Poster Abstract: Throughput-Delay Tradeoff in Small and Sparse Mobile Ad hoc Networks*

Swapnil Bhatia **Radim Bartoš** **Kevin Ma**
sbhatia@cs.unh.edu *rbartos@cs.unh.edu* *kjma@cs.unh.edu*
Department of Computer Science, University of New Hampshire, Durham, NH, USA

This paper takes the first step in characterizing the throughput-delay tradeoff for small and sparse MANETs which have many practical applications. We find that as the MANET becomes sparser, throughput decreases and delay increases, as expected. If relaying is disabled then the throughput and delay depend on the size of the area of operation. While relaying does increase throughput, the single packet relaying strategy worsens the delay for small MANETs in the Grossglauser traffic model. Greedy relaying overcomes this worsening without trading throughput, but only for rapidly mixing mobility. Unlike in dense networks, local broadcasting does not provide any significant benefit. Packet repetition does decrease delay, but only at the expense of reduced throughput. Our results are useful in practical underwater MANETs which are typically small and sparse.

I. Motivation

The characterization of the throughput-delay tradeoff in wireless ad hoc networks has been the subject of study in a number of papers in recent years [1–3]. Most of the previous work in this area, except that of Spyropoulos et al. [1], has focused on dense wireless networks with the tradeoff being studied as the number of network nodes n goes to infinity. A fundamental assumption in such work is that the wireless network under study is sufficiently dense, with the scaling behavior under increasing density being the subject of study. A motivating example justifying this assumption is the ad hoc sensor network where a dense deployment of sensor nodes is desirable. In contrast, the practical deployment scenario for many wireless ad hoc networks, particularly those involving mobile nodes, is such that while a dense deployment is desirable, it is rarely feasible. Consider a MANET of autonomous underwater vehicles (AUVs) deployed for bathymetry or underwater surveillance. Even for such basic underwater missions, the oceanic region involved is far too vast to be amenable to sensing and measurement by a dense MANET. As a result, practical AUV MANETs tend to be small and sparse for which extant capacity results studying scaling behavior as a function of increasing density provide little insight into the tradeoffs involved in such networks. A fundamental differentiating characteristic of a sparse MANET is the high probability with which a mobile node may be outside the transmission range of any other node. Whereas the interference among concurrent transmissions plays a deciding role in the throughput-delay tradeoff in dense networks, such interference is rare in sparse networks. In what other respects might sparse MANETs be different from dense ones? This is the motivating question for our work and this short paper takes the first step towards answering it.

*This research was supported in part by grant N00014-05-1-0666 from the U.S. Office of Naval Research. This work was presented as a poster at ACM MobiHoc 2007, Montreal, Canada.

II. Model

We model the spatial region in which the mobile nodes of a sparse MANET move as a discrete undirected graph with loops. We experiment with two graphs: the complete graph on m^2 vertices and the $m \times m$ two-dimensional torus. We consider $n \geq 2$ mobile nodes performing a random walk on the underlying graph at each discrete time step. Each mobile node i produces data packets destined for exactly one other node denoted $\mathrm{dest}(i)$. This source-destination mapping is fixed and is chosen by a random derangement of $\{1, \ldots, n\}$. At any time step, a node has exactly one packet available for transmission. After a packet has been transmitted, a new packet is available for transmission immediately as in the traffic model of Grossglauser et al. [3]. If a set M_v of (more than one) mobile nodes meet at any vertex v, then data transmission occurs according to the following rules:

1. Every mobile node $i \in M_v$ such that $\mathrm{dest}(i) \in M_v$, transmits a single packet to $\mathrm{dest}(i)$. We call this a *direct delivery*.

2. Every node $i \in M_v$ that could not perform a di-

rect delivery chooses at random a $j \in M_v$ for which it carries one or more packets delegated to it by j's source. It then delivers at most p such packets to j, where the particular packets transmitted, if more than p are available, are also chosen at random. We call this a *relayed delivery*.

3. Every node $i \in M_v$ that could not perform a direct or relayed delivery transmits exactly one packet to another randomly chosen node $j \in M_v$ requesting j to deliver the packet to $\mathrm{dest}(i)$. We refer to this as *packet delegation.*

When $|M_v| > 1$, each node either makes a direct or relayed delivery or delegates a packet. Each node possesses infinite space for storing delegated packets that it has accepted. Transmissions occur in a round-robin manner and are coordinated through some TDMA scheme at each vertex v. Transmission of a single packet takes a constant amount of time and the total time spent in communication at each vertex is negligible in comparison to the inter-vertex travel time. Since the network is sparse, transmissions occur concurrently at all vertices v without mutual interference.

The above rules are similar to those used in Grossglauser et al. [3]. In addition, we study the following variants. When *delegation* is disabled, a node is only capable of direct delivery via method (1) above. When delegation is enabled and $p = \infty$ in method (2), we call this *greedy relaying*. If *local broadcasting* is enabled, then a packet transmitted by a node i via method (3) is broadcasted to all nodes $j \in M_v$, i.e., i delegates the packet to all other nodes present at v through a single transmission [2]. If *packet repetition* is enabled with parameter r, then every packet produced by a node i is delegated $1 + r$ times by i or until it is delivered directly, whichever occurs earlier.

III. Simulation results

We simulated our model of a sparse MANET on a complete graph and a torus and measured the average throughput and delay. We adopt a natural definition for sparsity; it is the difference in the orders of magnitude of the number of mobile nodes (n) and the number of vertices in the underlying graph (m^2).

Figure 1 and Figure 2 show the throughput-delay tradeoff on a complete graph and the torus for a fixed $m = 100$ with n varied between 2 and 500. We can make the following observations, the most striking of which is that while packet delegation increases throughput, it worsens delay, for high sparsity. This is true for both the complete graph as well as the torus. Moreover, whereas greedy relaying mitigates this rise

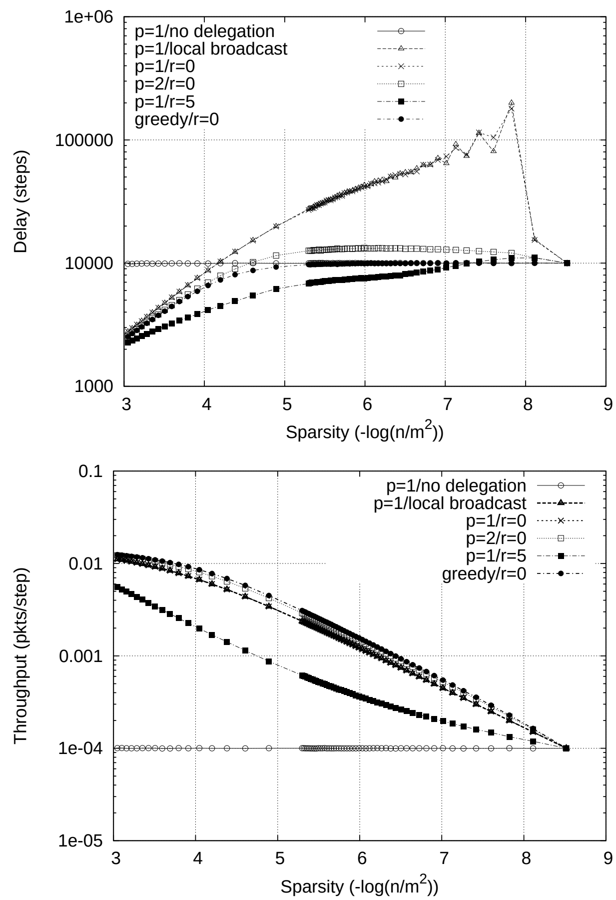
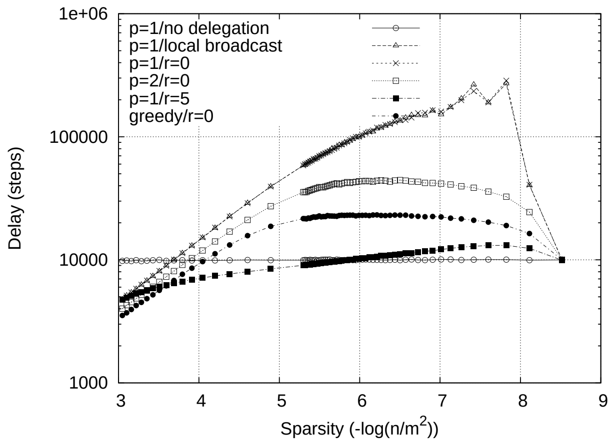


Figure 1: MANET on a complete graph when n is varied with $m = 100$.

in delay for the complete graph, it is ineffective on the torus. Except for the delay when delegation is disabled, the delay for the torus is higher than that for the complete graph when the sparsity is high.

While throughput improves, the average delay when relaying is enabled is worse than when it is disabled. This is mainly an artifact of using the Grossglauser traffic model. In this model, a node has a new packet available for transmission immediately after it has transmitted the previous one. When relaying is disabled, only the source can transmit packets to its destination. Thus, a new packet becomes available only and immediately after the source delivers its previous packet. When relaying is enabled, a source can generate a new packet as soon as it has delegated the previous one to a relay. Thus, a larger number of packets may wait in the queue of various relays. Moreover, if $p = 1$, then a relay carrying packets for $\mathrm{dest}(j)$ can only deliver one packet to $\mathrm{dest}(j)$ per meeting, as in the Grossglauser model. Furthermore, the chance that a packet for $\mathrm{dest}(j)$ will be chosen via method (2) is lowered if the relay has packets for several destinations which it meets at the same time. Thus, a relay carrying packets for $\mathrm{dest}(j)$ may have to meet $\mathrm{dest}(j)$



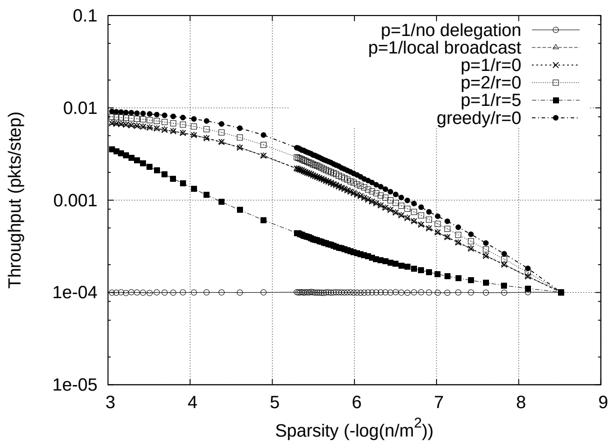


Figure 2: MANET on a two-dimensional torus when n is varied with $m = 100$.

multiple times to deliver all the waiting packets that it has queued up for it. This type of queuing is impossible when relaying is disabled: a packet's waiting time starts only when it is generated and it is generated only when the previous packet has been delivered.

It is no surprise that greedy relaying is effective in mitigating the rise in delay discussed above. With greedy relaying, a relay can deliver all the packets in its queue to $\mathrm{dest}(j)$ in a single meeting, provided that $\mathrm{dest}(j)$ is randomly chosen for delivery via method (2). However, whereas greedy relaying lowers the delay to that achieved without relaying on the complete graph, it is not as effective on the torus. The delay for the torus is also higher than that for the complete graph. The mixing and meeting time of the torus is $\Theta(m^2)$ and $\Theta(m^2 \log m^2)$ respectively while that of the complete graph is $\Theta(1)$ and $\Theta(m^2)$ respectively [4]. Because of this, the time to meet a node's destination is also higher on the torus. Also, it is more likely on a torus that a node will meet the same relay multiple times in quick succession, each time delegating a new packet to it. Because of this, packets are not spread out uniformly across many different relays. Hence, the fate of many packets may end up relying on the meeting time of a few relay nodes. These factors conspire together to increase the average number of packets carried by a relay on the torus for a particular destination, as compared to the number carried by a relay on the complete graph. Thus, a larger number of packets accrue the inherently higher delay of the torus due to its meeting and mixing time, and this leads to a higher delay despite greedy relaying. Note that when the delay is higher, the corresponding throughput observed on the torus is also slightly higher than the throughput on the complete graph.

Packet repetition is highly effective in decreasing the delay because it increases the likelihood that packets are spread uniformly across a larger number of different relays. This decrease comes at the price of lowered throughput. Local broadcast has no discernible effect on delay or throughput because $|M_v| > 2$ necessary for local broadcast to occur is a rare event in sparse MANETs.

IV. Conclusions and future work

In this short paper, we reported our results on the throughput-delay tradeoff for sparse MANETs which show that the scaling behavior of small and sparse networks is different from dense networks studied in the past. What is the effect of a traffic model that is more natural than the Grossglauser model used here? What kind of motion, graph, and traffic model captures practical AUV missions? We intend to investigate such questions about sparse networks more deeply in our future work.

References

[1] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks," in *Proc. of ACM SIGCOMM WDTN-05*, Philadelphia, PA, Aug. 2005.

[2] R. M. de Moraes, H. R. Sadjadpour, and J. Garcia-Luna-Aceves, "On Mobility-Capacity-Delay Trade-off in Wireless Ad Hoc Networks," in *Proc. of IEEE/ACM MASCOTS*, Volendam, The Netherlands, Oct. 2004.

[3] M. Grossglauser and D. Tse, "Mobility Increases the Capacity of Ad Hoc Wireless Networks," *IEEE/ACM Trans. on Networking*, vol. 10, no. 4, Aug. 2002.

[4] D. Aldous and J. A. Fill. Reversible Markov Chains and Random Walks on Graphs. http://www.stat.berkeley.edu/∼aldous/RWG /book.html