Capacity of Ad Hoc Wireless Networks

Jinyang Li Charles Blake Douglas S. J. De Couto Hu Imm Lee Robert Morris
M.I.T. Laboratory for Computer Science
{jinyang, cblake, decouto, hilee, rtm}@lcs.mit.edu

Abstract

Early simulation experience with wireless ad hoc networks suggests that their capacity can be surprisingly low, due to the requirement that nodes forward each others' packets. The achievable capacity depends on network size, traffic patterns, and detailed local radio interactions. This paper examines these factors alone and in combination, using simulation and analysis from first principles. Our results include both specific constants and general scaling relationships helpful in understanding the limitations of wireless ad hoc networks.

We examine interactions of the 802.11 MAC and ad hoc forwarding and the effect on capacity for several simple configurations and traffic patterns. While 802.11 discovers reasonably good schedules, we nonetheless observe capacities markedly less than optimal for very simple chain and lattice networks with very regular traffic patterns. We validate some simulation results with experiments.

We also show that the traffic pattern determines whether an ad hoc network's per node capacity will scale to large networks. In particular, we show that for total capacity to scale up with network size the average distance between source and destination nodes must remain small as the network grows. Non-local traffic patterns in which this average distance grows with the network size result in a rapid decrease of per node capacity. Thus the question "Are large ad hoc networks feasible?" reduces to a question about the likely locality of communication in such networks.

1. Introduction

Ad hoc wireless networks promise convenient infrastructure-free communication. We expect the total capacity of such networks to grow with the area they cover, due to spatial re-use of the spectrum: nodes sufficiently far apart can transmit concurrently. However, ad hoc routing requires that nodes cooperate to forward each others' packets through the network. This means that the throughput available to each single node's applications is limited not only by the raw channel capacity, but also by the forwarding load imposed by distant nodes. This effect could seriously limit the usefulness of ad hoc routing.

In this paper, we focus our analysis and simulations on *static* ad hoc networks. Note that in most mobility scenarios, nodes do not move significant distances during packet transit times. Thus, for capacity analysis, we can view mobile networks as effectively static.

The following simplification of an analysis by Gupta and Kumar [8] estimates the per node capacity to be expected in an ad hoc network. Radios that are sufficiently distant can transmit concurrently; the total amount of data that can be simultaneously transmitted for one hop increases linearly with the total area of the ad hoc network. If node density is constant, this means that the total one-hop capacity is $O(n)$, where n is the total number of nodes. However, as the network grows larger, the number of hops between each source and destination may also grow larger, depending on communication patterns. One might expect the average path length to grow with the spatial diameter of the network, or equivalently the square root of the area, or $O(\sqrt{n})$. With this assumption, the total end-to-end capacity is roughly $O(n/\sqrt{n})$, and the end-to-end throughput available to each node is

$$O\left(\frac{1}{\sqrt{n}}\right) \tag{1}$$

Gupta and Kumar also demonstrated the existence of a global scheduling scheme achieving $\Omega(1/\sqrt{n \log n})$ for a uniform random network with random traffic pattern.

It is not encouraging that the throughput available to each node approaches zero as the number of nodes increases. Furthermore, this simple analysis omits the constant factors which determine whether any particular networks will have a useful per node throughput.

A common observation in analyses of ad hoc routing protocols [2, 10, 4] is that capacity is the limiting factor; that is, the symptom of failure under stress is congestion losses. A high volume of routing queries or updates, caused by mobility or a large number of nodes, causes congestion; the result is not just dropped data packets, but also lost routing information and consequent mis-routing of data. Evaluations of ad hoc protocols tend to use very low data rates in order to avoid running out of capacity. For example, Das et al. [4] observe that in a simulated network of 100 nodes, each with a 2 Mbps radio, the throughput available to each node is on the order of a few kilobits per second. They report that their network has an area large enough that 7 transmissions may proceed concurrently without interfering; this means that the per node throughput

This research was supported by grants from NTT Corporation under the NTT-MIT collaboration.

Permission to make digital or hard copies of part or all of this work or personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee.
ACM SIGMOBILE 7/01 Rome, Italy
© 2001 ACM ISBN 1-58113-422-3/01/07…$5.OO

actually available was about 50 times smaller than the apparent capacity. The loads used in other ad hoc routing studies are consonant with this; for example, both Karp and Kung [9] and Broch et al. [2] limit the total offered load to about 60 Kbps despite using 2 Mbps radios. The interaction of ad hoc routing and capacity suggests that any evaluation of an ad hoc network requires an understanding of network capacity.

While the above discussion suggests that ad hoc networks are fundamentally non-scalable, it may not reflect reality. The studies cited above assume a random communication pattern: each pair of nodes is equally likely to communicate, so that packet path lengths grow along with the physical diameter of the network. This assumption is probably reasonable for small networks. However, users in large networks may communicate mostly with physically nearby nodes: their neighbors in the same lecture hall of a university, or on the same floor of a building, or in the same company in a city. If local communication predominates, path lengths could remain nearly constant as the network grows, leading to constant per node available throughput.

This paper makes two contributions to the understanding of practical ad hoc network scalability. At a detailed level, it examines the interaction between ad hoc forwarding and the 802.11 medium access protocol in order to estimate the constants in Equation 1. At a system level, it examines the impact of communication patterns on the form of Equation 1, and determines some conditions under which per node capacity is likely to scale to large networks. These results are likely to be useful both in understanding simulation studies of ad hoc network performance and in the deployment of real ad hoc networks.

2. 802.11 Background

This paper assumes use of the IEEE 802.11 [3] *Distributed Coordination Function*, the access method used in ad hoc mode. To reduce collisions caused by hidden terminals [1] in the network, 802.11 uses a four-way RTS/CTS/Data/Ack exchange. In brief, a node that wishes to send a data packet first sends an RTS (request to send) packet to the destination. If the destination believes the network is idle, it responds with a CTS (clear to send). The sender then transmits the data packet, and waits for an ACK (acknowledgment) from the receiver. If a node overhears an RTS or CTS, it knows the medium will be busy for some time, and avoids initiating new transmissions or sending any CTS packets.

802.11 RTS and CTS packets include the amount of time the medium will be busy for the remainder of the exchange. Each node uses these times to update its “network allocation vector” (NAV). The NAV value indicates the amount of time remaining before the network will become available. Upon successful receipt of an RTS frame *not* addressed to itself, a node updates its NAV to the maximum of the time carried in the RTS frame and its current NAV value. Upon receiving an RTS addressed to itself, a node returns a CTS frame only if its NAV value is zero, otherwise no CTS is sent. Hence, a sender will see no CTS if its RTS packet has collided with another transmission at the receiver, or if the receiver’s NAV indicates that the network is not available. A node times out and re-sends the RTS if it receives no CTS.

802.11 doubles its backoff window each time a timeout occurs; it resets the backoff to a minimum value after a packet is transmitted successfully or is dropped after reaching maximum retry limit.

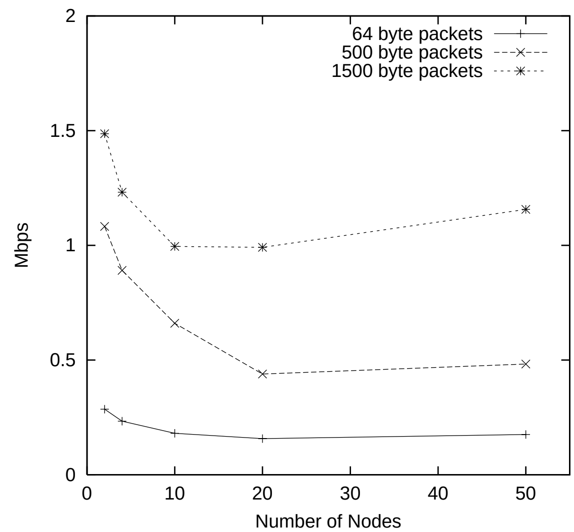

Figure 1: Total network throughput achieved as a function of the number of competing nodes. All nodes are within each others’ radio ranges, and all nodes send as fast as 802.11 allows.

3. MAC Interactions

This section presents simulations of scenarios that illustrate the detailed interaction between ad hoc forwarding and the 802.11 MAC. The section starts with simple scenarios and works towards complex situations that are more likely to be seen.

The simulator used is the *ns* [5] simulator with the CMU wireless extensions [7] whose parameters are tuned to model the Lucent Wavelan card at a 2 Mbps data rate.

Note that one node can interfere with packet reception at another node even when they are too far apart for successful transmission. At long enough distances the interference becomes negligible. In the simulator, the effective transmission range is 250 meters, and the interfering range is about 550 meters.

Most of the simulations involve stations separated by 200 meters, just under the transmission range. This separation is likely to yield close to the maximum capacity possible, since with higher node density the capacity must be divided up among more nodes.

All simulated data packets are preceded by an RTS/CTS exchange, regardless of size. Each data point is an average of 5 runs lasting 300 seconds of simulated time. Nodes are stationary.

3.1 Single Cell Capacity

As a baseline for comparison with more complex situations, Figure 1 shows the simulated total capacity of a single cell (200m by 200m) network as the number of nodes increases. Each node is a packet source, sending as fast as 802.11 allows, each packet to a randomly selected destination. The 2-node scenario has the highest capacity, since it has the minimum contention.

Figure 1 also shows that the RTS/CTS/ACK exchange adds significant overhead. An RTS packet is 40 bytes, CTS and ACK packets are 39 bytes, and the MAC header of a data packet is 47 bytes long.

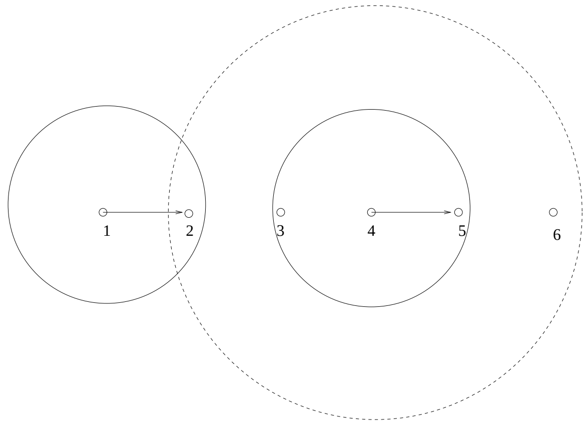

Figure 2: MAC interference among a chain of nodes. The solid-line circle denotes a node's valid transmission range. The dotted-line circle denotes a node's interference range. Node 4's transmission will corrupt node 1's transmissions at node 2.

Thus the data throughput is at most $\frac{1500}{1500+40+39+47} \times 2 \approx 1.8$ Mbps with 1500-byte data packets. When various inter-frame timings are also accounted for this limit is reduced to 1.7 Mbps.

3.2 Capacity of a Chain of Nodes

In an ad hoc network, packets travel along a chain of intermediate nodes toward the destinations. The successive packets of a single greedy connection interfere with each other as they move down the chain, forcing contention in the MAC protocol. This subsection examines the realizable capacity of a single chain of nodes where packets originate at the first node and are forwarded to the last node in the chain.

The following analysis shows that an ideal MAC protocol could achieve a chain utilization as high as $\frac{1}{3}$. Consider the network shown in Figure 2, where node 1 is the source and 6 is the sink. Assume for the moment that the radios of nodes that are not neighbors do not interfere with each other. Nodes 1 and 2 cannot transmit at the same time because node 2 cannot receive and transmit simultaneously. Nodes 1 and 3 cannot transmit at the same time because node 2 cannot correctly hear 1 if 3 is sending. Nodes 1 and 4 can, with the above assumption, send at the same time. This leads to a channel utilization of $\frac{1}{3}$.

However, if one assumes that radios can interfere with each other beyond the range at which they can communicate successfully, the situation is worse. For example, 802.11 nodes in the *ns* simulator can correctly receive packets from 250 meters away, but can interfere at 550 meters. Hence, in Figure 2, node 4's packet transmissions will interfere with RTS packets sent from 1 to 2, preventing 2 from correctly receiving node 1's RTS transmissions or sending the corresponding CTS. Therefore, we expect the maximum utilization of a chain of ad hoc nodes in the *ns* simulator to be $\frac{1}{4}$.

Figure 3 shows simulation results for a single chain. For this set of simulations, each node is 200 meters away from its immediate neighbors. Node 1 is the source of data traffic and the last node in the chain is the traffic sink. Node 1 sends data as fast as its MAC allows. A chain of only two nodes achieves a throughput of about 1.7 Mbps for 1500-byte packets, rather than 2 Mbps, due to the overhead of headers, RTS, CTS, and ACK packets.

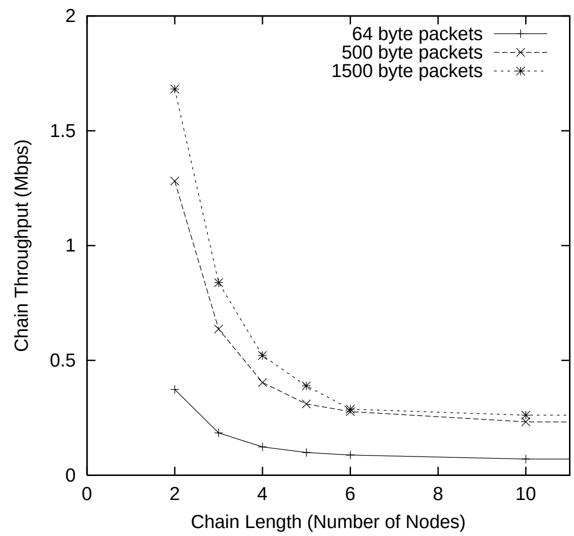

Figure 3: Throughput achieved along a chain of nodes, as a function of the chain length. The nodes are 200 meters apart. The first node originates packets as fast as 802.11 allows, to be forwarded along the chain to the last node. The throughputs for chains of 20 and 50 nodes are the same as for 10 nodes.

As the chains get longer, they approach a utilization of 0.25 Mbps for 1500-byte packets, or $\frac{1}{7}$ of the maximum of 1.7 Mbps. This is substantially less than the predicted $\frac{1}{4}$.

To shed light on the discrepancy between $\frac{1}{4}$ and $\frac{1}{7}$, we conducted a set of simulations in which the source (node 1) sent 1500-byte packets at various controlled rates. Figure 4 shows the results. The maximum throughput is achieved at 0.41 Mbps, which is very close to $1.7 \times \frac{1}{4} = 0.425$ Mbps. However, as the offered load increases (even a little) beyond this optimum, the chain throughput drops sharply. This shows that the 802.11 MAC is capable of sending at the optimal rate, but does not discover the optimum schedule of transmissions on its own.

802.11 fails to achieve the optimum chain schedule because an 802.11 node's ability to send is affected by the amount of competition it experiences. For example, node 3 in a 7-node chain experiences interference from 5 other nodes, while node 1 is interfered with by three other nodes. This means that node 1 could actually inject more packets into the chain than the subsequent nodes can forward, as detailed in Figure 5. These packets are eventually dropped at nodes 2 and 3. The time node 1 spends sending those extra packets decreases delivered throughput since it prevents transmissions from subsequent nodes. This unfairness was also noted by Nandagopal et al. [12]; their proposed solution, which tries to give each single-hop flow equal capacity allocation, might raise the efficiency of ad hoc chain forwarding configurations.

In addition to allocating bandwidth unevenly, 802.11 backoff works badly with ad hoc forwarding. Consider the case when node 4 is in the middle of transmitting a data packet to node 5 and node 1 attempts to initiate transmission to 2 (see Figure 2). Because of two-hop interference, node 1's RTS packet will be corrupted by node 4's transmission and node 2 will not respond with a CTS. Since node 1 does not know about node 4's transmission, it will back off and retry. Hence for the duration of node 4's transmission

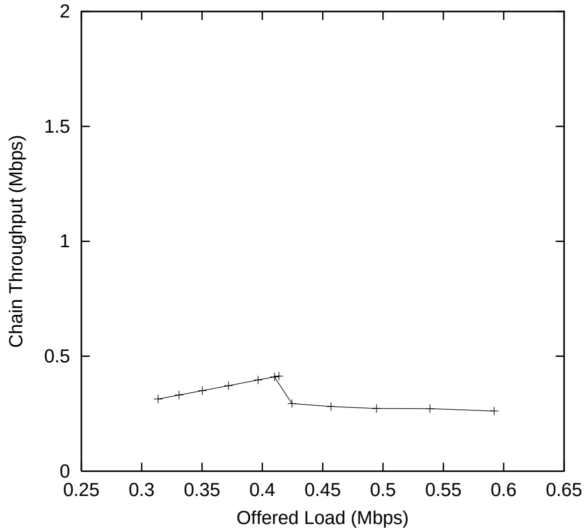

Figure 4: Throughput delivered by an 8-node chain with different send rates, using 1500-byte packets. The fact the peak rate of 0.41 Mbps is not maintained shows the 802.11 MAC does not schedule greedy senders optimally for ad hoc forwarding.

	Node					
	1	**2**	**3**	**4**	**5**	**6**
Send rate	0.48	0.35	0.27	0.26	0.26	0.26
Wasted time (%)	5.4	3.3	3.1	1.5	0	0

Figure 5: Individual node send rates in Mbps, and percent of total time spent in wasted backoff for a 7-node chain, with 1500-byte packets. Note that the 802.11 MAC allows node 1 to send much faster than nodes 2 or 3 can forward, resulting in lost packets.

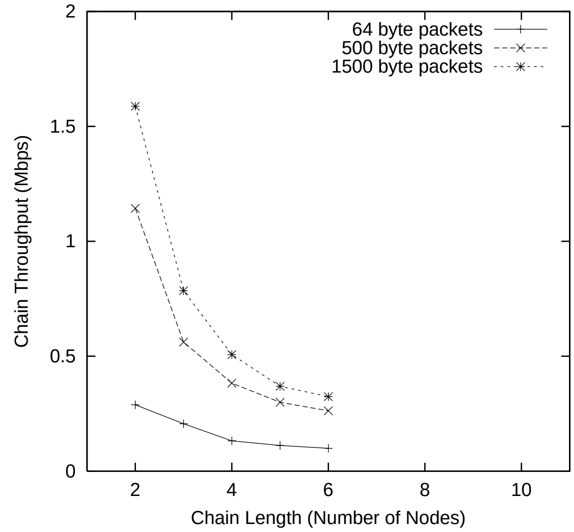

Figure 6: Real hardware throughput achieved along a chain of nodes, as a function of the chain length. Each node was placed at the maximum distance from the previous that allowed low-loss communications. Hardware parameters were set to mimic the simulation parameters as much as possible.

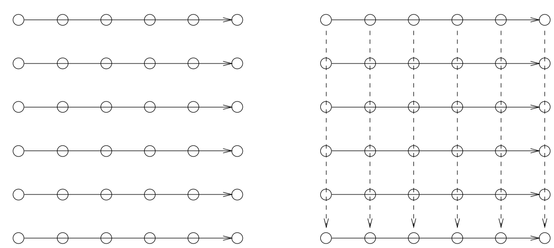

Figure 7: Lattice network topologies, showing just horizontal traffic on the left, and both horizontal and vertical on the right.

all transmission attempts from node 1 will fail, causing a dramatic increase in its backoff window under 802.11's binary exponential backoff scheme. Therefore when node 4 is done with its transmission and has nothing more to send, node 1 may remain backed off during a time in which it could be transmitting. Figure 5 shows the percent of time spent in wasted backoff for each node along a 7-node chain. We consider a certain period of backoff to be wasted when no node that might cause interference is transmitting. As we can see, even though node 3 is receiving packets from node 2 at a rate (0.35 Mbps) already much less than the optimum rate that can be supported (0.425 Mbps), node 3 is unable to maintain the same rate as node 2, while at the same time wasting time backing off.

To summarize, an ideal ad hoc forwarding chain should be able to achieve $\frac{1}{4}$ of the throughput that a single-hop transmission can achieve. Simulation shows that the 802.11 MAC protocol manages $\frac{1}{7}$ of the single-hop throughput.

3.3 Verification of Chain Results

As a rough check on the simulations presented above for ad hoc chains, Figure 6 shows results measured on real hardware. The hardware was configured to mimic the simulation parameters used in Figure 3 as closely as possible. The radios involved are Cisco 340 (Aironet PC4800) cards operated in ad hoc mode at 2 Mbps. Each node was placed as far from its predecessor as possible without sacrificing low-loss communication. Only 6 nodes were available. The fact that Figure 6 matches Figure 3 fairly closely suggests that the simulations do not contain major errors; for example, the average difference for the 1500-byte packet throughput is only 6%.

3.4 Capacity of a Regular Lattice Network

The previous analysis showed how the successive nodes in a single forwarding chain interfere with each other. To gauge the effectiveness of 802.11 channel allocation, we consider a lattice network. Two types of traffic pattern will be discussed: horizontal traffic flows moving from the left edge to the right edge and crossed horizontal and vertical flows (see Figure 7). The regularity of the network and traffic patterns allows estimation of nearly optimal global scheduling schemes to compare with 802.11's actual performance.

Consider the scenario in the left-hand half of Figure 7. Here a lattice of nodes has parallel traffic flows moving from the left edge to the right edge. Assume each node is 200 meters from its east,

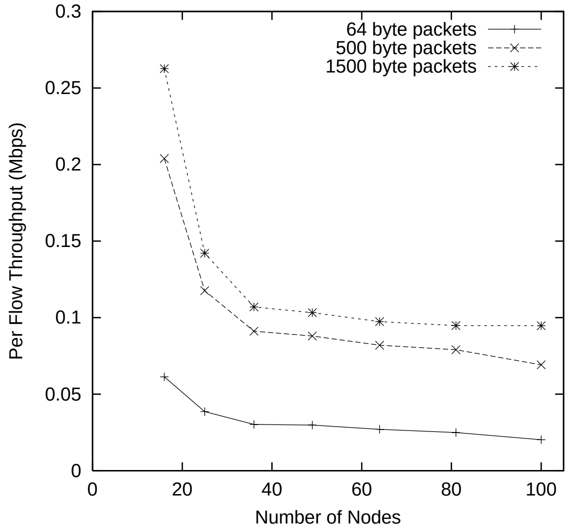

Figure 8: Average per flow throughput in square lattice networks with horizontal data streams only, as a function of network size. There are as many parallel chains as there are nodes per chain. The *X* axis value is the total number of nodes. Each node is separated from its four neighbors by 200 meters.

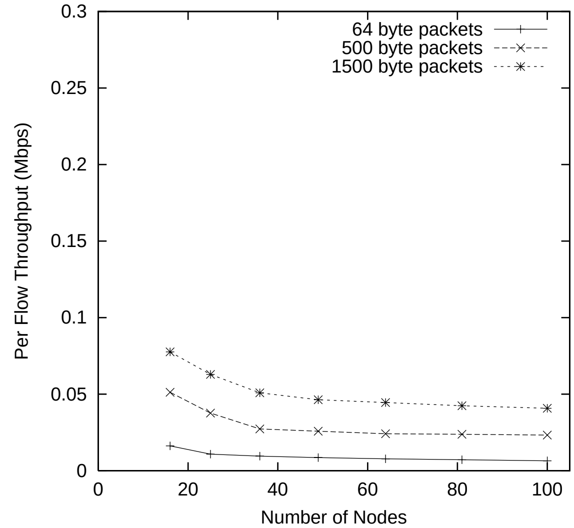

Figure 9: Average per flow throughput in square lattice networks with both horizontal and vertical data streams. This configuration has twice as many chains of traffic sharing the same network as Figure 8, which explains most of the difference between the two results.

west, north, and south radio neighbors. To account for inter-flow interference, when only every third chain is active, the active chains are separated vertically by more than the 550 meter interference limit. This implies that every third chain can operate without inter-chain interference, potentially delivering the $\frac{1}{4}$ of channel capacity derived in Section 3.2. Thus each flow in the lattice network may be expected to achieve a throughput of $\frac{1}{12}$ of the channel capacity. For 1500-byte packets, this is $\frac{1}{12} \times 1.7$ Mbps, or 0.14 Mbps.

Figure 8 shows the per flow throughput for a variety of lattice sizes. The number of chains is the same as the number of nodes in each chain, producing square lattices. The total number of nodes is shown on the *X* axis. As the network grows large, the per flow throughput for 1500-byte packets settles at about 0.1 Mbps, somewhat less than our estimated value. The inefficiencies of 802.11 we have found in the chain scenarios are still present: nodes in the beginning of the chain experience less contention and hence send more packets that could handled by nodes in the later part of the chain. There are also wasted backoff periods for the same reason as explained in the chain scenario.

3.5 Cross Traffic in a Lattice

Now consider a slightly more general situation, in which both vertical and horizontal flows are present, as in the right-hand diagram in Figure 7. All traffic originates at the top and left edges of the network, and is forwarded downward or rightward to the opposite edges; the middle nodes do not originate any traffic.

In this case, we should not expect the overall capacity of the network to decrease significantly. In theory we could impose a schedule on the entire network in which all the vertical flows operate in one time cycle, and all the horizontal flows in the next. This would cause each flow to see half as much throughput as in the previous section, but since there are twice as many flows, the overall network throughput is the same. Of course, 802.11 may not schedule packets this efficiently in practice. For example, the fact that each node has a single queue means that a node may lose a chance to send a packet vertically while the packet at the head of the queue is waiting for contention in the horizontal direction. Figure 9 shows the average per flow throughput obtained by simulation, which is slightly less than the predicted value of half of the per flow throughput for lattice networks without cross traffic. We find that the average percentage of time spent in wasted backoff is 2.23% as opposed to 0.75% in the 8 by 8 lattice network without cross traffic. We consider a backoff period to be wasteful if any packet in the queue (not necessarily at the head) might be transmitted successfully during that time. The increased wasted backoff reflects head-of-queue blocking.

As an alternate analysis, the efficiency of the 802.11 MAC under different topologies and traffic patterns can be evaluated by measuring total one-hop network throughput. Figure 10 illustrates the simulated total throughput obtained in various 2-dimensional network configurations. The *X* axis indicates the physical area of the network; the number of nodes is proportional to the area.

The Y axis indicates the one-hop throughput of the network with 1500-byte packets. One-hop throughput measurements count all radio transmissions for data packets that successfully arrive at their final destinations, including packets forwarded by intermediate nodes. One-hop throughput is similiar in concept to the bit-meter/second unit proposed in [8]. Figure 10 shows that one-hop throughput scales roughly linearly with the area of network. The actual slope of the curve depends on how effectively 802.11 schedules packet transmissions. The points marked “horizontal” reflect the network and traffic configuration described in the previous sub-section. The points marked “horizontal and vertical” show that the addition of vertical traffic decreases the total one-hop capacity. However, the fact that it is just a slight constant factor decrease implies that 802.11 does find a reasonably efficient schedule for interleaving the two directions.

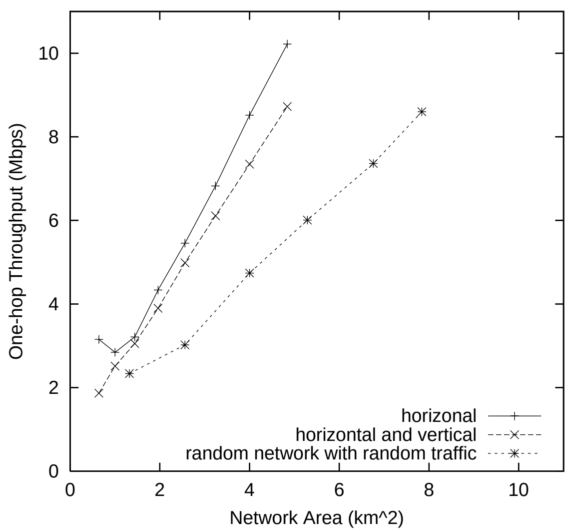

Figure 10: Total one-hop throughput for lattice networks with just horizontal traffic, lattices with both horizontal and vertical traffic, and networks with random node placement and random source-destination pairs. The *X* axis indicates the network area; the number of nodes is proportional to the area. The *Y* axis indicates total one-hop throughput measured as the sum total of bits of data sent by all nodes per second, including forwarded bits. The simulations use 1500-byte packets. Note that the total one-hop capacity scales similarly in all three situations.

3.6 Random Traffic in a Random Layout

As a final step toward evaluating realistic scenarios, let us relax both the regularity of node placement and the regularity of traffic patterns. Instead, assume that nodes are placed uniformly at random on a square universe, and that every node sends packets, each packet to a different randomly chosen recipient. The send rates are adjusted to keep the total drop rate below 20%. There is no routing protocol present: each packet is forwarded along a precomputed shortest path. The average node density is 75 nodes per square kilometer. This density is 3 times higher than in the lattices, but is required to guarantee connectivity despite an irregular layout. The extra nodes do not increase capacity, since more nodes in the same area can only interfere with each other.

We expect the total capacity of the random network, as measured by one-hop throughput, to be similar to that of a lattice with horizontal and vertical traffic. In the random network scenario, packets are sent along paths with a wide distribution of lengths, but the use of one-hop throughput as the capacity metric accounts for path length. This makes it possible to compare the capacity of random networks with that of lattice networks.

Irregular placement leads to some areas of the universe having no nodes. This wastes potential spatial diversity and thus lowers capacity. Random choice of destinations also causes a tendency for more packets to be routed through the center of the network than along the edges. This traffic concentration means that the network as a whole is limited by the capacity of the center. The lattice configurations, in contrast, had traffic patterns that used all parts of the network evenly.

The "random" points in Figure 10 show how the simulated capacity of a random network with random traffic grows with increasing network size. The random network has somewhat less capacity than the lattices, though not dramatically less; the differences result from the factors mentioned above.

4. Scaling Ad Hoc Networks

The previous section presented a detailed analysis of the ability of each localized piece of an ad hoc network to forward traffic. This section takes a larger view, comparing a large network's total capacity with the load that the network's nodes might impose. The goal is to estimate the useful bandwidth that each node can expect for its *own* traffic. The analysis is based on scaling relationships: load increases with the number of nodes, load also increases with the distance over which each node wishes to communicate, and total one-hop capacity increases with the physical area covered by a network.

The total one-hop capacity of the network is determined by the amount of spatial reuse possible in the network. Given constant radio range, spatial reuse is proportional to the physical area of the network. Assuming that the node density δ is uniform, the physical area of the network, A, is related to the total number of nodes by $A = \frac{n}{\delta}$. Therefore, the total one-hop capacity of the network, C, should be proportional to the area, or $C = kA = k\frac{n}{\delta}$ for some constant k. Figure 10 shows that k is approximately 1 Mbps/km^2 for random network simulations.

Assume each node originates packets at a rate of λ. Further, assume the traffic pattern in the network has an expected physical path length of $\overline{L}$ from the source to the destination. This means that the minimum number of hops required to deliver a packet is $\frac{\overline{L}}{r}$ where r is the fixed radio transmission range. Hence the total one-hop capacity in the network required to send and forward packets obeys $C > n \cdot \lambda \cdot \frac{\overline{L}}{r}$. Combining this with $C = k\frac{n}{\delta}$, we have $k\frac{n}{\delta} > \frac{n\lambda\overline{L}}{r}$. Therefore, the capacity available to each node, λ, is bounded by

$$\lambda < \frac{kr}{\delta} \cdot \frac{1}{\overline{L}} = \frac{C/n}{\overline{L}/r} \qquad (2)$$

The above inequality tells us that as the expected path length increases, the bandwidth available for each node to originate packets decreases. Therefore, the traffic pattern has a great impact on scalability.

4.1 Random Traffic Pattern

The most common traffic pattern used in simulations of ad hoc networks has been random traffic: each source node initiates packets to randomly chosen destinations in the network. Below we show the expected path length $\overline{L}$ for such traffic.

Since a node chooses every node as its destination with equal probability, the probability that a node Y chooses a destination within x distance away is proportional to the number of nodes in the disc with center Y and radius x (We assume Y is at the center of the network, hence there is no need to worry about boundary effect. The expected path length calculated as such will be smaller). When node density is constant, the number of nodes is proportional to area of the disc with radius x, and thus proportional to x^2; this is the

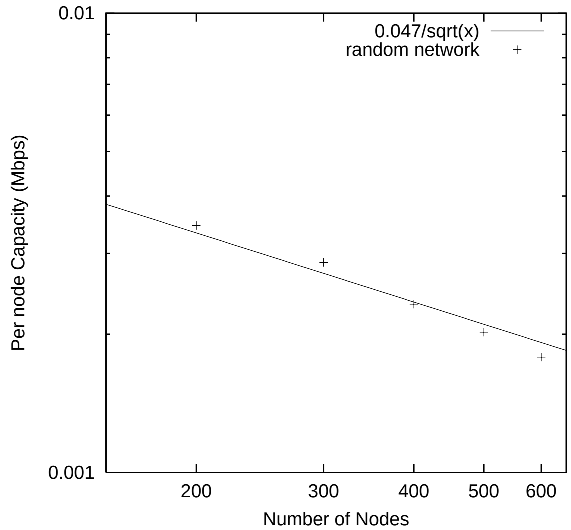

Figure 11: Log scale plot of simulated per node capacity with 1500-byte packets, as the number of nodes grows in a random network, and the fitted $O(1/\sqrt{n})$.

un-normalized cumulative distribution function (cdf) of the probability of a node communicating with a node at most x distance units away.

We know that the maximum distance is $\sqrt{A}$ for a square network with area A. Taking the derivative of the cdf and normalizing it, we get the probability density function (pdf) giving the probability of a node communicating with another node at distance x as

$$p(x) = \frac{x}{\int_0^{\sqrt{A}} t\, dt}$$

Therefore, the expected path length for a random traffic pattern is

$$\overline{L} = \int_0^{\sqrt{A}} x p(x)\, dx = \frac{2\sqrt{A}}{3}$$

When the node density is constant, the physical area of the network, A, is proportional to the number of nodes, n. Therefore, the capacity available to each individual node, λ is $O\left(1/\sqrt{n}\right)$.

Note that $O\left(1/\sqrt{n}\right)$ as derived above is an upper bound only. Gupta and Kumar [8] showed that it is possible to achieve a per node capacity of $\Omega\left(1/\sqrt{n \log n}\right)$, using global scheduling and near straight line routes. The $\log n$ factor is present because each node's radio transmission range needs to increase as $\log n$ in order for an ad hoc network to stay connected with high probability as the number of nodes increases. In their proof, the only place where global coordination is needed is the loose requirement that if a node has c interfering neighbors, then the node can occupy $\frac{1}{c+1}$ (i.e. some constant fraction) of total channel capacity for packet transmission and forwarding. We expect that this particular global scheduling requirement does not affect the asymptotic scaling behavior of the network when the 802.11 MAC is used instead. Also, since straight line routes resemble shortest paths (or geographic forwarding routes) in a dense network, we expect an ad hoc network with shortest path routing or geographic forwarding to agree with the theoretically achievable bound.

To show that the 802.11 MAC scheduling inefficiencies do not affect the scaling behavior, we simulated a network with random node positions and randomly chosen destinations. Packets are forwarded using pre-computed shortest path routes so we do not have to consider the overhead of any particular routing protocol. Figure 11 shows how well the capacity of the simulated network using 802.11 agrees with the asymptotic bounds on a log plot. Note that the asymptotic bound appears relevant for even small networks.

4.2 Traffic Patterns that Scale

From Equation 2, we can see that the expected path length of a particular traffic pattern determines capacity scaling. In this section, we investigate a number of concrete traffic patterns that might allow the per node capacity to scale well with the size of the network. In short, the less local the traffic pattern, the faster per node capacity degrades with network size.

The most obviously scalable traffic patterns are *exactly* local. That is, each node sends only to nodes within a fixed radius, independent of the network size. The expected path length clearly remains constant as the network size grows. Hence, the per node capacity with a local traffic pattern also stays constant. Another intuition behind this analysis is to observe that with only local traffic, we can view the entire network as consisting of "disconnected but overlapping" fixed-size sub-networks regardless of the actual network size. While connectivity across sub-networks may exist, the traffic does not use those connections.

Next we consider the class of traffic patterns with power law distance distributions. Specifically, the probability that a node communicates with a destination x distance units away is

$$p(x) = \frac{x^\alpha}{\int_\epsilon^{\sqrt{A}} t^\alpha\, dt}$$

A power law distribution is a convenient way to capture the gross features of how per node capacity scaling changes with the locality of the traffic pattern. The exponent in the power law is a "locality index" of sorts. For large negative α, destinations are clustered very closely to the sender. For large positive α, destinations are dispersed to the periphery of the network.

In order to make the pdf well-defined when $\alpha \le -1$, we introduce ϵ which is a non-zero minimum distance between any source-destination pair.

For such distance distributions, the average path length is

$$\overline{L} = \int_\epsilon^{\sqrt{A}} x \frac{x^\alpha}{\int_\epsilon^{\sqrt{A}} t^\alpha\, dt}\, dx$$

$$= \begin{cases} \frac{\alpha+1}{\alpha+2}\left(\frac{A^{\frac{\alpha+2}{2}} - \epsilon^{\alpha+2}}{A^{\frac{\alpha+1}{2}} - \epsilon^{\alpha+1}}\right) & \text{if } \alpha \neq -1, -2 \\ \frac{\frac{1}{2}\log A - \log \epsilon}{\frac{-1}{\sqrt{A}} + \frac{1}{\sqrt{\epsilon}}} & \text{if } \alpha = -2 \\ \frac{A^{\frac{1}{2}} - \epsilon}{\frac{1}{2}\log A - \log \epsilon} & \text{if } \alpha = -1 \end{cases}$$

When $\alpha < -2$ and A is large, the ϵ terms dominate the sums and $\overline{L}$ approaches $\frac{\alpha+1}{\alpha+2}\epsilon$. Hence if the distance distribution decays more rapidly than $\alpha = -2$, then the expected path length approaches a constant as the network size grows. This means per node capacity stays roughly constant.

When $\alpha = -2$, the expected path length scales as $O(\log A)$. So per node capacity in the network is $O(1/\log n)$. This result is relevant to the Grid Location Service (GLS) [10] whose location update traffic pattern is engineered to follow an $\alpha = -2$ power law.

Similar analysis shows that if $-2 < \alpha < -1$ and A is large, $\overline{L} = \frac{\alpha+1}{\alpha+2} A^{\frac{\alpha+2}{2}}$. The exponent $\frac{\alpha+2}{2}$ is a positive number when α is between 0 and $\frac{1}{2}$. When $\alpha = -1$, the expected length scaling becomes a mix of log and square root laws, and $\overline{L} = \frac{2A^{\frac{1}{2}}}{logA}$.

When $\alpha > -1$ the exponent on ϵ is positive. So we can set ϵ to 0 and $\overline{L} = \frac{\alpha+1}{\alpha+2} A^{\frac{1}{2}}$. This yields the interesting observation that any power law traffic pattern with $\alpha > -1$ scales basically the same way with network size as random traffic patterns. Thus a random traffic pattern is the most pessimistic traffic pattern one might assume for ad hoc networks. All of these traffic patterns will cause the per node capacity to decrease rapidly with network size.

As the power law distribution moves from a very local to a very distant destination selection, the capacity scaling moves from constant per node capacity to a $O(1/\sqrt{n})$ degradation of capacity with network size.

This leaves some hope for ad hoc networks. Some examples of networks with predominantly local traffic patterns are LAN users, the telephone system, and caching systems in the Internet at large.

5. Related Work

Gupta and Kumar [8] show that the per node capacity in an n-node random ad hoc network is $\Theta(1/\sqrt{n \log n})$, using a geometric analysis. They also show a global scheduling scheme which achieves that bound. In their work, a random communication pattern is assumed. Our work extends theirs by further considering the effects of different traffic patterns on the scalability of per node capacity. We also examine how ad hoc forwarding interacts with the 802.11 MAC and show that the use of 802.11 instead of a global scheduling scheme does not seem to affect the asymptotic bound on per node capacity.

Shepard [13] considers limits on capacity imposed by aggregate interference from many senders spread over a large area, concluding that such networks are scalable. He points out that capacity can be increased with minimum-energy routing, and proposes an efficient distributed channel-access technique. Our work, in contrast, focuses on the capacity likely to be available with the existing 802.11 channel access algorithm, which cannot easily support minimum-energy routing. We also focus on capacity limits imposed by multi-hop traffic patterns rather than by aggregate interference.

We assume that nodes are stationary. Grossglauser and Tse [6] consider ad hoc networks of mobile nodes, showing that *long term* per node throughput can stay constant in a network where node movement process is ergodic with a stationary distribution uniform over the network. The basic idea is for a source node to distribute packets to as many different nodes as possible; these nodes relay the packets to the final destination whenever they get close to the destination. Therefore, the expected path length remains constant. However, this result depends critically on the movement model. Furthermore, the fixed throughput guarantee is achieved only over very long time frames. This result, nevertheless, suggests a way to take advantage of node movement when sending packets from applications that can tolerate long delays.

Some existing studies have focused on the fairness of 802.11 in the context of ad hoc forwarding. Nandagopal et al. [12] propose an algorithm that gives each flow in the network a fair allocation of capacity no matter how much more contention it perceives in comparison to other flows. Luo et al. [11] propose an algorithm that, in addition to giving each flow its fair share, maximizes the total network capacity by giving more chances to flows whose transmissions cause less interference. The proposed algorithms might improve 802.11's efficiency in ad hoc forwarding.

6. Conclusion

This paper examines the capacity of wireless ad hoc networks via simulations and analysis from first principles. In particular, it studies 802.11 MAC interactions with ad hoc forwarding, their effect on network capacity, and the scaling behavior of per node capacity as networks grow bigger.

The ideal capacity of a long chain of nodes in isolation is $\frac{1}{4}$ of the raw channel bandwidth obtainable from the radio. The simulated chain capacity that the 802.11 MAC achieves with a greedy sender is about $\frac{1}{7}$, because nodes early in the chain starve later nodes.

We find that, in general, 802.11 does a reasonable job of scheduling packet transmissions in ad hoc networks. 802.11 is more efficient for orderly local traffic patterns, such as a lattice network with only horizontal flows. 802.11 is also able to approach the theoretical maximum capacity of $O(1/\sqrt{n})$ per node in a large random network of n nodes with random traffic.

We argue that the key factor deciding whether large ad hoc networks are feasible is the locality of traffic. We present specific criteria to distinguish traffic patterns that allow scalable capacity from those that do not.

7. References

[1] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: A Media Access Protocol for Wireless LAN's. In *Proc. ACM SIGCOMM Conference (SIGCOMM '94)*, August 1994.

[2] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols. In *Proc. ACM/IEEE MobiCom*, pages 85–97, October 1998.

[3] IEEE Computer Society LAN MAN Standards Committee. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. New York, New York, 1997. IEEE Std. 802.11–1997.

[4] S. Das, C. Perkins, and E. Royer. Performance Comparison of Two On-demand Routing Protocols for Ad hoc Networks. In *Proc. IEEE Infocom*, March 2000.

[5] Kevin Fall and Kannan Varadhan. *ns* Notes and Documentation. Technical report, UC Berkeley, LBL, USC/ISI, and Xerox PARC, November 1997.

[6] Matthias Grossglauser and David Tse. Mobility Increases the Capacity of Ad-hoc Wireless Networks. In *Proc. IEEE Infocom*, April 2001.

[7] CMU Monarch Group. CMU Monarch extensions to *ns*. `http://www.monarch.cs.cmu.edu/`.

[8] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, 46(2):388–404, March 2000.

[9] Brad Karp and H. T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *Proc. ACM/IEEE MobiCom*, August 2000.

[10] Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, and Robert Morris. A Scalable Location Service for Geographic Ad Hoc Routing. In *Proc. ACM/IEEE MobiCom*, pages 120–130, August 2000.

[11] Haiyun Luo, Songwu Lu, and Vaduvur Bharghavan. A New Model for Packet Scheduling in Multihop Wireless Networks. In *Proc. ACM/IEEE MobiCom*, pages 76–86, August 2000.

[12] Thyagarajan Nandagopal, Tae-Eun Kim, Xia Gao, and Vaduvur Bharghavan. Achieving MAC Layer Fairness in Wireless Packet Networks. In *Proc. ACM/IEEE MobiCom*, pages 87–98, August 2000.

[13] Timothy J. Shepard. A channel access scheme for large dense packet radio networks. In *Proc. ACM SIGCOMM Conference (SIGCOMM '96)*, pages 219–230, August 1996.