Robust Routing in Dynamic MANETs

Victoria Manfredi
Dept. of Computer Science
University of Massachusetts
Amherst, MA, USA
Email: vmanfred@cs.umass.edu

Robert Hancock
Roke Manor Research Ltd
Romsey, Hampshire, U.K.
Email:robert.hancock@roke.co.uk

Jim Kurose
Dept. of Computer Science
University of Massachusetts
Amherst, MA, USA
Email: kurose@cs.umass.edu

***Abstract*—The military environment presents special challenges for wireless mesh networking. In addition to optimal use of scarce radio resources, dependable network operation is critical: given frequent link and topology changes, rapid recovery of connectivity may be vital for mission success. This paper proposes a routing algorithm for wireless mesh networks with the primary goal of maximizing connectivity while limiting overhead. Rather than using one or more disjoint routes between a source and destination, a set of non-disjoint routes, or braid, is selected. To adapt to link changes, local rerouting is performed within the braid, thus avoiding network-wide recalculations. We analytically characterize the source-destination connectivity of the braid. Through simulation, we compare the reliability of braided routing and various other MANET routing protocols, including AODV, and quantify the relative amounts of control overhead incurred by braided routing and AODV.**

I. INTRODUCTION

Wireless networks, and MANETs in particular, are characterized by time-varying link characteristics and network topology. In such environments, the network must accommodate these changes, providing end-end packet delivery while at the same time incurring low control overhead. Yet this ideal is difficult to meet in practice: end-end delivery requires some form of end-end (potentially global) coordination, and frequent changes make adaptation to each and every change costly. Link and mobility characteristics may also be difficult to estimate *a priori*, making proactive or predictive routing approaches difficult to implement in practice.

In this paper, we investigate *robust* routing in MANETs. By "robust" we mean that although a particular routing configuration (in our case, a set of multipath routes) may not be optimal for a single *specific* configuration (e.g., specific network topology and link characteristics), it will perform well over a larger set of likely network configurations: i.e., it is robust to changes without requiring global recomputation. The issues of local versus global adaptation to link/topology changes, and the timescale(s) at which this adaptation occurs (and the concomitant overhead incurred) are central to the MANET routing problem. There is a growing recognition [2], [3], [4] that the scale and dynamic nature of MANETS, especially in military deployments, present severe challenges for classical MANET protocols, which are conceptually based on maintaining a consistent network-wide topological viewpoint. In contrast, the approach to MANET routing explored in this paper is based on the intuition that a global routing configuration should be determined at a coarse time scale (e.g., periodically, every T time units), with local adaptation to link or topology changes occurring at finer times scales within the current global configuration.

This paper specifically investigates an approach towards MANET routing, which we refer to as "braided routing," that is robust to changes in link characteristics and network topology. Informally, braided routing operates at two timescales. At the longer timescale, a subgraph of links and nodes is constructed that connects a source and destination. Unlike many existing "backup routing" approaches that pre-compute disjoint paths, e.g., [10], or partially disjoint paths, e.g., [6], a braid does not impose such requirements on the subgraph. Like approaches such as [6], braided routing performs local adaptation in response to link and topology changes. But unlike approaches that route packets over the entire network topology to achieve robustness (e.g., [23]), the subgraph over which packets are forwarded in a braid is purposefully limited to limit control overhead (e.g., for braid construction and state maintenance). The tradeoff between the control overhead incurred (which depends in turn on the size of the braid and the interval at which the braid is re-computed), and packet delivery performance will be of principal concern to us. Our results show that braided routing can indeed achieve a performance gain over a traditional MANET routing algorithm such as AODV and other approaches such as k-disjoint routing, without significantly increasing overhead.

We analyze braided routing from several different viewpoints in order to fully explore and understand its properties. We analytically characterize the reliability (the probability that the source and destination nodes have a contemporaneous path) of a class of braids, their optimality properties, and counter-examples to conjectured optimality properties in a well-structured (grid) network. We also compare the reliability of braided, disjoint-path, and full-network routing in Matlab simulations in both torus and random networks. Finally, we investigate the control overhead of braided routing and AODV from implementations of these two protocols in a GloMoSim simulation of a mobile scenario. In addition to quantifying the gains and overheads of braided routing, our Matlab and GloMoSim simulations also illustrate the impact and subtleties involved with using different underlying network models.

The remainder of this paper is structured as follows. In Section II, we discuss related work on backup routing. In

Section III we describe the reliability metric we use as a measure of robustness and in Section IV we present analytic and experimental results evaluating our braid structure in terms of reliability. Then in Section V we present a simple algorithm for constructing and maintaining a braid and in Section VI we present simulation results evaluating its performance. Finally, Section VII summarizes our results and outlines future work.

II. Related Work

A variety of work has considered the use of disjoint routes in ad hoc networks, including [12], [15], [18], [21]. In addition to the overhead cost of finding disjoint paths, if any link in a path breaks then the path itself breaks. Detection and recovery from failures is also expensive since it cannot be carried out locally. These considerations have thus motivated research on the use of non-disjoint paths.

The backup routing algorithm of [13] reinforces the path selected by AODV [19] by allowing nodes that overhear AODV control messages to become part of the routing subgraph, to be used only when links on the AODV path break. [22] proposes duct routing in mobile packet radio networks, where nodes neighbouring the primary route may be used. Specifically, when sending packets to the ith hop node along the primary path, one of either the ith hop node or one of its neighbours will hear the transmission first. The first node that hears the transmission will forward the packet to the $(i+1)$st hop node; the other nodes will overhear the forwarding transmission and refrain from transmitting. Considering an underwater network, [17] proposes a geo-routing mesh using only nodes within a given distance from the vector from the source or current forwarding node to the sink. We note that [22] (when all nodes neighbouring the primary path are used) and [13], [17] build routing subgraphs which structurally correspond to what we will describe in Section IV as a 1-hop braid. Braided multipaths are proposed in [6] to protect against node failure. A braided multipath corresponds to selecting a primary path and then adding an additional path for each node i on the primary path that does not use node i, possibly reusing parts of the primary path.

Specifically considering reliability, [16] argues for the reliability benefits of using non-disjoint paths in wireless mesh networks, showing gains over disjoint paths. [7] considers the problem of finding the most reliable subgraph for routing. Due to the #P-hardness of this problem, they propose a method to approximately compute reliability and a routing algorithm that leverages known contact probabilities between node pairs to select a routing subgraph.

III. What do we mean by robust?

Informally, we consider a routing subgraph to be *robust* if there is at least one path up between the source and destination with high probability. If a subgraph is robust, then even if a link or path between the source and destination breaks, an alternative link or path is available with (high) probability. In reliability theory [5], the probability that there is at least one path up between a source and destination is known as 2-terminal reliability, a metric we will use for evaluating the robustness of different routing subgraphs and providing intuition about what types of graphs are "highly" reliable.

Following Colbourn [5], the 2-terminal reliability of a graph $G = (V, E)$ with IID edges up with probability p is given by,

$$R(G,p) = \sum_{i=0}^{m} N_i p^i (1-p)^{m-i} \quad (1)$$

where $m = |E|$, N_i is the number of pathsets with i edges, and $p^i(1-p)^{m-i}$ is the probability that a pathset with i edges is up. A pathset is defined as a subset of edges for which there is a route between the two terminals.

Ideally, for a given source and destination, and specified number of edges or nodes, we would select the subgraph that has maximum 2-terminal reliability while using at most the specified number of links or nodes. Computing reliability exactly, however, is generally $\#P$-complete [5], as is solving the corresponding optimization problem [7]. For all-terminal reliability (the probability that a graph is connected), [9] gives a randomized fully polynomial time approximation scheme. For very reliable graphs, [9] shows that only small cuts are likely to fail and that there are only a polynomial number of such cuts, otherwise Monte Carlo simulation may be used. The approach in [9] could presumably be used to approximate 2-terminal reliability, although this does not efficiently solve the optimization problem, nor lend itself easily to theoretical comparisons of the reliability of different subgraphs.

Given the difficulty of exactly computing reliability, except for relatively simple networks, we also use Monte Carlo simulation to estimate reliability. In a discrete-time simulation of a time-varying network, we can check whether there is a path from the source to the destination at each time-step. The ratio of the number of time-steps when there was a path and the total number of time-steps simulated is then an estimate of the probability of there being at least one path from source to destination. We refer to computing the reliability in this way as "computing the reliability experimentally."

IV. Braided Graphs

In this section, we characterize the properties of braids in well-structured grid networks. Our goal is to determine how well a braided routing subgraph performs with respect to 2-terminal reliability given a fixed number of nodes/links in the subgraph. These results then provide valuable insight into more general network topologies, which are analytically intractable. We use the idealised network model shown in Figure 1: the source s and destination d lie in a bounded half-plane grid, with N nodes on the shortest path between them. All links are IID with reliability p.

A. The k-hop Braid

Our goal here is to explore the use of a k-hop braid built around the best (most reliable) path. A k-hop braid consists of the path itself, and all nodes and links within k hops of nodes on the best path. Figure 2 shows an example best path, 1-hop braid, and 2-hop braid between a source and destination.

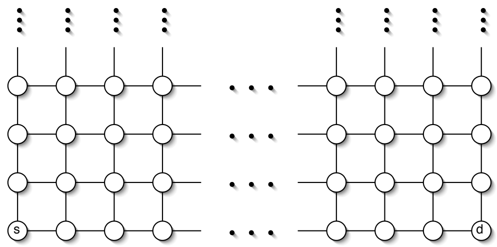


Fig. 1. Model used in Section IV, comprising source (s) and destination (d) on a line in a bounded half-plane grid.

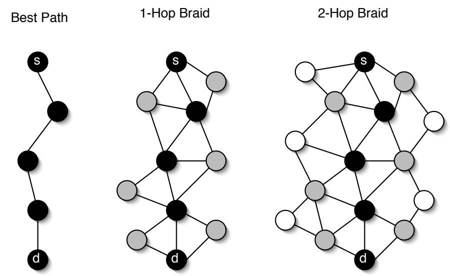


Fig. 2. Example best path, 1-hop braid, and 2-hop braid between a source (s) and destination (d).

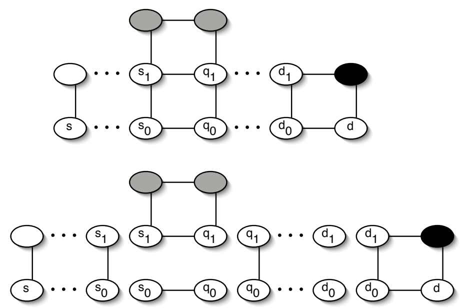


Fig. 3. Graphs used to decide whether to add two nodes on top of the $2 \times N$ strip or one node to the end. The bottom graph decomposes the top graph so that we need only compute the reliability for the subgraphs of interest.

	Two Nodes on Top	One Node at End
$P(q_0 q_1 \mid s_0 s_1) = P(q_0 \mid s_0) P(q_1 \mid s_1)$	$p(p + p^3 - p^4)$	$p \cdot p$
$P(q_0 \bar{q}_1 \mid s_0 s_1) = P(q_0 \mid s_0) P(\bar{q}_1 \mid s_1)$	$p(1 - p - p^3 + p^4)$	$p(1 - p)$
$P(\bar{q}_0 q_1 \mid s_0 s_1) = P(\bar{q}_0 \mid s_0) P(q_1 \mid s_1)$	$(1 - p)(p + p^3 - p^4)$	$(1 - p)p$
$P(q_0 q_1 \mid s_0 \bar{s}_1) = P(q_0 \mid s_0) P(q_1 \mid \bar{s}_1)$	0	0
$P(q_0 \bar{q}_1 \mid s_0 \bar{s}_1) = P(q_0 \mid s_0) P(\bar{q}_1 \mid \bar{s}_1)$	p	p
$P(\bar{q}_0 q_1 \mid s_0 \bar{s}_1) = P(\bar{q}_0 \mid s_0) P(q_1 \mid \bar{s}_1)$	0	0
$P(q_0 q_1 \mid \bar{s}_0 s_1) = P(q_0 \mid \bar{s}_0) P(q_1 \mid s_1)$	0	0
$P(q_0 \bar{q}_1 \mid \bar{s}_0 s_1) = P(q_0 \mid \bar{s}_0) P(\bar{q}_1 \mid s_1)$	0	0
$P(\bar{q}_0 q_1 \mid \bar{s}_0 s_1) = P(\bar{q}_0 \mid \bar{s}_0) P(q_1 \mid s_1)$	$p + p^3 - p^4$	p
$P(d \mid d_0 d_1)$	p	$\leq p + p^3 - p^4$
$P(d \mid d_0 \bar{d}_1)$	p	$p + p^3 - p^4$
$P(d \mid \bar{d}_0 d_1)$	p^2	$2p^2 - p^4$

TABLE I

RELIABILITY COMPUTATIONS FOR THE BOTTOM SUBGRAPHS IN FIGURE 3.

In the small p limit, the reliability polynomial (1) will be dominated by terms from shorter paths; this indicates that the most reliable path is an appropriate part of the braid, at least in this limit. Conversely, [5], [20] gives an alternative expression for (1) as a polynomial in $q = 1 - p$ as

$$R(G, p) = 1 - \sum_{C_i \in \mathcal{C}} P(E_i) \quad (2)$$

$$P(E_i) = \left[\prod_{e \in C_i} q \right] \left[1 - \sum_{C_j \in L(C_i)} \frac{P(E_j)}{\prod_{e \in C_i \cap C_j} q} \right]$$

where $\mathcal{C}$ is the set of minimal cuts separating source and destination, and $L(C_i)$ is the set of minimal cuts lying entirely between the source and C_i. In the small q limit, the unreliability $1 - R$ is dominated by the smallest cuts, So we observe that a good braid will have large minimal cut: i.e., that the braid should widen uniformly along the best path. We show this for $k = 1$ and arbitrary p in Lemma 1.

Lemma 1: *When incrementally adding nodes (one or two at a time), adding all nodes one hop away from the strip before adding any nodes that are two hops away maximizes reliability.*

Proof: Consider the top graph in Figure 3 and suppose we can add either one of the grey nodes or the black node. Adding only one of the grey nodes will not affect the reliability from s to d, as no links will be reinforced, while adding the black node will increase the probability of getting from nodes d_0 and d_1 to node d.

Now consider adding two nodes at a time. Again consider the topologies in Figure 3. We partition the top graph in Figure 3 into the sub-graphs shown in the bottom graph; note that each edge appears only once, although nodes may be repeated (which will not affect the reliability). Using sub-graph decompositions we decompose the reliability by conditioning on the intermediate nodes as follows. We first condition on intermediate nodes s_0 and s_1 to obtain,

$$P(d|s) = P(d|s_0 s_1) P(s_0 s_1|s) + P(d|s_0 \bar{s}_1) P(s_0 \bar{s}_1|s) + P(d|\bar{s}_0 s_1) P(\bar{s}_0 s_1|s) \quad (3)$$

where e.g., $P(d|s_0 \bar{s}_1)$ is the probability that node d can be reached given that node s_0 but not s_1 can be reached, and $P(s_0 \bar{s}_1|s)$ is the probability that node s_0 but not s_1 can be reached given that node s can be reached. We recursively condition on nodes $\{d_0, d_1\}$ and $\{q_0, q_1\}$ to obtain an equation for $P(d|s)$ as the sum of 27 terms (see [14] for details).

We use the resulting equation to compute both the reliability when adding both of the grey nodes in Figure 3 and when adding the black node. Ignoring those terms that correspond to subgraphs that are identical for both we need only compute the reliability for the $\{s_0, s_1\} \to \{q_0, q_1\}$ and $\{d_0, d_1\} \to d$ subgraphs. These calculations are shown in Table 1. Examining Table 1 shows that for each $P(\{q_0, q_1\}|\{s_0, s_1\}) P(d|\{d_0, d_1\})$ product, adding the black node to the end of the $2 \times N$ node strip gives the same or higher reliability as compared with adding the two grey nodes anywhere on top of the strip. $\diamondsuit$

Note that this result is actually more general than stated as it does not depend on the form of the subgraph between s and s_i, nor between q_i and d_i. It also follows from Lemma 1 that for short paths (with $n \leq 5$ nodes), that given n additional nodes, reliability is maximized by the $2 \times n$ node strip.

The results above suggest that k-hop braids have several desirable reliability properties, at least in this well-structured

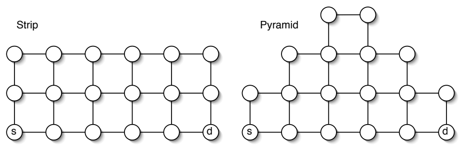


Fig. 4. Two topologies, both using 18 nodes.

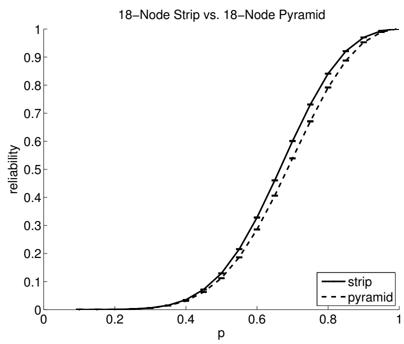


Fig. 5. Reliability comparison of "strip" and "pyramid" graphs from Figure 4. Reliability is averaged over 100 runs of 10,000 time-steps each. 95% bootstrap confidence intervals over the runs are shown.

environment, giving confidence in studying k-hop braids in other scenarios where the optimum subgraph cannot be determined. Referring to Figure 1, the $k \times N$ node strip from s to d is a $(k-1)$-hop braid. We close with two conjectures:

Conjecture 1: *Given N additional nodes (and their associated edges), the $2\times N$ node strip is the most reliable subgraph.*

Conjecture 2: *Given $2N$ additional nodes, the $3 \times N$ node strip is more reliable than the corresponding pyramid.* Comparing the $3\times N$ node strip for $N=6$ versus the 18-node pyramid (see Figure 4), we find experimentally that the strip has higher reliability than the pyramid, as shown in Figure 5.

B. Exact Results for $2 \times N$ Node Strip

For the $2\times N$ node strip, we determine an exact expression for reliability. Define $R_N \equiv$ P(s is connected to d), $S_N \equiv$ P(s is connected to d'), and $T_N \equiv$ P(s is connected to both d and d'), where d' is the node diagonally opposite to s. Then the recurrence relationships as the strip grows in length by 1 are,

$$\begin{aligned} R_{N+1} &= R_N p + S_N p^2 - T_N p^3 \\ S_{N+1} &= R_N p^2 + S_N p - T_N p^3 \\ T_{N+1} &= (R_N + S_N)p^2 + T_N(p^2 - 2p^3) \end{aligned} \tag{4}$$

This is a simple set of linear relations; the equation for $R_N - S_N$ is trivial, and the remaining equations for $R_N + S_N$ and T_N can be solved as a 2×2 matrix, see [14]. Then,

$$\begin{aligned} R_N &= \frac{1}{2}(C_0(p\lambda_0)^N + C_1(p\lambda_1)^N + (1-p)(\lambda_0\lambda_1)^N) \\ \lambda_{0,1} &= \frac{1}{2}\left(1 + 2p(1-p) \pm \sqrt{1 + [2p(1-p)]^2}\right) \\ C_0 &= \frac{(1+p)^2 - 2p^3 - \lambda_1(1+p)}{\lambda_0 - \lambda_1} \\ C_1 &= -\frac{(1+p)^2 - 2p^3 - \lambda_0(1+p)}{\lambda_0 - \lambda_1} \end{aligned} \tag{5}$$

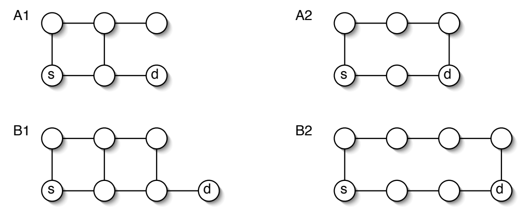


Fig. 6. Counterexamples when adding links rather than nodes.

Note that λ_0, λ_1 and $p(1-p) = \lambda_0\lambda_1$ are the eigenvalues of the recurrence relation, and the largest (λ_0) controls the large N behaviour. The evolution of R_N with N indicates that as s and d get one hop further apart, the reliability of the strip decreases by a factor of $p\lambda_0 > p$, compared to a factor of p for the single path or a pair of disjoint paths. For $p \to 1$,

$$p\lambda_0 \to 1 - (1-p)^2 + O[(1-p)^3]$$

which is a much slower degradation of robustness. We return to this theme in a more general setting in the next subsection.

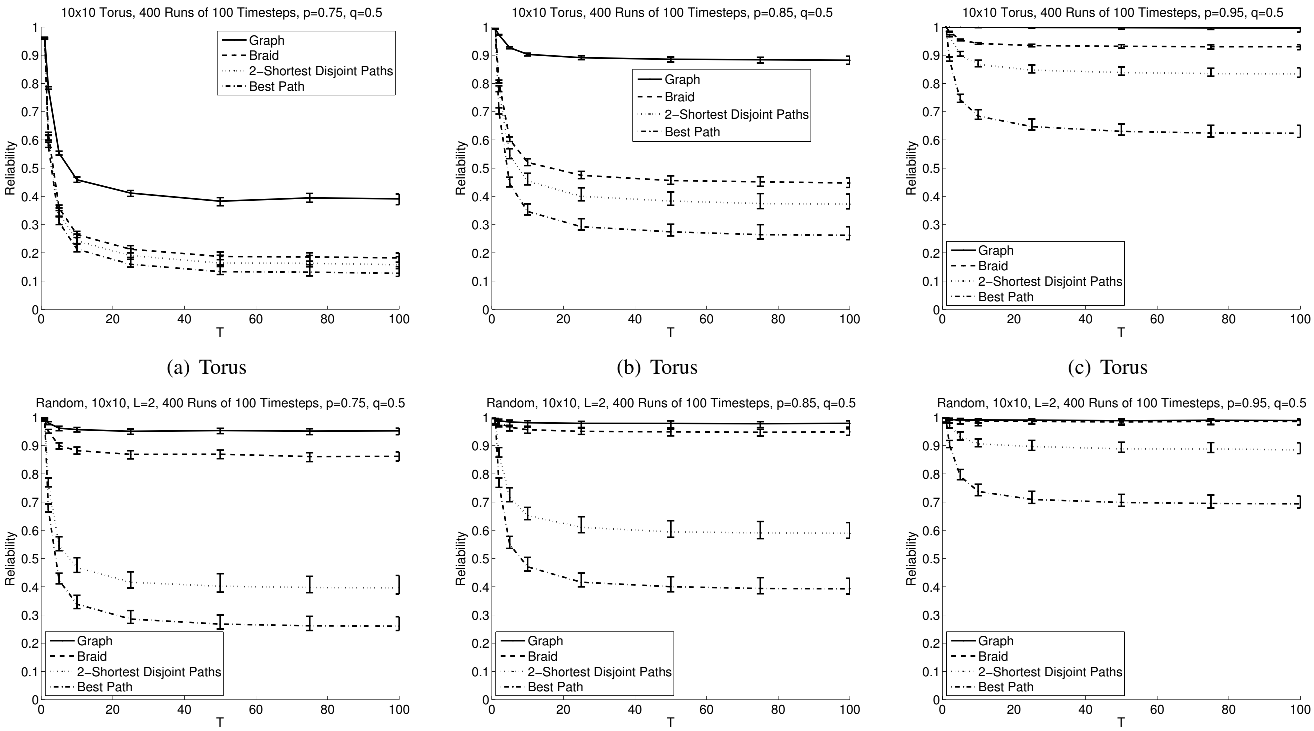
We also use (5) to establish a subsidiary result about the growth of the $2 \times N$ node strip: that it is optimal to add nodes contiguously. We consider adding nodes to give a total of $L+M$ links 1 hop away the shortest path, either in a single group of $L+M+1$ nodes or separate groups of $L+1$ and $M+1$. Formally, we show that (see [14]) $R_{L+M-1} \geq R_L R_M$ for arbitrary L and M, despite the use of fewer nodes/links.

C. Comparison with Disjoint Path Routing

A degenerate case of a 1-hop braid, where all internal links are missing, is a pair of disjoint paths which use neighbouring nodes. Does the optimal braid with a constraint on the number of links contain holes of this type? The answer depends on the measure of overhead and the value of p. Consider the examples in Figure 6 of a partial braid and a pair of disjoint paths. Graphs A_1 and A_2 both use six links total, however $R(A_1) = p^2 + p^4 - p^5$ while $R(A_2) = p^2 + p^4 - p^6$. Hence, graph A_2 is more reliable than graph A_1 for all values of p. Similarly, graphs B_1 and B_2 both use eight links total, however $R(B_1) = p^3 + 3p^5 - 2p^6 - 3p^7 + 2p^8$ while $R(B_2) = p^3 + p^5 - p^8$. Now, however, graph B_2 is more reliable than graph B_1 when $p > \sqrt{2/3}$, i.e., the braid is only more reliable for low values of p. More generally (see [14]), for even $N = 2n+2$ and $2N$ links, a pair of disjoint paths, containing $2n+1$ and $2n+3$ links respectively, has reliability $P_{disjoint} = p^{2n+1}(1+p^2-p^{2n+3})$. In comparison, a braided path of length $n+1$ links, combined with a path of single links of length n has reliability $P_{braided} = p^n R_{n+2}$. The reliabilities match at a critical value of p, which scales with N according to the following equation,

$$1 - P_{critical} \quad \to \quad \frac{2}{N} \log \frac{1+\sqrt{5}}{2} \tag{6}$$

i.e., the regime in which the braid is more reliable becomes larger for larger networks. Thus, one criterion for evaluating the applicability of braided routing is that the product of the network diameter and the mean link unavailability be greater



(a) Torus (b) Torus (c) Torus

(d) Random (e) Random (f) Random

Fig. 7. Reliability of different routing subgraphs. Reliability is averaged over 400 runs of 100 time-steps. 95% confidence intervals over the runs are shown. As not all sets of samples were normally distributed, bootstrap confidence intervals were computed using Matlab (hence the error bars are not symmetric).

than a threshold value of approximately 1. Note that this analysis assumes that "links used" is the appropriate overhead metric; an alternative metric is "nodes used," for which the appropriate comparison is between the disjoint paths and the full 1-hop braid, and the latter is always more reliable.

We extend this analysis to k-hop braids and k'-disjoint paths, studying how the reliability scales with increasing N by one hop, for large values of N. For disjoint paths, the reliability decreases by factor p for each additional hop regardless of the value of k' (which only affects a fixed overall coefficient). For the k-hop braid an exact expression is not available (except for $k = 1$ as given above); however, we can use the Provan-Ball equation (2) to find the leading terms in the reliability in the limit of small q. As discussed above, these leading terms correspond to the minimum cuts in the network; as the source-destination distance increases by 1, there is a single additional minimum cut of length $k+1$. Thus the corresponding reliability decrease is just $1-(1-p)^{k+1}$ which can be made arbitrarily small by increasing k. Even for moderate values of p, it is possible to grow to large networks without compromising robustness. These results confirm that, in terms of reliability, braids will be most effective for large diameter networks with relatively unreliable links.

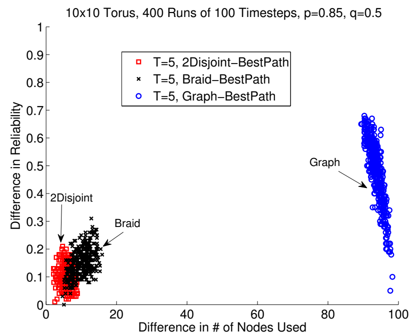
D. Simulation Results

In this section we compare the reliability of a 1-hop braid with that of the shortest path, the two shortest disjoint paths, and the entire graph, using a time-varying network simulated in Matlab. We first describe the model and then present results.

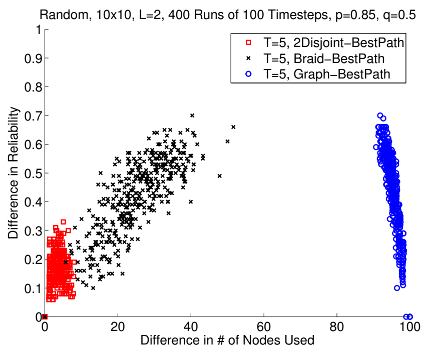
1) Network model: Consider a graph $G = \{V, E\}$ with nodes V and edges E. We examine (i) an $\sqrt{|V|} \times \sqrt{|V|}$ torus where $|E|$ comprises the set of all edges in the torus and (ii) a random model, where $|V|$ nodes are placed uniformly randomly and independently in the plane, and edges exist between those nodes within a communication radius L of each other. We assume links are IID; to model link changes, we use a two-state Markov model where links stay up with probability p and stay down with probability q at each time-step.

In our experiments, we use (i) a 10×10 torus and (ii) 100 nodes distributed randomly in an area of size 10×10 using a communication radius $L = 2$. We perform 400 simulation runs, each comprising 100 timesteps. In each run, a random source-destination pair is selected. For each time-step, we check whether each link is up. For the two-state Markov model we use $p = \{0.75, 0.85, 0.95\}$ and $q = 0.5$. We use the steady-state probability that a link is up to initially select which links are up or down. The routing sub-graph for each algorithm is recomputed every T timesteps, using only links that are up in the graph at the time of re-computation. All algorithms were evaluated on identical network topologies, and we estimate the reliability experimentally as discussed in Section III.

2) Results: Figure 7 shows that for all p, that as the update interval T is increased, the reliability of the selected routing subgraph decreases and eventually reaches steady-state. For the torus, Figure 7(a) shows that for $p = 0.75$, the reliability of the 1-hop braid, 2-shortest-disjoint paths, and shortest path



(a) Torus, T=5



(b) Random, T=5

Fig. 8. Overhead of 1-hop braid vs. that of the shortest path, the two shortest disjoint paths, and the entire graph. Reliability was estimated experimentally.

are all within a range of 0.1. Increasing p to 0.85 in Figure 7(b) shows a larger gap in reliability between the 1-hop braid and the 2-shortest disjoint paths, and also a larger gap between the braid and the full graph. Using $p = 0.95$ in Figure 7(c) shows an even larger gap in reliability between the 1-hop braid and the 2-shortest disjoint paths, but now a much smaller gap between the braid and the full graph. For the random model, Figures 7(d), (e), and (f) again show that for all p, the 1-hop braid has consistently higher reliability than the shortest path or 2-shortest disjoint paths, now as much as 0.4 greater than the 2-shortest disjoint paths when $p = 0.75$ or $p = 0.85$. This is in part a consequence of there not always being 2 disjoint paths in the graph (unlike in the torus). When $p = 0.95$, in Figure 7(f), the reliability achieved by the 1-hop braid is almost identical to that achieved by the full graph.

Figure 8 plots the reliability gain and number of additional nodes used over the shortest path by the 2-shortest disjoint paths, 1-hop braid, and full graph. Each point represents a simulation run (i.e., a selected source destination pair); for clarity we show only results for when $T = 5$. For the torus, Figure 8(a) indicates that the 1-hop braid provides an increase in reliability while using fewer than 20 extra nodes. For the random model, Figure 8(b) indicates that while the braid provides consistent and significant (up to about 0.4) gains in reliability, it also uses around 40 more nodes than the shortest path, but fewer than half the nodes used by the full graph.

In summary, the torus results indicate that the 1-hop braid can achieve reliability greater than that of the shortest path and the 2-shortest disjoint paths, and that the gains increase as p increases. We expect, however, that using a 2-hop braid would increase the reliability gain of the braid for small p. The results from the random model indicate that while using more nodes, the 1-hop braid can achieve reliability close to that of the full graph, and that the gain increases as p decreases.

TABLE II
ROBUST ROUTING ALGORITHM.

```
Let G be graph of entire network
Loop every T:
    Select "best" path P from graph G
    Build k-hop braid around P to obtain graph B
    Perform local forwarding on B
```

V. Robust Routing Algorithm

In this section we outline our robust routing algorithm; summarized in Table 3. The important features are as follows.

Recomputing routes periodically. We select a routing sub-graph that can be found efficiently, and that is expected to perform "well" over the time period T during which it is not updated. Based on our analysis in the previous section, we choose this subgraph to be a k-hop braid.

Local forwarding within braid. Given the braid sub-graph B, rather than forwarding packets over a path, we consider all of B: i.e., we make local forwarding decisions to select the next hop out of all possible next hops within B. While the sub-graph B changes every T timesteps, local forwarding decisions are computed by nodes every timestep. A simple approach to perform local forwarding (which we use to obtain simulation results in Section VI) is to have a node select its next hop based on which of its outgoing links have dropped packets; we describe this approach further in the next section.

VI. Evaluation

In this section we compare the performance of the braided routing algorithm using a 1-hop braid, with that of AODV [19]. We first describe our implementation in GloMoSim [11], and then present experimental results.

A. Algorithm Implementation

AODV is used to construct the best path for the braid algorithm (but any other single path routing algorithm could be used). The 1-hop braid around this best path is then constructed as follows. When a node receives data to forward along the AODV path, it sends a braid request for the associated destination (if one has not yet been sent). When a node receives a braid request for a destination, it groups the request with other requests for that destination. If it finds it can hear at least two nodes on the path, it sends a braid reply to all nodes it can hear (except the node closest to the destination).

To tear the braid down, a braid node sends error messages to nodes it can hear on the AODV path when either one of its links to the AODV path breaks (i.e., drops a packet) and T has elapsed, or when it receives a more recent braid request for the destination (indicating that the current AODV path has been replaced). A node deletes its next hop braid for a destination when either (i) its next hop or later link on its AODV best path has dropped a packet for that destination, or (ii) a node for that destination is updated in its AODV routing table. A node marks a link as "bad" whenever the node attempts to use a link and has a packet dropped. The AODV path and/or braid will be recomputed only when T has elapsed. Whenever routes are recomputed, links are marked as "good."

Nodes perform local forwarding within the braid as follows. Nodes on the AODV path select their AODV next hop with probability 1 if it is "good" or if there is no next hop braid node, and with probability 0.1 if it is "bad." If the AODV next hop was not selected, then the node iterates through its braid links. A braid link is selected with probability 1 if it is good or probability 0.1 if it is bad. If the node iterates through all of its braid links without selecting a next hop, then by default the AODV next hop is returned. If the node is a braid node, then it iterates through the nodes it can hear on the AODV path, selecting the AODV path node that is currently both closest to the destination and good. To ensure that bad links are also attempted, any AODV path node can be selected with probability 0.1. If the node iterates through all of its AODV path nodes without selecting a next hop, then by default the first AODV path node in its list is returned.

B. Simulation Setup

Our GloMoSim environment consists of 60 nodes in a 1.5km x 1.5km area, moving according to a random waypoint mobility model with the nodes initially placed randomly. For the mobility model, the pause time was zero seconds and nodes moved at a speed uniformly chosen between 4km/hr and 10km/hr. We used a constant bit rate flow between two nodes for which data was generated every 0.5sec and a total of 5 million packets were generated. The simulation was run for 60 simulated days. To address the problem of a long transient phase, the length of the flow was selected by examining the packet drop rate for progressively longer flows; when the change in % of packets dropped was sufficiently small ($< 0.05\%$), we assumed that steady-state had been reached. A better method would be to implement the "perfect simulation" method of Le Boudec and Vojnovic [1]; we leave this for future work. The MAC protocol used was 802.11 and the transmission radius was about 250 meters (from setting the radio transmit power to 7.9dBM).

C. Results

Figure 9 compares AODV and braid routing with respect to throughput, overhead, and links used. Figure 9(a) shows that the braid achieves a maximum of about 5% higher throughput than AODV for $T = 50$ and $T = 100$. Figure 9(b) shows that the braid uses about the same amount of AODV overhead when building its best path as AODV (as measured by the number of path requests and replies transmitted by AODV). While the braid also incurs overhead from braid requests and replies, this overhead is less than 1/4 of that used to construct the AODV path. Figure 9(b) also shows that the total number of error packets transmitted for braid routing (aggregating error packets for both AODV and the braid) is perhaps five times greater than AODV error packets, in part because the braid involves more nodes in routing. Figure 9(c) shows that the braid algorithm attempts to use more links than AODV (where "attempt" indicates that the routing algorithm attempted to transmit a packet over a link, but may not have been successful), in part because it may use a longer path. The braid, however, also has fewer links broken than does AODV.

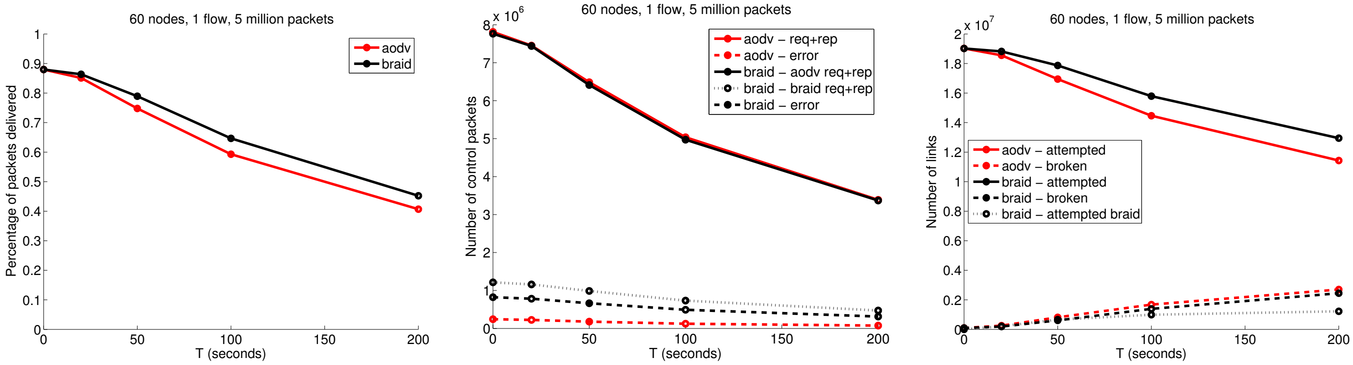
In summary, Figure 9 indicates that the 1-hop braid is able to increase throughput up to about 5% while using significantly less overhead than, for instance, would be needed to construct a second disjoint AODV path. The gains in throughput, however, are not as significant as the gains in reliability shown in the Matlab experiments in Section IVD. We conjecture that this discrepancy is a consequence of the different network models, particularly in how they differ with respect to the rate at which links appear/disappear, and the temporal and spatial correlations among links changes.

Consider first the rate at which links appear/disappear. Results from [8] indicate that the inter-meeting times for two nodes using the random waypoint model are "well-approximated by an exponential distribution, at least for small to moderate transmission radii (with respect to the size of the area)." Using Lemma 1 in [8], we compute that for the transmission radius and random waypoint model considered here, the expected inter-meeting time for two nodes is given by $1/\lambda$ with $\lambda = 2.65/\text{hr}$. Hence, on average, two nodes will meet once every 22.7 minutes. This indicates that in our GloMoSim experiments, that when a link breaks (due to mobility) that it likely stays down for an interval significantly longer than the update interval T. Conversely, the probability of transitioning from down to up during T was 0.5 in the models used in the Matlab experiments in Section IVD. Long inter-meeting times limit the throughput gains achieved by the braid since when braid links fail it is unlikely that they will re-appear before the remaining time in the update interval T has elapsed.

Next consider correlations among links. In the Matlab experiments we assumed links failed independently. Conversely, we would expect that outgoing links of a given node would tend to have correlated failures when links break due to mobility. We would also expect that since all link failures are varying functions of how much time t of the interval T has elapsed, that link failures among different nodes would also be dependent due to the shared dependence on t. Correlated link failures limit the throughput gains achieved by the braid since if a link on the AODV path fails, it is also more likely that one of the links routing around the failed link will also fail soon (if it has not already).

VII. Conclusions

This paper described a braided routing algorithm to improve the robustness of dynamic MANETs. We proposed a general method for braid construction and presented analytic optimality results in a restricted class of networks. We developed scaling laws for robustness as a function of path length and braid



(a) (b) (c)

Fig. 9. Comparison of 1-hop braid with AODV using GloMoSim.

width. We validated the theoretical results through simulation, finding additional effects due to link failure correlations.

For future work we are interested in triggering route updates as a result of changes in end-to-end network performance, rather than using a fixed update interval. Similarly, rather than using a fixed braid width, we are interested in techniques to locally widen the braid to meet a robustness target. Addressing the issue of correlated links, we would like to explicitly consider correlations as well as more realistic radio link models. Finally, we would like to explore rate control mechanisms such as backpressure routing [23] for local forwarding to achieve a solution which is robust in throughput as well as connectivity.

Acknowledgments

The authors would like to thank Majid Ghaderi, Matt Grossglauser, Andy Lam, John Spicer, Patrick Thiran, Don Towsley, and Andy Twigg for their input. This research was supported in part under the International Technology Alliance sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-06-3-0001, and by the the National Science Foundation under award number CNS-0519998 and via an International Research in Engineering Education supplement to Engineering Research Centers Program award number EEC-0313747. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the US National Science Foundation, the US Government, the UK Ministry of Defense, or the UK Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] J.-Y. L. Boudec and M. Vojnovic. The random trip model: Stability, stationary regime, and perfect simulation. *IEEE/ACM Transactions on Networking*, pages 1153–1166, 2006.

[2] J. Burbank, P. Chimento, B. Haberman, and W. Kasch. Key challenges of military tactical networking and the elusive promise of MANET technology. *Communications Magazine, IEEE*, 44(11):39–45, 2006.

[3] K.-W. Chin. The behaviour of MANET routing protocols in realistic environments. In *IEEE 11th Asia-Pacific Conference on Communications*, 2005.

[4] K.-W. Chin, J. Judge, A. Williams, and R. Kermode. Implementation experience with MANET routing protocols. *Computer Communication Review*, 32(5):49–59, 2002.

[5] C. J. Colbourn. *The Combinatorics of Network Reliabiilty*. Oxford University Press, New York, 1987.

[6] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *Mobile Computing and Communications Review*, 4(5), 2001.

[7] J. Ghosh, H. Ngo, S. Yoon, and C. Qiao. On a routing problem within probabilistic graphs and its application to intermittently connected networks. In *Infocom*, 2007.

[8] R. Groenevelt, P. Nain, and G. Koole. The message delay in mobile ad hoc networks. Technical Report RR-5372, INRIA Sophia Antipolis, Nov. 2004.

[9] D. Karger. A randomized fully polynomial time approximation scheme for the all-terminal network reliability problem. *SIAM Review*, 43:3, 2001.

[10] M. Kodialam and T. V. Lakshman. Dynamic routing of restorable bandwidth-guaranteeed tunnels using aggregated network resource usage information. *Transactions on Networking*, 11:3, 2003.

[11] U. C. S. D. P. C. Laboratory and W. A. M. Laboratory. GloMoSim: A scalable simulation environment for wireless and wired network systems. *http://pcl.cs.ucla.edu/projects/glomosim/*.

[12] S. Lee and M. Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. In *IEEE ICC*, 2002.

[13] S.-J. Lee and M. Gerla. AODV-BR: Backup routing in ad hoc networks. In *IEEE Wireless Communications and Networking Conference*, 2000.

[14] V. Manfredi, R. Hancock, and J. Kurose. Robust routing in dynamic MANETs. Technical Report 08-25, U of Massachusetts Amherst, Department of Computer Science, 2008.

[15] M. Marina and S. Das. On-demand multipath distance vector routing in ad hoc networks. In *IEEE ICNP*, pages 14–23, 2001.

[16] M. Mosko and J. J. Garcia-Luna-Aceves. Multipath routing in wireless mesh networks. In *IEEE Workshop on Wireless Mesh Networks*, 2005.

[17] N. Nicolaou, A. See, P. Xie, J.-H. Cui, and D. Maggiorini. Improving the robustness of location-based routing for underwater sensor networks. In *MTS/IEEE OCEANS conference*, 2007.

[18] P. Papadimitratos, Z. Haas, and E. G. Sirer. Path set selection in mobile ad hoc networks. In *MOBIHOC*, 2002.

[19] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *IEEE Workshop on Mobile Computing Systems and Applications*, 1999.

[20] J. Provan and M. O. Ball. Computing network reliability in time polynomial in the number of cuts. *Operations Research*, 32:3, 1984.

[21] A. Reeve. Resilient real-time communications across meshed networks under adverse conditions. In *1st SEAS DTC Technical Conference*, 2005.

[22] N. Shacham, E. Craighill, and A. Poggio. Speech transport in packet-radio networks with mobile nodes. *IEEE Journal on Selected Areas in Communications*, SAC-1:6, 1983.

[23] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37:12, 1992.