Routing Scalability in Multicore-Based Ad Hoc Networks

Ami Marowka
12 Anna Frank, Ramat-Gan,
52526, Israel
E-mail: amimar2@yahoo.com

Keywords: ad hoc networks, routing speedup, multicore, power management

Received: December 4, 2007

The integration of multicore processors into wireless mobile devices is creating new opportunities to enhance the speed and scalability of message routing in ad hoc networks. In this paper we study the impact of multicore technology on routing speed and node efficiency, and draw conclusions regarding the measures that should be taken to conserve energy and prolong the lifetime of a network.
We formally define three metrics and use them for performance evaluation: Time-to-Destination *(T2D),* Average Routing Speedup *(ARS), and* Average-Node-Efficiency *(ANE). The T2D metric is the time a message takes to travel to its destination in a loaded traffic network. ARS measures the average routing speed gained by a multicore-based network over a single-core based network, and ANE measures the average efficiency of a node, or the number of active cores.*
These benchmarks show that routing speedup in networks with multicore nodes increases linearly with the number of cores and significantly decrease traffic bottlenecks, while allowing more routings to be executed simultaneously. The average node efficiency, however, decreases linearly with the number of cores per node. Power-aware protocols and energy management techniques should therefore be developed to turn off the unused cores.

Povzetek: Narejena je analiza povezljivosti vozlov omrežja.

1 Introduction

The recent emergence of affordable dual-core processors in consumer products will overturn many currently accepted standards for software applications(5). The transition from single to multicore CPUs calls for the parallelization of all applications. Developers will be faced with the challenge of designing single-threaded applications that run efficiently on multiple cores.

Dual-core processors are only the beginning. Chip makers are currently working on the next generation of multicore processors, which will contain 4, 8 or 16 cores on a single die. According to the roadmap introduced by Intel(15), dual core processors are slated to reach mobile devices as well.

The integration of multicore processors into wireless communication and mobile computation devices will in general strengthen the communication infrastructure. Ad hoc wireless networks of mobile devices will also become more robust. An ad hoc network is a self-organized mobile network, whose every node is responsible for both computation and communication operations (1; 11).

In this paper we address the problem of how to measure the gain in routing speedup in ad hoc wireless networks where nodes are equipped with multicore processors, in particular when the network is heavily loaded. Moreover, we analyze the efficiency of multicore nodes in the network and show that the energy consumption of multicore nodes could be dramatically reduced by adapting existing methods and techniques.

We use location-based routing protocols in our traffic simulations(12). Location-based protocols are usually compared and analyzed by means of the "hops count" metric; i.e., the best single-path routing protocol is the one that finds a path from the origin node to destination node with the fewest number of hops. In real networks, however, many routing tasks are carried out at the same time. This is known in the literature as a multiple-sessions scenario, in which several routing sessions compete for the node's services. Each message wants to reach its destination node without waiting in a queue of routing sessions to be served. In such a scenario, we need a metric to quantify the arrival time of messages to their destinations.

This work therefore starts by defining a routing metric, called the *Time to Destination (T2D).* This metric will quantify the time required for a message to travel along the best path determined by the routing protocol. Then, two other metrics are defined based on the T2D: the *Average Routing Speedup* (ARS) and *Average Node Efficiency* (ANE). The first quantifies the average improvement in message travel time for a multicore network compared to a single-core network. The second measures the average node efficiency in a multicore network, compared to the efficiency of a network with single-core nodes.

These metrics serve to analyze the behavior of routings in our simulations of heavily loaded networks. We de-

rive conclusions regarding the benefits of message routing in multicore networks, and show how power-aware management protocols can reduce the consumption of energy. Moreover, we analyze the effect of mobility on the delivery success rate in multicore ad hoc networks and show that multicore nodes can improve the dependability of mobile networks. To the best of our knowledge, this is the first work in the open literature discussing routing protocols in ad hoc networks of multicore devices.

The design principles behind the speedup and the efficiency metrics present in this work were inspired from the parallel computing literature. However, our evaluation methods using scalable networks for measuring the *real* scalability of routing protocols are novel. In order to explore how protocols scale as the number of the nodes increases and as the average-node-degree increases, our simulator has the capability to generate extendable networks. In the open literature, the scalability is evaluated by measuring the performance of different number nodes such as 10, 20, 30,... where, for example, the 10-nodes network has different topology from the 20-nodes network. Our unique method for generating planar random networks enables us to generate 20-nodes network which is *extension* of the 10-nodes network and therefore to measure and to evaluate *real* scalability. In Section 4 we describe in detail this method, that to the best of our knowledge, we are the first to use. A work-in-progress versions of this paper was presented in (9; 10).

The remainder of the paper is organized as follows. Section 2 describes state-of-the-art works in the area of MIMO ad hoc networks. Section 3 defines the network model. In section 4, the metrics described above are defined. Section 5 details the simulations and analyzes the resulting data. Section 6 summarizes our results.

2 State-of-the-Art of MIMO MANET

Multiple-input Multiple-output (MIMO) wireless communication systems are the most promising multiple antenna technology today (16; 17). The advantages of MIMO communication, which exploits the physical channel between many transmit and receive antennas, are currently receiving significant attention (18). The integration of an air interface technology, such as MIMO, with a modulation scheme called orthogonal frequency division multiplexing (OFDM) (19) has the potential to lay the foundation for the data rate and capacity gains that will be needed for years to come. Since multiple data streams are transmitted in parallel from different antennas there is a linear increase in throughput with every pair of antennas added to the system. MIMO systems do not increase throughput simply by increasing bandwidth. They exploit the spatial dimension by increasing the number of unique spatial paths between the transmitter and receiver.

MIMO-OFDM combines OFDM and MIMO techniques thereby achieving spectral efficiency and increased throughput. A MIMO-OFDM system transmits independent OFDM modulated data from multiple antennas simultaneously. At the receiver, after OFDM demodulation, MIMO decoding on each of the sub-channels extracts the data from all the transmit antennas on all the sub-channels. The IEEE 802.16e standard incorporates MIMO-OFDMA.

The MIMO capability of antenna arrays has been studied at the physical layer and over a single link. There are few research directions that have studied MIMO in a multi-hop network from the perspective of higher layers. The research in (20) proposed a scheduling algorithm to offer fair medium access in a network where nodes are equipped with MIMO antennas. The model under study provides a simple abstraction of the physical layer properties of MIMO antennas. At the routing layer, a routing scheme to exploit MIMO gains is proposed (21). The idea is to adaptively switch the transmission/reception strategy using MIMO so that the aggregate throughput at the routing layer is increased. At each hop along a route this decision is made dynamically based on network conditions such as node density and traffic load.

At the transport layer, TCP performance over MIMO communications was studied (22). Focusing on the two architectures previously proposed to exploit spatial multiplexing and diversity gains (namely BLAST and STBC), the authors studied how the ARQ and packet combining techniques impact on the overall TCP performance. Their results indicate that, from the standpoint of TCP performance, the enhanced reliability offered by the diversity gain is preferable to the higher capacities offered by spatial multiplexing.

3 A Multicore Network Model

A highly realistic network model would take into account many complexities, such as the control traffic overhead, traffic congestion, mobility of the nodes, the irregular shape of radio coverage areas, and the intermittence of communication due to weather conditions and interference from preexisting infrastructure (power lines, base stations, etc.). Including *all* these details in the network model, however, would make it extremely complicated and scenario-dependent. This would hamper the derivation of meaningful and sufficiently general analytical results. It was shown that a simple parameterized model can accurately reflect the simulations (14). The model defined here, which is used in our simulations, therefore makes some widely accepted simplifying assumptions.

We formally define a multicore network model as follows:

Definition 1: A multicore network model is an undirected graph $G \equiv (V, E, D, M)$*, where:*

1. *V is the set of nodes;*
2. $E \subseteq VxV$ *is the set of undirected edges i.e.,* $(u, v) \in E$ *if u is able to transmit to v;*

3. D is the average node degree;
4. M is the number of cores in a node;
5. $P_{i,j}$ is the link probability of retransmission;
6. For $v \in V$, the node mobility is determined every pause time pau(v) by its velocity vel(v) towards destination des(v).

A realistic multicore model must reflects the potential cost of retransmissions required to recover from link errors. We use a linear transmission cost function: $T_{i,j} = \frac{\lambda L}{1-P_{i,j}}$ where a link is assumed to exist between node pair (i, j) as long as node j lies within the transmission range of node i, λ is the transmission rate, L is the message size and $P_{i,j}$ is the link packet error probability associated with that link. This cost function captures the cumulative delay expended in reliable data transfer, for both reliable and unreliable link layers.

Node mobility is based on the *random waypoint* parameterized model (2): a node chooses a destination uniformly at random in the simulated region, chooses a velocity uniformly at random, and then moves to that destination at the chosen velocity. Upon arriving at the chosen waypoint, the node pauses for a period before repeating the same process. In this model, the pause time represents the degree of mobility in a simulation; a longer pause time amounts to more nodes being stationary for more of the simulation.

The nodes communicate using omnidirectional antennas with maximum range r. We assume that all the nodes are equipped with identical transceivers, and further that each $M - core$ node is equipped with M transceivers and M antennas. Each core has multiple wireless channels and identical hardware and software mechanisms. A multicore node can hence transmit multiple packets at the same time to different nodes. The network is assumed to be homogeneous, consisting of nodes with the same transmission range, number of cores, and battery power. Imposing a common transmission range induces a strongly connected communication graph, often called a *unit disk graph* in the literature of routing protocols.

We assume that the traffic in the network is highly loaded, and that routing sessions are issued in a random manner. In other words, the origin and destination nodes of each routing are chosen randomly and no *a priori* knowledge is available regarding future sessions. Each node maintains a queue, so that incoming routing sessions are served on a FIFO basis. For simplicity it is assumed that the queue is large enough to store all arriving messages, so no messages are lost due to overflow. If the network consists of multicore nodes with M cores, then each node can serve M routing requests simultaneously.

The model assumes that different networks may have different ratios of computation time to communication time, and that communication and computation overlap. The communication time is the amount of time it takes for a packet transmitted by one node to be received by the next node on the path. The computation time is the amount of time it takes for a packet to be processed, from the time it is received by a node to the time it is transmitted to the next node. This interval includes computationally intensive functions such as signal processing, encoding and decoding, and encrypting and decrypting. It also includes relatively lightweight functions such as next-hop routing decisions and channel access delay.

Routing requests in our model are managed by localized, location-based protocols (12). In such protocols the location of the destination node is known, and the distance to neighboring nodes can be estimated on the basis of incoming signal strengths. The routes between nodes are created through a series of localized hop decisions; each node decides which neighbor will receive the message based on its own location, its neighboring nodes, and the message's destination. In our simulations we use the *Most Forward within Radius* (MFR) protocol (13), which forwards the message to the neighbor that maximizes its progress.

4 Routing Speedup and Efficiency

In measuring the scalability of parallel applications, the most commonly used practical metrics are *relative speedup* and *relative efficiency* (7; 8). *Relative Speedup* is the ratio of a parallel application's run time on a single processor to its run time on N processors; it is important to emphasize that the application and its test problem are identical in both situations. *Relative Efficiency* is the relative speedup divided by the number of processors N. Researchers use the speedup metric to check the performance of their applications on multiple platforms; it is a natural choice, since it is dimensionless and captures the relative benefit of solving a problem in parallel.

Motivated by these definitions, we define similar metrics for measuring the scalability of routing in heavily loaded, multi-session ad hoc networks. First, we define a time-based routing metric called the *Time-to-Destination* (T2D). Second, we define the *Routing-Speedup* (RS) and the *Average-Routing-Speedup* (ARS) metrics. Finally, we define the *Node-Efficiency* (NE) and *Average-Node-Efficiency* (ANE) metrics.

The metrics used in simulations of wireless ad hoc networks usually reflect the goal of the network design protocol. Most routing schemes thus use *hop count* as their metric, where hop count is the number of transmissions along a given route from source to destination. This choice agrees with the assumption that delay is proportional to hop count, which is reasonable when the impact of congestion is not significant. However, this assumption is not warranted for realistic ad hoc network scenarios in which routing sessions are issued from many and various origin nodes simultaneously and to random destinations. We therefore need a metric that reflects the delay caused by traffic congestion in a wireless network.

We formally define the Time-to-Destination metric as follows:

Definition 2: Time-to-Destination (T2D).
Assume a unit disk multicore graph G, and a route

$R = v_s, ..., v_d$ *from source node* v_s *to destination node* v_d. *The Time-to-Destination (T2D) is the aggregate time taken to route a message from the origin node to the destination node.*

The T2D metric is also known in the open literature as the end-to-end latency. We assume that different networks have different ratios of computation time to communication time, and that communication and computation overlap. Thus, in the case of a routing with delay times $d_s, ..., d_d$ at each hop the total time it takes for a message to traverse the route is given by

$T2D = L * (T_{comm} + T_{comp}) + \sum_{i=s}^{d} d_i$,

Where T_{comm} and T_{comp} are the total communication time and total computation time respectively. L is the length of the route, the number of hops from the source node to the destination node.

Definition 3: Routing Speedup (RS).

Given a unit disk multicore graph G and a route $R = v_s, ..., v_d$ *from source node* v_s *to destination node* v_d, *the Routing Speedup (RS) of* R *is the ratio of message routing time for a network with single-core nodes to the message routing time for a network with* M*-core nodes:*

$RS = \frac{T2D_1}{T2D_m}$,

Where the subscripts 1 and m indicate the single-core and M-core networks respectively. Since it is more practical to measure the average speedup in a scenario with multiple sessions, we also define the *Average Routing Speedup* as follows:

Definition 4: Average Routing Speedup (ARS).

The Average Routing Speedup (ARS) is the average speedup over R *routings in G:* $ARS = \frac{1}{n} \sum_{i=1}^{n} RS_i$.

The ARS metric is a practical tool for evaluating the speed gained by moving from an ad hoc network with single-core nodes to one with multicore nodes. ARS does not, however, measure the efficiency of a multicore node. (By node efficiency, we mean the average number of cores that are busy per node.) As will be shown in the next section, knowing the node efficiency permits a dramatic reduction of power consumption in each node and thus increases the lifetime of the whole network.

Definition 5: Node Efficiency (NE).

Assume a unit disk multicore graph G, and R *routings executed in G over a period* Δt. *The Node Efficiency (NE) of a given node during* Δt *is the ratio of* T^1_{comp}, *its aggregate computation time in the case of a single-core network, to* M *times* T^m_{comp}, *its aggregate computation time in the case of an* M*-core network.*

$NE = \frac{T^1_{comp}}{M * T^m_{comp}}$,

For the same practical reasons mentioned with respect to the RS and ARS metrics, we define the *Average Node Efficiency* as follows:

Definition 6: Average Node Efficiency (ANE).

The Average Node Efficiency (ANE) in G of $|V| = n$ *nodes during time* Δt *is:* $ANE = \frac{1}{n} \sum_{i=1}^{n} NE_i$.

It is important to notice that NE and ANE do not take into account idle times, only those times when the nodes are involved in the routing process. The aim of these metrics is to measure the efficiency of the cores within a node, rather than a given node's dominance over other nodes in the routing process. Although this information is important for intelligent power management and energy conservation, the efficiency metrics we define here focus on what happens inside a multicore node.

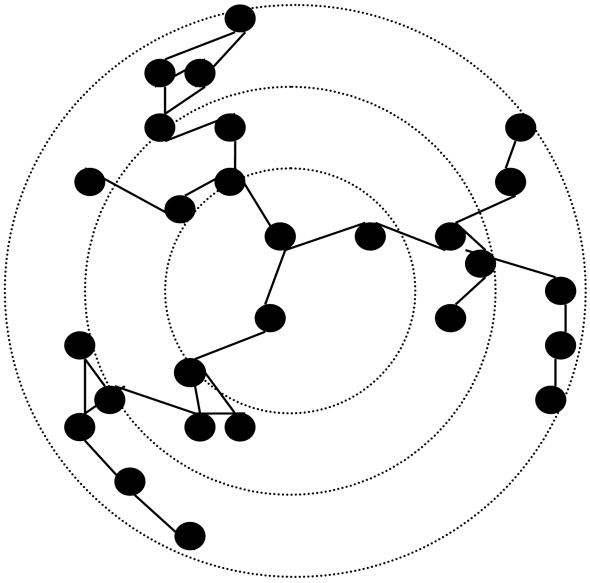

Figure 1: Illustration of Scalable Planar Network of 27 nodes with c=3.

In the next section we use these metrics to evaluate and analyze the implications of multicore nodes on position-based routing protocols and power management strategies in ad hoc networks.

5 Simulator and Simulations

A discrete event simulator was developed in order to monitor, observe, and measure ARS and ANE in multicore ad hoc networks. We generated a database with hundreds of random unit graphs, with values of V and D spanning a wide range of sparse and dense networks. The results shown in this paper are only a representative sample of the many simulations performed, and each result is averaged over many runs.

The network generator first partitions the plane into k regions: an innermost disk whose radius is equal to the transmission range r, and a series of concentric annuli of width r surrounding the disk. The number of nodes in each annulus is proportional to its area. If c nodes are randomly located in the inner disk of a network with k regions, then a total of $c * k^2$ nodes will be randomly placed in the entire network. For example, if the innermost region of a network has 3 nodes (c=3) then 9, 15, and 21 nodes will be located in the

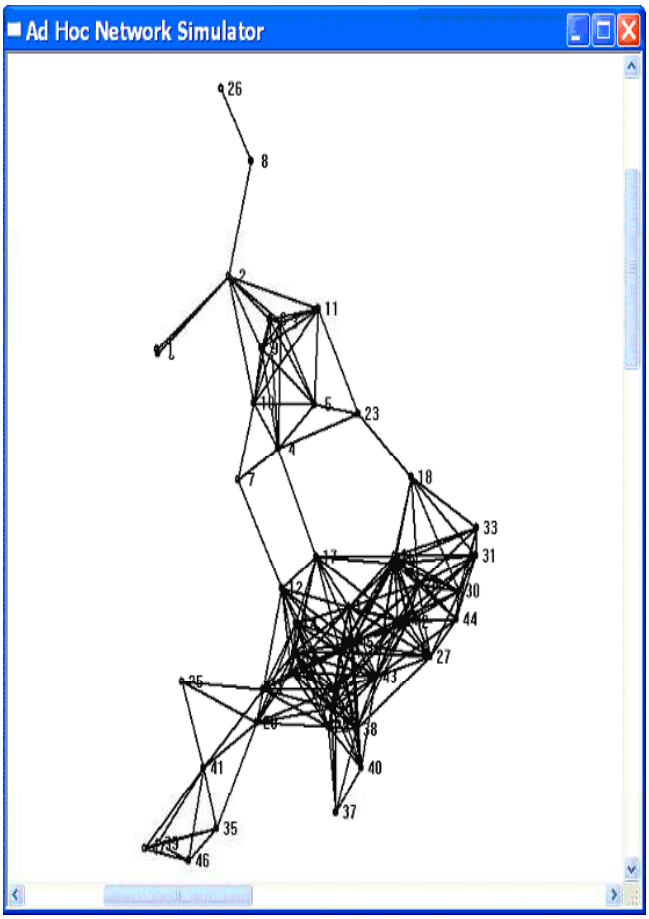

Figure 2: A randomly generated 4-regions network of 48 nodes and average node degree of 4.

first three rings respectively. Figure 1 illustrates a network of 27 nodes for c=3. The networks are thus generated incrementally, ring after ring. A network with k regions is just an extension of the network with $(k-1)$ regions. In this way we can calculate the scalability of our protocols exactly.

For small networks (up to 50) we choose the value of c to be 3 and for large networks the value was 4. The networks were extension of a base network of 27 nodes (in case of c=3) or 36 nodes (in case of c=4). The average-node-degree of the base network was preserved in the extended networks. Usually, the topology of randomly generated scalable networks are not symmetrical like the illustration shown in Figure 1. An example of a randomly generated 4-region network of 48 nodes and average-node-degree of 4 is shown in Figure 2.

All the simulations shown in this paper were carried out on a network of 108 nodes, with an average node degree of 7. Measurements were performed for three traffic loads: 100, 1K and 10K routings. The source and destination nodes of each route were randomly chosen, and all routings were issued simultaneously. The scalability of routing was tested assuming values of 2, 4, 8, and 16 cores per node. Each simulation shown in this section (except Fig. 5) was performed assuming that the ratio of computation time to communication time is 1. The routing protocol used in our benchmarks is the *Forward within Radius (MFR)* protocol.

For the simulations of our multicore network model we used parameter values that are the average values of the IEEE 802.11-based interfaces as appeared in (4) and summarized in table 1. Each core is assumed to be able to transmit and receive 2Mbps. The packet size was of 64KB and the link probability of retransmission was set to 0.25.

For the network mobility it was assumed that all nodes move according to the random waypoint model. First, a node chooses a destination uniformly at random in the simulated region. The area $A(N,R)$ of a simulated region is determined relative to the number of nodes in the network ($N = c * k^2$) and the data transmission range R. Thus, $A(N,R) = (k*R)^2 * \pi = (N/c) * R^2 * \pi$. For example, a scalable network of 48 nodes with $c = 3$ has 4 rings and thus the area of the network region is $(4R)^2\pi$. In the simulations we used data transmission range of 250m. Next, the node chooses a velocity uniformly at random, with a maximum velocity of 10 m/s, and moves to that destination at the chosen velocity. Upon arriving at the chosen waypoint, the node pauses for a period before repeating the same process. We simulate pause times in the range of 0-10 seconds.

Table 1: Parameter values used in the simulations.

Communication	
Packet size	64KB
Retransmission probability	0.25
Mobility	
Simulated area	$(N/c) * R^2 * \pi$
Transmission range	250m
Velocity	0-10m/s
Pause time	0-10s

Figures 3 and 4 depict the routing scalability that can be expected from a multicore-based network. Figure 3 plots ARS as a function of the number of routings for nodes with 2, 4, 8 and 16 cores. Figure 4 plots ARS as function of the number of cores per node for traffic loads of 100, 1K and 10K routings. We choose to present the same information in two different ways to make all the relationships clearly visible.

Analysis of these results leads to the following findings. First, ARS increases as the number of the routings increases; this is obvious from Figure 3. For 10K routings the ARS increases linearly with the number of cores, up to values of 1.96, 3.82, 7.26 and 13.27 for 2, 4, 8 and 16 cores respectively. For 100 routings, however, the ARS reaches its maximum value of 1.35 at only 4 cores (Figure 4). Adding more cores does not increase the ARS unless there is more traffic to handle. These encouraging results show that mul-

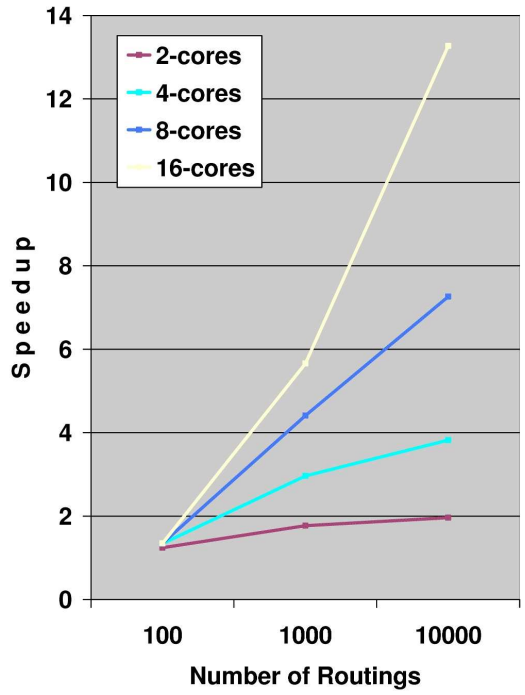

Figure 3: Graphs of Average-Routing-Speedup as function of number of routings.

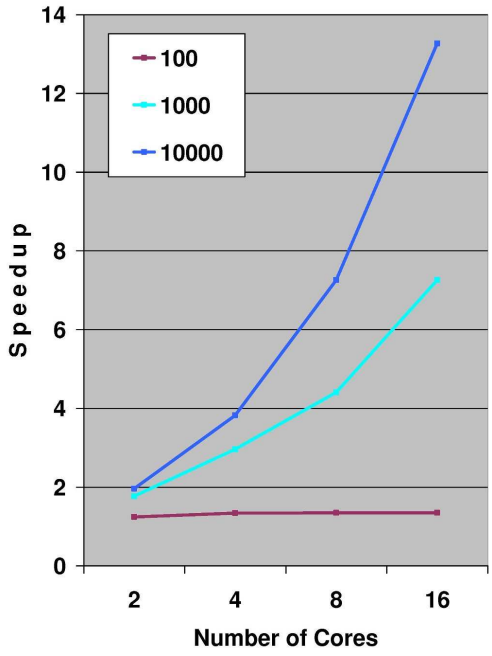

Figure 4: Graphs of Average-Routing-Speedup as function of number of cores.

ticore nodes decrease the congestion of loaded networks, increase the routing speedup, and decrease the travel time of messages.

Figure 5 plots the ARS as function of the number of cores per node and the ratio of computation time to communication time. The goal of this paper is to study the impact of various multicore ad hoc networks on routing scalability, so we varied the ratio of computation time to communication time in the range 0.5 to 3. Delays in computation and communication can be incurred from many sources: next-hop routing decisions, channel access delays, transmission delays, traffic load, the sizes of contending packets, the medium access control algorithm used by the nodes, the modulation and symbol rate of the packets, and finally the distance that the packets must travel. Moreover, wireless communication usually requires several network-dependent, computationally-intensive functions such as signal processing, encoding and decoding, and encrypting and decrypting. Figure 5 shows that as the ratio of computation time to communication time increases, the degree of speedup for large core numbers improves. In the case of 16 cores the ARS increases by 34% as the computation time to communication time ratio is varied in the range 0.5 to 3. In the case of 8 cores the improvement is only by 8% and for 2 and 4 cores there is no improvement at all. However, in the case of 2, 4 and 8 cores the ARS is high. This phenomenon matches also the results shown in Figure 8. The network reaches its load balancing equilibrium point when the number of cores reaches 16. At this point each node has enough available cores to amortize the increase in the ratio of computation time to communication time.

Figures 6, 7 and 8 describe the node efficiency of multicore-based networks. Figure 6 graphs ANE as a function of core number for the cases of 100, 1K and 10K routings. Figure 6 graphs ANE as a function of the number of routings for the cases of 2, 4, 8, and 16 cores. Once again, we choose to present the same information in two different ways to clarify the relations. Figure 8 is a histogram showing the distribution of routing loads over all nodes for the case of 10K simultaneous, random routings.

Analysis of these results reveals the following relationships. The ANE asymptotically increases with the number of routings (Figure 6). For example, in the case of 8-core network ANE approaches the values 0.18, 0.54 and 0.70 for 100, 1K, and 10K routings respectively. However, the ANE decreases dramatically as the number of cores increases (Figure 7). For example, in the case of 10K routings ANE approaches the values 0.88, 0.78, 0.70 and 0.63 for 2, 4, 8 and 16 cores respectively. In other words, as the number of cores increases more cores will remain idle.

Figure 8 presents another view of this phenomenon. This histogram shows how the routing load is distributed among the nodes. For example, in the case of a single-core network most nodes (80 of 108) were not busy at least 40% of the time (marked *low* in the histogram). 18 nodes were busy between 40% to 80% of the time (marked *moderate*), and 10 nodes were busy more then 80% of time (marked *high*). These 10 nodes are the *dominant set* of the network, through which most of the traffic passes. However, as the number of cores increases more nodes become busy more

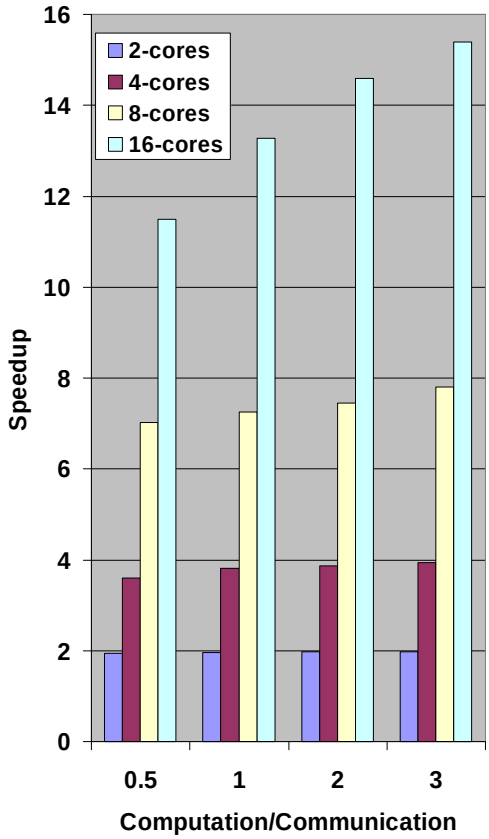

Figure 5: Graphs of Average-Routing-Speedup as function of number of cores and computation to communication ratio.

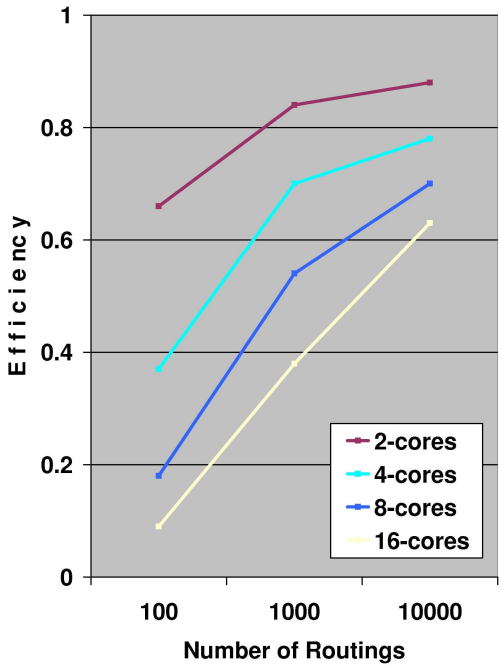

Figure 6: Graphs of Average-Node-Efficiency as function of number of routings.

of the time. This indicates improved load balancing and decreased traffic congestion. For example, in the case of 16 cores only 42 nodes are idle most of the time, 28 nodes are busy about half of the time, and 38 are busy most of the time.

These observations have important implications for the power management strategies that should be taken to reduce energy consumption in multicore-based networks. Many design techniques for reducing the power consumption of mobile devices have been introduced over the last decade (4; 6; 3). Among these are technologies which enable the processor to operate at multiple voltages and frequencies depending on the workload required by the user. Thus, when workload drops the processor steps down to a lower voltage and frequency, conserving battery power. Processors including a power-optimized system bus with a low-power L2 cache, which turn off parts of the high-speed memory when it isn't needed, also result in an overall reduction of the power consumption. A *minimization of capacitance* can also be achieved by relying on chip-based resources such as caches and registers.

Our measurements of Average Node Efficiency show that it is essential to develop power-aware protocols and complementary hardware capabilities that will permit a multicore node to dynamically adjust the number of active cores. As can be seen from Figure 6, a 16-core network under traffic loads of 10K and 1K routings has ANE values of 0.6 and 0.4 respectively. In other words, energy savings of up to 40% to 60% could be achieved by allowing individual cores to be turned off or put in a sleep mode.

Figure 9 and Figure 10 depict the effect of the network topology characteristics on the routing scalability. Figure 9 plots the speedup as function of number of nodes, 16 cores each, for 100, 1K and 10K routings. It can be observed from the graphs that for high traffic load (10K routings) the speedup decreases from 15.4 to 13.27 when the number of nodes increases from 27 to 108 respectively. Since low traffic load leads to low speedup, as shown in Figure 3, increasing the number of nodes, for the same global traffic load, decreases the local traffic load in each node and thus decreases the speedup. For low traffic loads (100 and 1K routings) the impact of the number of nodes on the speedup is marginal. Figure 10 plots the speedup as function of average-node-degree for network of 108 nodes, 16 cores each, and for 100, 1K and 10K routings. Since low average-node-degree increases the traffic load in the nodes, it is expected that the speedup will increase as well. Figure 10 shows that for high traffic load (10K routings) the speedup increases from 12.2 to 14.8 when the average-node-degree decreases from 9 to 4 respectively as expected. For traffic load of 1K routings the speedup increases slightly and for 100 routings the speedup remains 1.35 for 4, 7 and 9 degrees.

Figure 11 depicts the effect of mobility on the delivery

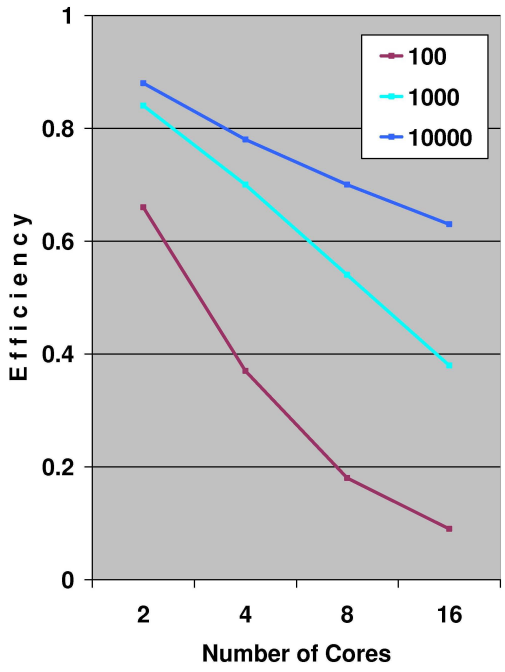

Figure 7: Graphs of Average-Node-Efficiency as function of number of cores.

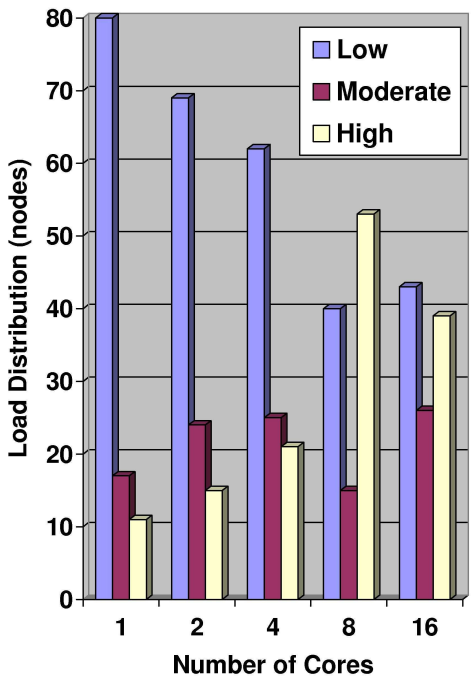

Figure 8: Histogram of routings load distribution. The nodes are grouped into three load levels: low (nodes that were busy between 0% and 40% of the time), moderate (40%-80%) and high (80%-100%).

success rate in multicore ad hoc networks. Figure 11 plots the routing delivery success rate as a function of pause time for 1, 2, 4, 8 and 16 cores per node. The delivery success rate decreases as the number of cores per node decreases since less cores means higher traffic load. High traffic load increases the time-to-destination a message travels and thus increases the possibility that the message will not reach its destination due to nodes mobility. For example, for the case of pause time of 0 (continuous mobility) the delivery success rate increases from 0.75 to 0.85 when the number of cores per node increases from 1 to 16 respectively. Moreover, increase in the number of cores per node may achieves the desired delivery success rate although the mobility rate increases. For example, for the case of pause time of 10 seconds and single core nodes, the success delivery rate is 0.85. Increasing the mobility rate, by decreasing the pause time to 5 seconds, and using 8 cores per node, instead of one, yield success delivery rate of 0.89. The conclusion arises from these results is that multicore nodes decrease the the number of messages that do not reach their destination due to mobility and thus increase the dependability of ad hoc networks.

6 Conclusions

We have studied the effects of introducing multicore nodes on the routing performance of wireless ad hoc networks. First, we formally defined a multicore-based network model and three metrics as evaluation tools: Time-to-Destination, Routing Speedup, and Node Efficiency. Next we evaluated the routing speedup over a wide range of network configurations through intensive routing simulations.

We discovered that a network with more cores in each node has a larger routing speedup and handles more routings efficiently. Multicore networks decreased traffic congestion, balanced the load among the nodes and improve the dependability of mobile networks. However, there are side effects that must still be resolved. Adding more cores also decreases node efficiency; not all the node's cores are used all the time. This phenomenon calls for the development of better power-saving protocols and energy management strategies in order to conserve battery power.

References

[1] S. Basgani, M. Conti, S. Giordano, and I. Stojmenoviv, *Mobile ad hoc networking*, IEEE Press, 2004.

[2] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, *A performance comparison of multi-hop wireless ad hoc network routing rotocols*, ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '98), August 1998.

[3] R. Chary and J. Gilbert, *Intel pca: Advanced power management for the mobile platform*, Intel Developer UPDATE Magazine, August 2002.

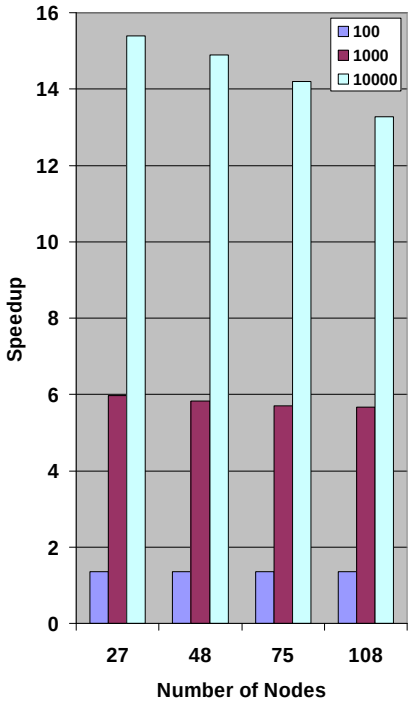

Figure 9: Graphs of Average-Routing-Speedup as function of number of nodes for 16-cores nodes and for 100, 1K and 10K routings requests.

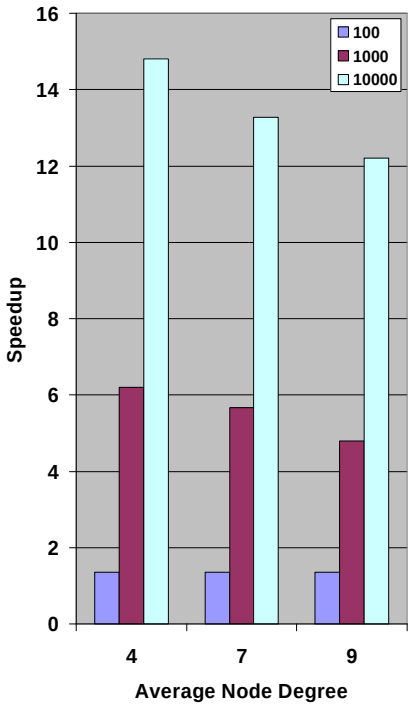

Figure 10: Graphs of Average-Routing-Speedup as function of average-node-degree for network of 108(16) nodes(cores) and for 100, 1K and 10K routings requests.

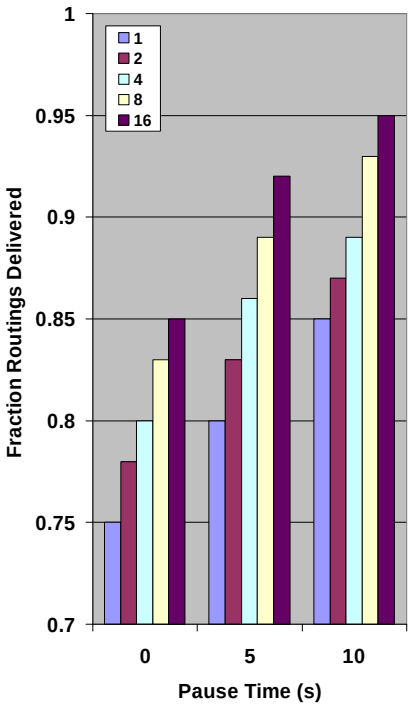

Figure 11: Routing delivery success rate as a function of pause time for 1, 2, 4, 8 and 16 cores.

[4] L. M. Feeney, *Energy-efficient communication in ad-hoc wireless networks*, In Mobile Ad Hoc Networking Editors S. Basagni, M. Conti, S. Giordano and I. Stojmenovic, pages 301–327, IEEE-Wiley, 2004.

[5] D. Geer, *Chip makers turn to multicore processors*, IEEE Computer, May 2005.

[6] P. J. M. Havinga and G. J. M. Smit, *Design techniques for low-power systems*, Journal of Systems Architecture, 46(1):1–21, 2000.

[7] K. Hwang and Z. Xu, *Scalability and speedup analysis*, Scalable Parallel Computing, McGraw Hill, pages 134–148, 1998.

[8] V. Kumar, A. Grama, A. Gupta, and G. Karipis, *Performance and scalability of parallel systems*, Introduction to Parallel Computing, Benjamin/Cummings, pages 117–146, 2004.

[9] A. Marowka, *Routing speedup and node efficiency in multicore ad hoc networks*, 4th International Conference on Performance Modeling and Evaluation of Heterogeneous Networks (HET-NETs'06), pages P02/1–P02/11, September 2006.

[10] A. Marowka, *Routing speedup in multicore-based ad hoc networks*, 6th International Symposium on Parallel and Distributed Computing (ISPDC'07), July 2007.

[11] C. Siva, R. Murthy, and B. S. Manoj, *Ad hoc networks, architecture and protocols*, Prentice Hall, 2004.

[12] I. Stojmenovic, *Position-based routing in ad hoc networks*, IEEE Communications Magazine, July 2002.

[13] H. Takagi and L. Kleinrock, *Optimal transmission ranges for randomly distributed packet radio terminals*, IEEE Transactions on Communications, 32(3):246–257, 1984.

[14] M. Viennot, T. Clausen, and P. Jacquet, *Analyzing control traffic overhead versus mobility and data traffic activity in mobile ad-hoc network protocols*, ACM Wireless Networks journal, 10(4), July 2004.

[15] U. Weiser, *Future direction in microprocessors*, In: Intel ŚTechnology in MotionŠ conference, May 30 2005, Ramat-Gan, Israel.

[16] E. Biglieri, R. Calderbank, A. Constantinides ,A. Goldsmith, A. Paulraj, H. Poor, *MIMO Wireless Communications*, Cambridge University Press, 2007

[17] C. Oestges, B. Clerckx, *MIMO Wireless Communications : From Real-world Propagation to Space-time Code Design*, Academic, 2007.

[18] D. Gesbert, M. Kountouris, R. Heath, C. Chae, T. Salzer, *Shifting the MIMO Paradigm: From Single User to Multiuser Communications*, IEEE Signal Processing Magazine, vol. 24, no. 5, pp. 36-46, Oct., 2007.

[19] A. Bahai, B. Saltzberg, M. Ergen, *Multi Carrier Digital Communications: Theory and Applications of OFDM*, Springer, 2004.

[20] K. Sundaresan, R. Sivakumar, M. Ingram, T. Chang, *A fair medium access control protocol for ad-hoc networks with MIMO links*, In IEEE Infocom, 2004.

[21] K. Sundaresan and R. Sivakumar, *Routing in ad-hoc networks with MIMO links*, In IEEE ICNP, 2005.

[22] A Toledo and X Wang, *TCP performance over wireless MIMO channels with ARQ and packet combining*, IEEE Transactions on Mobile Computing, 5(3):208-223, 2006.