and identically distributed over all cells in the network. Such a mobility model is of course an oversimplification. Indeed, actual mobility is better described by Markovian dynamics, where users choose new locations every timeslot from the set of cells adjacent to their current cell. However, analysis under the simplified iid mobility model provides a meaningful bound on performance in the limit of *infinite mobility*. With this assumption, the network topology dramatically changes every timeslot, so that network behavior cannot be predicted and fixed routing algorithms cannot be used. Rather, because information about the current and future locations of users is unknown, one must rely on robust scheduling algorithms.

We compute an exact expression for the per-user transmission capacity of the network (for any number of users $N \geq 3$), and show that this capacity cannot be increased by using redundancy. When no redundancy is used, a modified version of the Grossglauser-Tse 2-hop relay algorithm in [2] is presented and shown to achieve capacity. The queueing delay in the network is explicitly computed and shown to be $O(N)/(\mu - \lambda_i)$ (where μ is the per-user network capacity, and λ_i is the rate at which user i transfers packets intended for its destination). Furthermore, it

is shown that no scheduling algorithm can improve upon $O(N)$ delay performance unless redundancy is used.

We then consider modifying the 2-hop relay algorithm to allow redundant packet transmissions. It is shown that no scheme which restricts packets to two hops can achieve a better delay than $O(\sqrt{N})$. A scheduling protocol that employs redundant packet transmissions is developed and shown to achieve this delay bound when all users communicate at a reduced data rate.

Previous work on the capacity of ad-hoc wireless networks is found in [1-6]. Gupta and Kumar present asymptotic results for static networks in [3], [4], where it is shown that per-user network capacity is $O(1/\sqrt{N})$, and hence vanishes as the number of users N increases. The effect of mobility on the capacity of ad-hoc wireless networks was first explicitly developed in [2], where a 2-hop relay algorithm was developed and shown to support constant per-user throughput which does not vanish as the size of the network grows. These works do not consider the associated network *delay*, and analysis of the fundamental queueing delay bounds for general networks remains an important open question.

In this paper, we consider the delay performance offered by cell partitioned wireless networks. The contributions are twofold: First, we demonstrate network delay analysis which considers the full effects of queueing, and show that delay grows as $O(N)$ when no redundancy is used. Second, we show that redundancy can be used to improve delay at the cost of reducing capacity.

In the next section, we establish the capacity of the cell partitioned network and analyze the delay of the capacity achieving relay algorithm. In Section III we develop performance bounds for transmission schemes with redundancy, and in Section IV we provide scheduling protocols which achieve these bounds.

II. Capacity, Delay, and the 2-Hop Relay Algorithm

Consider a cell partitioned network such as that of Fig. 1. The shape and layout of cell regions is arbitrary, although we assume that cells have identical area, do not overlap, and completely cover the network area. We define:

- N = Number of Mobile Users
- C = Number of Cells
- $d = N/C$ = User/Cell density

Users move independently according to the *full-mobility model*, where the steady state location of each user is uniform over all cells. Let λ_{ij} represent the rate user i sends packets destined for user j (in units of packets/slot). These packets are transmitted and routed through the network to reach their destinations according to some scheduling algorithm. A scheduling algorithm is *stable* if the (λ_{ij}) data rates are satisfied for all users so that queues do not grow to infinity and average delays are bounded.

A. Network Capacity

We define the *symmetric capacity region* as the region of all stabilizable data rates such that no user is transmitting at a higher total rate than any other. Let K represent the maximum number of destination users to which a source transmits (i.e., for each user i, at most K of the λ_{ij} terms are nonzero).

Theorem 1. *The symmetric capacity region of the network has the form:*

$$\sum_j \lambda_{ij} \leq \frac{(1-e^{-d}-de^{-d})}{2d} + O(K/N) \ \forall i \quad (1)$$

$$\sum_i \lambda_{ij} \leq \frac{(1-e^{-d}-de^{-d})}{2d} + O(K/N) \ \forall j \quad (2)$$

Proof: The proof of this theorem is similar to the proof of Theorem 2 below, and is omitted for brevity (see [5] for details). □

We now consider a special case of the above result when all users communicate with the same rate λ to a unique destination user (i.e., $K = 1$), similar to the situation treated in [4], [2]. We assume N is even and consider the one-to-one pairing: $1 \leftrightarrow 2, 3 \leftrightarrow 4, \ldots, (N-1) \leftrightarrow N$; so that user 1 communicates with user 2 and user 2 communicates with user 1, user 3 communicates with user 4 and user 4 communicates with user 3, and so on. Other source-destination scenarios can be treated similarly. In the following, we compute an exact expression for the capacity λ for every value of N, and then provide an algorithm for achieving this capacity with bounded average delay. Users move independently with a steady state probability $1/C$ of being in any particular cell.

Theorem 2. *The capacity of the network is:*

$$\mu = \frac{p+q}{2d} \quad (3)$$

where

$$p = 1 - \left(1-\tfrac{1}{C}\right)^N - \tfrac{N}{C}\left(1-\tfrac{1}{C}\right)^{N-1} \quad (4)$$

$$q = 1 - \left(1-\tfrac{1}{C^2}\right)^{N/2} \quad (5)$$

and hence the network can stably support users simultaneously communicating at any rate $\lambda < \mu$.

Proof: The proof of the above theorem involves proving that $\lambda \leq \mu$ is necessary for network stability, and that $\lambda < \mu$ is sufficient. Sufficiency is established in the next subsection, where a bound on average delay is derived. Here we prove necessity.

Consider any stabilizing scheduling strategy, perhaps one which uses full knowledge of future events. Let $X_h(T)$ represent the total number of packets transferred over the network from sources to destinations in h hops during the interval $[0, T]$. Fix $\epsilon > 0$. For network stability, there must be arbitrarily large values of T such that the sum output rate is within ϵ of the total input rate:

$$\frac{\sum_{h=1}^{\infty} X_h(T)}{T} \geq N\lambda - \epsilon \tag{6}$$

The total number of packet transmissions in the network during $[0, T]$ is at least $\sum_{h=1}^{\infty} hX_h(T)$. This value must be less than or equal to the total number of transmission opportunities $Y(T)$, and hence:

$$\sum_{h=1}^{\infty} hX_h(T) \leq Y(T) \tag{7}$$

where $Y(T)$ represents the total number of cells containing at least 2 users in a particular timeslot, summed over all timeslots $1, 2, \ldots, T$. By the law of large numbers, it is clear that $\frac{1}{T}Y(T) \to Cp$ as $T \to \infty$, where p is the steady state probability that there are two or more users within a particular cell, and is given by (4).

From (6) and (7), it follows that

$$\frac{1}{T}X_1(T) + 2\left((N\lambda - \epsilon) - \frac{1}{T}X_1(T)\right) \leq \frac{1}{T}Y(T)$$

and hence

$$\lambda \leq \frac{\frac{1}{T}Y(T) + \frac{1}{T}X_1(T) + 2\epsilon}{2N} \tag{8}$$

It follows that maximizing λ subject to (8) involves placing as much rate as possible on the single hop paths. However, the time average rate $\frac{1}{T}X_1(T)$ of 1-hop communication between source-destination pairs is bounded. Indeed, the probability q that a particular cell contains a source-destination pair during a timeslot can be written as 1 minus the probability that no such pair is present. For the source-destination matching $1 \leftrightarrow 2, 3 \leftrightarrow 4, \ldots$, this probability is given as the value q specified in (5). Let $q(T)$ represent the empirical fraction of time a cell contains a source-destination pair (averaged over all cells), so that $q(T) \to q$. It follows that:

$$\frac{1}{T}X_1(T) \leq Cq(T) \tag{9}$$

Combining constraints (8) and (9) and taking limits as $T \to \infty$, we have:

$$\lambda \leq \frac{Cp + Cq + 2\epsilon}{2N}$$

The necessary condition follows by using the user/cell density definition $d = N/C$, and noting that ϵ can be chosen to be arbitrarily small. □

Taking limits as $N \to \infty$, we find the network capacity tends to the fixed value $(1 - e^{-d} - de^{-d})/(2d)$, verifying (1) and (2). This expression goes to zero when d tends either to zero or infinity. Hence, for nonzero capacity, the ratio $d = N/C$ should be fixed as both N and C scale up. The optimal user/cell density d^* and the corresponding capacity μ^* are: $d^* = 1.7933$, $\mu^* = 0.1492$. Thus, large cell partitioned networks cannot support more than 0.1492 packets/slot, but can achieve arbitrarily close to this data rate by scaling the number of cells C with N to maintain a constant user/cell density d^*.

We note that the capacity arguments in Theorems 1 and 2 are the same if we assume packet scheduling uses redundancy in transmissions, or even if scheduling is performed using perfect knowledge of future events (see Appendix A). We thus have the following corollary.

Corollary 1. *Redundant packet transfers or perfect knowledge of future cell states of all users does not increase network capacity.*

B. Delay Analysis and the 2-Hop Relay Algorithm

In this section, we consider the following modified version of the Grossglauser-Tse relay algorithm of [2], and show the algorithm is capacity achieving with a bounded average delay.

Cell Partitioned Relay Algorithm: Every timeslot and for each cell containing at least two users:

1) If there exists a source-destination pair within the cell, randomly choose such a pair (uniformly over all such pairs in the cell). If the source contains a new packet intended for that destination, transmit. Else remain idle.
2) If there is no source-destination pair in the cell, designate a random user within the cell as sender. Independently choose another user as receiver among the remaining users within the cell. With equal probability, randomly choose one of the two options:
   - *Send a Relay packet to its Destination:* If the designated transmitter has a packet destined for the designated receiver, send that packet to the receiver. Else remain idle.

- *Send a New Relay Packet:* If the designated transmitter has a new packet (one that has never before been transmitted), relay that packet to the designated receiver. Else remain idle.

Because packets that have already been relayed are restricted from being transmitted to any user other than their destination, the above algorithm restricts all routes to 2-hop paths. The algorithm schedules packet transfer opportunities without considering queue backlog. Performance can be improved by allowing alternative scheduling opportunities in the case when no packet is available for the chosen transmission. However, the algorithm as stated admits a nice *decoupling* between sessions, where individual users see the network only as a source, destination, and intermediate relays, and transmissions of packets for other sources are reflected simply as random ON/OFF service opportunities.

We assume data arrives to the source every slot as an independent Bernoulli stream of rate λ. Using the above Cell Partitioned Relay Algorithm and assuming iid mobility each slot, the source queue becomes a Bernoulli/Bernoulli queue, and can be shown to be *reversible* [7], [8]. Because of the correlations at the relay queues of the network, namely, that a relay cannot transmit when receiving from the source, and that no two relays can simultaneously receive or transmit, the relay queues are not Bernoulli/Bernoulli. However, these queues individually receive Bernoulli streams from the output of the source, and have individual Markov chains which can be written as a simple birth-death process [7].

Theorem 3. *Consider a cell partitioned network (with N users and C cells) under the 2-hop relay algorithm, and assume that users change cells iid and uniformly over each cell every timeslot. If the exogenous input stream to user i is a Bernoulli stream of rate λ_i (where $\lambda_i < \mu$), then the total network delay W_i for user i traffic satisfies:*

$$\mathbb{E}\{W_i\} = \frac{N - 1 - \lambda_i}{\mu - \lambda_i} \tag{10}$$

where the capacity μ is defined in (3).

Proof: The proof of this theorem uses reversibility of the first stage queue, and is omitted for brevity. □

Note that the decoupling property of the cell partitioned relay algorithm admits a decoupled delay bound, so that the waiting time for user i packets depends only on the rate of the input stream for user i, and does not depend on the rate of other streams—even if the rate of these streams is greater than capacity. It follows that the network is stable with bounded delays whenever all input streams are less than capacity, i.e., when $\lambda_i < \mu$ for all users i. Thus, the relay algorithm achieves the capacity bound given in (3) of Theorem 1. The form of the delay expression is worth noting. First note the classic $1/(\mu - \lambda_i)$ behavior, representing the asymptotic growth in delay as data rates are pushed towards the capacity boundary. Such behavior is typical of single queue systems (consider for example the classic P-K formula for an M/G/1 queue [7]), and hence the behavior of the *entire network* is similar to the behavior of a single queue. Second, note that for a fixed loading value $\rho_i = \lambda_i/\mu$, delay is $O(N)$, growing linearly in the size of the network.

It is remarkable that exact delay analysis of a capacity achieving control strategy can be obtained for this multi-user wireless network. The analysis is enabled by the Bernoulli input assumption. If inputs are assumed to be Poisson, we cannot derive exact delay expressions. However, the delay theory in [6], [5] can be used in this case to develop a delay bound, and the bound for Poisson inputs is not considerably different from the exact expression for Bernoulli inputs given in (10). These results can also be extended to the case when the mobility model conforms to a Markovian random walk, rather than an iid mobility model [6], [5].

III. Fundamental Delay Bounds

In the previous subsection we showed that the cell partitioned relay algorithm yields an average delay of $O(N/(\mu - \lambda_i))$. Inspection of (10) shows that this $O(N)$ characteristic cannot be removed by decreasing the data rate λ. The following questions emerge: Can another scheduling algorithm be constructed which improves delay? What is the minimum delay the network can guarantee, and for what data rates is this delay obtainable? More generally, for a given data rate λ (assumed to be less than the system capacity μ), we ask: What is the optimal delay bound, and what algorithm achieves this? In this section we present several fundamental bounds on delay performance, which establishes initial steps towards addressing these general questions.

A. Scheduling Without Redundancy

Suppose that no redundancy is used: that is, packets are not duplicated and are held by at most one user of the network at any given time.

Theorem 4. *Algorithms which do not use redundancy cannot achieve an average delay of less than $O(N)$.*

Proof: The minimum delay of any packet is computed by considering the situation where the network is empty and user 1 sends a single packet to user 2. It is easy to verify that relaying the packet cannot help,

and hence the delay distribution is geometric with mean $C = N/d$. □

Hence, the relay algorithm not only achieves capacity, but achieves the optimal $O(N)$ delay performance among all strategies which do not use redundancy. Other policies which do not use redundancy can perhaps improve upon the delay coefficient, but cannot change the $O(N)$ characteristic.

B. Scheduling With Redundancy

Here we consider schemes using redundancy. Although redundancy cannot increase capacity, it can considerably improve delay. Clearly, the time required for a packet to reach the destination can be reduced by repeatedly transmitting this packet to many users of the network—improving the chances that some user holding an original or duplicate version of the packet reaches the destination. Consider any network algorithm (which may or may not use redundancy) that restricts packet transfers to 2-hop paths.

Theorem 5. *No algorithm (with or without redundancy) which restricts packets to 2-hop paths can provide an average delay better than $O(\sqrt{N})$.*

The theorem is proved by again considering the situation of sending a single packet from source to destination. Clearly the optimal scheme is to have the source send duplicate versions of the packet to new relays whenever possible, and for the packet to be relayed to the destination as soon as either the source or a duplicate-carrying relay enters the same cell as the destination.

We let $\mathbb{E}\{T_N\}$ represent the expected time to reach the destination under this scheme for transmitting a single packet. The following lemma reveals that $\mathbb{E}\{T_N\}$ is $O(\sqrt{N})$, which proves Theorem 5.

Lemma 1. *For large N, we have:*

$$e^{-d}\sqrt{N} \leq \mathbb{E}\{T_N\} \leq \sqrt{N}\left(\frac{1}{1-e^{-d}} + \frac{1}{d}\right)$$

Proof: (a) *Lower Bound:* To prove the lower bound, note that during timeslots $\{1, 2, \ldots, \sqrt{N}\}$, there are fewer than $\sqrt{N}$ users holding the packet. Hence, $Pr[T_N > \sqrt{N}] \geq (1-1/C)^{\sqrt{N}\sqrt{N}}$ (where $(1-1/C)^{\sqrt{N}}$ is the probability that nobody within a group of $\sqrt{N}$ particular users enters the cell of the destination during a given timeslot). Recall that the user/cell density d is defined $d \triangleq N/C$. Thus:

$$\begin{aligned}\mathbb{E}\{T_N\} &\geq \mathbb{E}\left\{T_N | T_N > \sqrt{N}\right\} Pr[T_N > \sqrt{N}] \\ &\geq \sqrt{N}\left(1 - \frac{d}{N}\right)^N \rightarrow e^{-d}\sqrt{N}\end{aligned}$$

(b) *Upper Bound:* To prove the upper bound, note that $\mathbb{E}\{T_N\} \leq S_1 + S_2$, where S_1 represents the expected number of slots required to send out duplicates of the packet to $\sqrt{N}$ different users, and S_2 represents the expected time until one user within a group of $\sqrt{N}$ users containing the packet reaches the cell of the destination. The probability of the source meeting a new user is at least $1-(1-1/C)^{N-\sqrt{N}}$ for every timeslot where fewer than $\sqrt{N}$ users have packets, and hence the average time to reach a new user is less than or equal to the inverse of this quantity (i.e, the average time of a geometric variable). Hence:

$$S_1 \leq \frac{\sqrt{N}}{1-(1-1/C)^{N-\sqrt{N}}} \rightarrow \frac{\sqrt{N}}{1-e^{-d}}$$

To compute S_2, note that $P(success)$, the probability that one of the $\sqrt{N}$ users reaches the destination during a slot, satisfies:

$$\begin{aligned}P(success) &= 1-(1-1/C)^{\sqrt{N}} \\ &= 1-(1-d/N)^{N\frac{1}{\sqrt{N}}} \\ &\geq 1-e^{-d/\sqrt{N}}\end{aligned}$$

where the last inequality follows because $(1-d/N)^N$ increases to e^{-d} as $N \rightarrow \infty$. A success occurs after a sequence of Bernoulli trials, and hence:

$$\begin{aligned}S_2 &\leq \frac{1}{1-e^{-d/\sqrt{N}}} \\ &\leq \frac{1}{1-\left(1-d/\sqrt{N}+d^2/(2N)\right)} \\ &= \frac{\sqrt{N}}{d-d^2/\left(2\sqrt{N}\right)} \rightarrow \sqrt{N}/d\end{aligned}$$

where the second to last inequality[1] follows because $e^{-x} \leq 1-x+x^2/2$ whenever $x \geq 0$. Summing S_1 and S_2 proves the result. □

It is also possible to compute a bound for general duplication algorithms which do not restrict to 2-hop paths. Indeed, considering the single packet, single destination scenario, it is clear that an optimal algorithm schedules all users to send duplicate versions of the packet whenever possible. In this way, the number of users containing the packet grows (roughly) geometrically with time, and minimum delay is $O(\log(N))$. However, this method causes complications in a multi-user situation, because a single packet quickly uses up all resources of the network. We consider multi-user scheduling in the next section.

[1] Note that $\left(1-d/\sqrt{N}+d^2/(2N)\right) < 1$ when N is suitably large.

IV. Scheduling for Delay Improvement

In the previous section an $O(\sqrt{N})$ delay bound was developed for redundant scheduling by considering a single packet for a single destination. Two complications arise when designing a general scheduling protocol using redundancy: (1) All sessions must use the network simultaneously, and (2) Remnant versions of a packet that has already been delivered to its destination create excess congestion and must somehow be removed.

Here we show that the properties of the 2-hop relay algorithm make it naturally suited to treat the multi-user problem. The second complication of excess packets is more troublesome. Clearly a network with global feedback could deal with the complication by sending immediate feedback to all users over a low bandwidth control channel whenever a packet has been received by its destination. However, such feedback is impractical for large networks. Below we present a scheduling protocol which uses only *partial, in-cell feedback*, requiring only that a receiving node tell its transmitter which packet it is looking for before transmission begins. We assume all packets are labeled with *send numbers* SN, and the in-cell feedback is in the form of a *request number* RN delivered by the destination to the transmitter just before transmission.

Partial Feedback Scheme with Redundancy: The 2-hop relay algorithm is used to establish transmission opportunities for all users. Additional protocol is as follows:

1) Users send each packet $\sqrt{N}$ times, sending out duplicate versions every time they are scheduled to transmit to a new node until either $\sqrt{N}$ have been transmitted, or the user transmits to the intended destination.
2) When a user is scheduled to transmit a relay packet to its destination, the following handshake is performed:
   - The destination delivers its current RN number for the packet it desires.
   - The transmitter deletes all packets in its buffer intended for this destination which have SN numbers lower than RN.
   - The transmitter sends packet RN to the receiver. If the transmitter does not have the requested packet RN, it remains idle for that slot.

Thus, no packet is ever transmitted twice to its destination. Notice that the destination thus receives all packets in order.

Theorem 6. *The Partial Feedback Scheme achieves the $O(\sqrt{N})$ delay bound, with user data rates of $O(1/\sqrt{N})$.*

Proof: A full proof is omitted for brevity (see [5]). The intuition is that the time required for a new packet to reach its destination is at most $X = S_1 + S_2$, where S_1 and S_2 respectively represent the time required to send out $\sqrt{N}$ transmissions and the time required to reach the destination given that $\sqrt{N}$ users have the packet. The expectations of S_1 and S_2 can be computed similarly to the proof of Lemma 1. Blocking due to other sessions increases delay but does not change the $O(\sqrt{N})$ delay characteristic because the average number of blockers in any cell does not scale with N. □

Using similar techniques, we can develop a scheduling protocol which uses redundancy with an *unconstrained* number of hops, to achieve $O(\log(N))$ delay with data rates of $O(\frac{1}{N\log(N)})$. We thus have the following achievable capacity/delay performance tradeoffs:

scheme	capacity	delay
no redundancy	$O(1)$	$O(N)$
redundancy 2-hop	$O(1/\sqrt{N})$	$O(\sqrt{N})$
redundancy multi-hop	$O(\frac{1}{N\log(N)})$	$O(\log(N))$

A simple observation reveals that $delay/rate \geq O(N)$. We conjecture that this inequality is a necessary condition. However, it is likely that improvements on the scheduling algorithms developed here can lead to constant factor improvements in the delay coefficients. Further note that the 'redundancy 2-hop' entry in the table demonstrates that a cell partitioned mobile network can emulate the delay/capacity performance of a Gupta-Kumar static network [4], [3]. It is interesting to explore whether this result generalizes to other mobility models.

References

[1] R.L. Cruz and A. V. Santhanam. "hierarchical link scheduling and power control in multihop wireless networks". *Proceedings of the 40th Annual Allerton Conference on Communication, Control, and Computing*, Oct. 2002.

[2] M. Grossglauser and D. Tse. "mobility increases the capacity of ad-hoc wireless networks". *Proceedings of IEEE INFOCOM*, 2001.

[3] P. Gupta and P.R. Kumar. "critical power for asymptotic connectivity in wireless networks". *IEEE Conference on Decision and Control*, 1998.

[4] P. Gupta and P.R. Kumar. "the capacity of wireless networks". *IEEE Transactions on Information Theory*, Vol. 46:388–404, March 2000.

[5] M.J. Neely. *Dynamic Power Allocation and Routing in Satellite and Wireless Networks with Time Varying Channels*. PhD thesis, Massachusetts Institute of Technology, Laboratory for Information and Decision Systems (LIDS), 2003.

[6] M.J. Neely, E. Modiano, and C.E. Rohrs. "dynamic power allocation and routing for time varying wireless networks". *Proceedings of IEEE INFOCOM*, April 2003.

[7] R. Gallager. *Discrete Stochastic Processes*. Kluwer Academic Publishers, Boston, 1996.

[8] H. Daduna. *Queueing Networks with Discrete Time Scale*. Springer, 2001.