Analysis of Multiple Trees on Path Discovery for Beacon-Based Routing Protocols

Baruch Awerbuch*, David Holmer and Herbert Rubens
Johns Hopkins University
Department of Computer Science
Baltimore, MD, USA 20723–6099
Email: baruch@cs.jhu.edu

Robert G. Cole
Johns Hopkins University
Applied Physics Laboratory
and Department of Computer Science
Laurel, MD, USA 20723–6099
Email: robert.cole@jhuapl.edu

***Abstract*—Routing in Mobile Ad-hoc Networks (MANETS) has proved to be an extremely challenging research problem due to the high frequency of link changes in wireless mobile environments. Most known routing protocols struggle to maintain complex routing structures, e.g., N spanning trees, or use expensive (from the perspective of communications cost) methods for path discovery. These methods incur extreme communications costs in most MANET deployments. In this paper, we investigate the performance of a relatively unstudied class of routing protocols which we refer to as Beacon-Based Routing Protocols. This class of protocols maintain minimum routing structures and use these structures to boot strap path discovery. In a well defined sense, this class of routing protocols achieves optimal performance with respect to minimizing the communications costs associated with on-demand routing. We specifically investigate the impact of multiple tree implementations for path discovery and analyze their impact on the effectiveness of optimal path discovery.**

I. INTRODUCTION

It has long been realized that routing in highly dynamic mobile ad-hoc networks (MANETs) represents an extremely challenging problem for protocol designers. Fundamentally, the problems with traditional approaches to routing in these environments is that protocols attempt to maintain either overly complex routing structures pro-actively or hard to maintain path structures reactively in an environment where the network topology turns over on the order of each t_{link} seconds, where t_{link} is the mean lifetime of a communications link between neighboring nodes. In [1] and [2] a alternative, hybrid approach to routing in MANETs was developed called the *Pulse Protocol*. The Pulse Protocol pro-actively builds a single spanning tree structure in the network periodically, and then uses this structure to bootstrap path discovery on-demand for active network flows. An implementation of the Pulse Protocol has been developed and fielded at JHU Campus [9].

Link State protocols [6], and their numerous MANET variants, such as Hazy Link State [8], Optimized Link State [4], and others, attempt to maintain $O(N)$ spanning trees in the network, where N is the number of nodes. Because the maintenance cost (in terms of communications) of each spanning tree structure is also $O(N)$, the overall cost to proactively maintain the Link State structures in the network scales like $O(N^2/t_{link})$ where t_{link} is the mean lifetime of a communications link. Additionally, Link State protocols are required to maintain link structures to each nodal neighbor. This further reduces the scalability and reactivity of these routing protocols. In highly dynamics MANET environments, this cost of Link State maintenance becomes problematic.

On-Demand protocols, such as Dynamic Source Routing [5] and Ad-hoc On-demand Distance Vector [7] routing, attempt to avoid these problems by building and maintaining only paths for active flows. Hence they maintain Nf paths in the network, where f is the mean number of active flows per node. Unfortunately, they have no means to path discovery other than broadcasting route request packets throughout the network. Since the cost to broadcast is $O(N)$, the scalability of these protocols is $O(fN^2/t_{link})$.

Instead, the Pulse Protocol maintains a single spanning tree in the network which costs $O(N/t_{pulse})$ to maintain, where t_{pulse} is the mean time interval between the root node broadcasting the pulse, or beacon message. It then avoids per flow broadcasts, instead sending one or two unicast messages on the spanning tree for flow establishment. Hence, this protocol appears to have extremely attractive scaling properties.

In general, the Pulse Protocol can be classified as a member of a class of routing protocols which rely upon the maintenance of B spanning trees, where B is a parameter of the specific routing protocol. The protocol then relies on these B tree structures in the network for the purpose of efficient, reactive path discovery and maintenance for data flows. The Pulse Protocol represents the case where $B = 1$. We refer to this class of routing protocols as *Beacon-Based Routing (BBR) Protocols*.

In this paper we investigate four different routing methods in the context of BBR protocols, i.e., *Root*, *Tree*, *Short-Cut* and *Gratuitous*. The B-spanning trees are periodically refreshed by broadcasting a beacon message. Once complete, the nodes have forwarding pointers to their B-parents. To discover a forwarding path, a *Source* node unicasts B-messages to the B-root nodes requesting paths to the *Destination*. During the next beacon broadcasts, the root nodes page the Destination, which responds with unicast messages up the trees towards the Source. Once the Source receives these unicast messages,

---

*Baruch Awerbuch is supported by the following grants NSF-CNS-0617883, NSF CCF 0515080, NSF ANIR-0240551, NSF-CCR-0311795.

three types of forwarding paths exist in the network. A Root path, where traffic flows through a root node. A Tree path, where traffic flows through the common parent on the tree. And a Short-Cut path, where nodes overhear one-hop Short-Cut paths when operating in promiscuous mode with omni-directional antennas. Finally, in this paper, we also investigate the performance of a *Gratuitous* routing method whereby intermediate nodes can issue one-hop broadcast messages announcing shorter paths through themselves.

In a previous paper [3], we investigated the effectiveness, in terms of *Routing Stretch*, of bootstrapping paths off of a single spanning tree, e.g., the Pulse Protocol. In this paper, we specifically investigate the impact on Routing Stretch of supporting multiple trees for path discovery, e.g., generalized Beacon-Based Routing Protocols.

II. ROUTING OVERHEAD AND STRETCH

Following [8]. the routing overhead is comprised of three components:

- *Proactive Overhead (POH)* - this component incorporates all proactive routing overhead, e.g., control messages, etc., which are transmitted throughout the network in order to maintain the necessary routing structures within the network.
- *Reactive Overhead (ROH)* - this component incorporates all control messages generated within the network based upon the establishment and maintenance of data flows within the network.
- *Stretch Overhead (SOH)* - this component is due to the additional data packet transmissions that are required due to the inefficiencies introduced by a particular routing protocol relative to the best path routing [1].

In this paper, the Routing Stretch is the critical routing overhead component. Since we are focusing on routing protocols which rely on the maintenance of one or a few spanning trees from which to bootstrap data paths, the Routing Stretch is the component of overhead often of concern to engineers when discussing tree-based routing. Although our routing protocols are not tree-based, they do build paths off of the existence of one or a few spanning tree. Hence, the purpose of this paper is to demonstrate that this technique of bootstrapping off of one or a few spanning trees does in fact result in extremely efficient paths having a low value for their Stretch Overhead.

The Stretch Overhead is in fact highly traffic dependent [3], although often it is discussed only in the context of a uniform traffic matrix. Let $T_{i,j}$ be the traffic volume from node i to node j in the network. Let $\tilde{T}_{i,j}$ be the relative traffic volume from node i to node j in the network. It is defined such that $\sum_i \sum_{j\neq i} \tilde{T}_{i,j} = 1$. Hence, we have that $T_{i,j} = T \times \tilde{T}_{i,j}$, where T is the total network traffic volume. If the network traffic is totally uniform (sometimes referred to as totally peer-to-peer), then we can write the relative traffic matrix as $\tilde{T}^{(u)}_{i,j} = 1/[N(N-1)]$.

We define $P_{i,j}$ as the realized path length from node i to node j, and $P^{(m)}_{i,j}$ as the minimum path length from node i to node j. Then we can write the Stretch as

$$SOH = T \times \sum_i \sum_{j\neq i} \tilde{T}_{i,j} \tilde{S}_{i,j} \tag{1}$$

where $\tilde{S}_{i,j} = P_{i,j}/P^{(m)}_{i,j}$. For the uniform traffic matrix, this reduces to

$$SOH^{(u)} = T \frac{1}{N(N-1)} \sum_i \sum_{j\neq i} S_{i,j} = T\tilde{S}^{(u)} \tag{2}$$

In this paper, we analyze the relative Stretch under assumptions of a uniform traffic matrix. The relative Stretch is always greater than or equal to unity; approaching unity in the case of best path routing.

III. MULTI-TREE RESULTS

In this section, we present our initial investigations of the Stretch Performance of multiple tree-based BBR protocol systems. We describe our test topology models. Then we discuss some protocol issues. Then we present or simulation modeling results.

A. Topology Models

As a first investigation of multiple spanning tree routing, we analyze topologies where the nodes are placed at random within a square grid. Each node is assumed to have an identical radio whose transmission range is defined within a disk of radius equal to the range. For simplicity, we define a base topology consisting of 50 nodes randomly placed within a 1000 m by 1000 m grid. The node's radio range is 250 m representing an omni-directional transmission within a free-space transmission model. We study the effectiveness of our routing algorithms with respect to this topology model as a function of number of nodes, obtained by fixing the nodal density and increasing the grid size, and density, by fixing the grid size and increasing the number of nodes placed within the grid.

When representing this model graphically, we will use a "+" symbol to indicate the placement of the node and we will encircle the node with a circle of radius equal to one half the radio range. Hence, when two nodes' circles overlap then they are assumed to have radio contact and are connected, else they are assumed to be not directly connected. We show this model in Figure 1.

The objective of this study is to determine the effectiveness of using multiple spanning tree structures to bootstrap efficient path discovery. Looking at typical network topologies as shown in Figure 1 we can see that the graphs are not uniform in the sense of homogeneous node placement and connectivity. In fact the resulting graphs are highly heterogeneous in their local connectivity and fluctuations of the local nodal density. An apparent challenge in bootstrapping from a single (or multiple) tree structure(s) is the possibility of finding two nodes in close physical proximity to one another while finding themselves at large distances on the tree structure(s). Further, due to the

---

[1] Although in this paper we focus on the min-hop path routing stretch, in general the Stretch Overhead ought to be defined in terms of the 'best path' routing metric

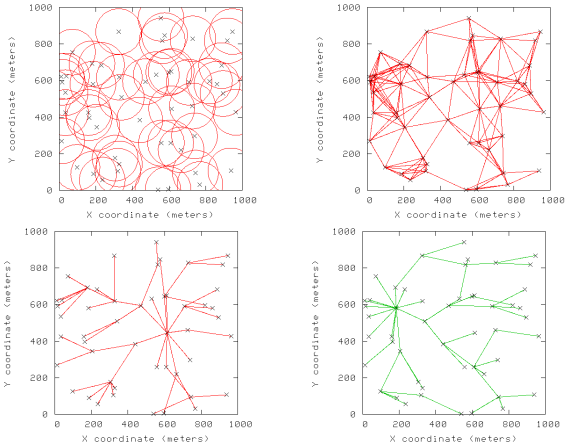

Fig. 1. Example nodes, connectivity and 2-tree in random topologies. The upper left plot shows typical nodal topologies, the upper right plot shows the resulting network graph, and the two lower plots show example spanning trees with randomly chosen root nodes.

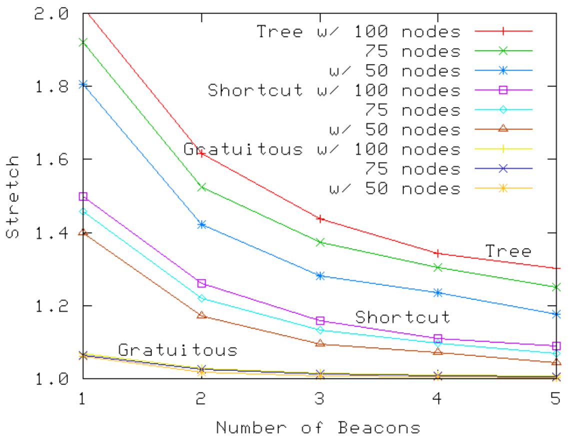

Fig. 2. The effects of multiple Beacons ranging from 1 to 5 randomly placed root nodes on the stretch, where the top figure gives results for Tree routing, the middle plot gives results for Short-Cut Routing, and the bottom gives results for Gratuitous Routing.

local heterogeneity in the graphs, we observe that the specific choice of a root node can make significant differences in the effectiveness in building paths between nodal pairs.

B. Routing Techniques

As is the case for the single tree routing system [3], there is a dynamic leader election protocol. However, there are multiple leaders elected so that each node maintains B forwarding table entries, one for each Beacon. When a source node wishes to establish a flow to a destination node, it has various choices available to it. It could choose a single Beacon's tree to bootstrap a route based upon previous history or other local information. However, it could also unicast a route_req packet to all (or a subset) of the B Beacons. The Beacons, following the previous results for single tree BBR [3], would each page the destination node in the next broadcast packet. The destination would then unicast a route_req to all (or a subset) of the page requests from the multiple Beacons. The source, receiving multiple connection_complete messages with their associated distance metric, can choose a best path to the destination. However, the metrics carried on the connection_complete messages reflect the accumulated path metric associated with the Short-Cut path, and not the Gratuitous path. Once the source chooses a path to forward packets over, the associated Gratuitous path will develop following a handful of packet transmissions. It is possible that the resulting Gratuitous path is not the best overall Gratuitous, but we suspect it is a good approximation to it.

C. Stretch Computations

We wrote a C++ simulation to investigate the impact of multiple, simultaneous Beacons within the network as defined above. The source node then estimates the best tree to bootstrap a route to a given destination based upon the best Short-Cut route returned from the route_request queries over the multiple trees. The Gratuitous path results from the tree with the best Short-Cut.

In this section we randomly and independently placed the multiple Beacons within the network topology. We varied the number of multiple Beacons from a low of one to a high of five. Figure 2 shows the impact of the addition of more Beacons on the stretch results for the various routing mechanisms is similar. The results for two Beacons show roughly a 50% reduction in the deviation of the Stretch values from the ideal case (i.e., a Stretch of unity), for all three routing mechanisms. The results for Gratuitous routing show an even greater reduction with two Beacons. The relative benefit of adding additional Beacons diminishes as more Beacons are added. At five Beacons the benefits of further additions is small.

Although the Gratuitous routing mechanism shows the greatest relative benefit in adding multiple Beacons (the best case on the plot has a Gratuitous stretch of 1.0031 ± 0.20 for the smallest density of nodes and with five Beacons), it is interesting to note that the stretch results for the Short-Cut mechanism drops to around 1.1 for 3-5 multiple Beacons. Even the Tree-based routing mechanism shows good Stretch performance with multiple Beacons; approaching a stretch of 1.2 for five beacons.

D. Optimal Beacon Placement

As for the case of a single Beacon, see [3], it is interesting to consider the question of optimal placement of multiple Beacons within the network. The previous results showed that the Stretch performance of the system improved as the number of Beacons increased from one to five. So we now fix the number of Beacons in the network and measure the Stretch performance as we move their location.

Various placement strategies are possible, including strategies based upon topology considerations, upon tree location

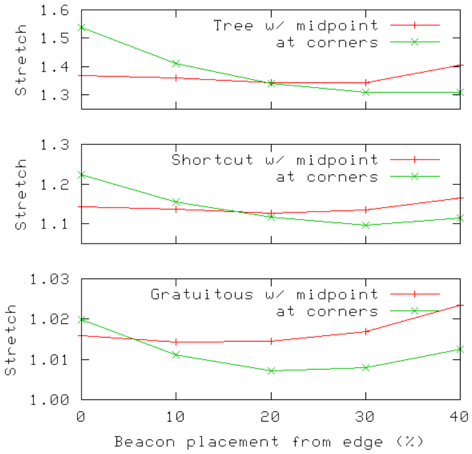

Fig. 3. Two placement strategies for two Beacons, where the top figure gives results for Tree routing, the middle plot gives results for Short-Cut Routing, and the bottom gives results for Gratuitous Routing.

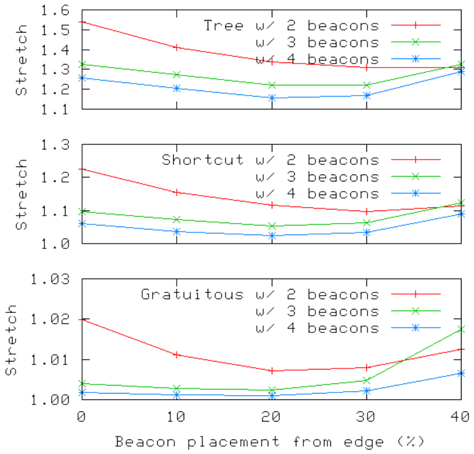

Fig. 4. Optimal, topology-based placement of multiple Beacons, where the top figure gives results for Tree routing, the middle plot gives results for Short-Cut Routing, and the bottom gives results for Gratuitous Routing.

considerations, upon nodal degree considerations, etc. Other placement considerations can take into account the nature of the traffic flows, e.g., sources and sinks, in the network, as an indicator of Beacon node placement. In this paper, we discuss topology-based Beacon placement. Placement based upon traffic considerations will be investigated in the future.

As a first exploration, consider the results shown in Figure 3. Here we considered two different placement strategies for the placement of two Beacons in the random topology. The cases labeled “w/ midpoint” have one Beacon placed at the geometric center of the topology grid and a second Beacon placed along a line from one edge to the center of the grid. We varied the location of this second Beacon and computed the Stretch overhead for each case. The other alternative, labeled “at corners”, had the two Beacons placed symmetrically opposite one another along a line running diagonally across the grid. As always, the grid is 1000m by 1000m, there are fifty nodes randomly placed within the grid (other than the forced placement of the Beacons), and the nodes are assumed connected if they line within 250m of one another. From the optimal placement results for a single Beacon [3], we know that centrally placing a Beacon is beneficial. The results show that the placement of a second Beacon, when not too close to the central Beacon, is beneficial, although the benefit is not too large. The shape of the curves for the placement of the two Beacons symmetrically opposite one another on the diagonal is more interesting. Here, in all cases but most prominently for the Gratuitous Stretch results, the effect of different Beacon placements shows a minimum at a placement of the Beacons roughly halfway between the edges and the mid-point of the grid. For these cases, the minimum Stretch for the Gratuitous, Short-Cut and Tree routing mechanisms were 1.0071 ± 0.20, 1.0964 ± 0.21 and 1.3092 ± 0.26 respectively.

Based upon the results for the symmetric placement of two Beacons, we next considered a similar strategy for the placement of two, three and four Beacons within the network. The placement for the case of two Beacons is as described above. For the case of three Beacons, we symmetrically placed the Beacons on the corners of a triangle where two of the Beacons lie along diagonals between the two upper corners of the grid and the grid mid-point. The third Beacon is placed along a line from the mid-point running straight down to the mid-point of the lower grid edge. For the case of four Beacons we placed them symmetrically along the four diagonals from each corner to the grid mid-point. The results are shown in Figure 4, where the Tree routing results are shown at the top, the Short-Cut routing results are shown in the middle and the Gratuitous routing results are shown at the bottom of the figure. In all cases, the results show that the best Stretch performance is obtained when the Beacon nodes are placed somewhat mid-point along their line segments. The best case results show Stretch values within 15% for Tree-based routing, within 3% for Short-Cut routing and within 1% for Gratuitous-based routing relative to the perfect case, shortest path routing. Specifically, the best case Stretch results with four Beacons are 1.1580 ± 0.23 for Tree-based routing, 1.0238 ± 0.20 for Short-Cut routing, and 1.0010 ± 0.20 for Gratuitous-based routing.

These results are rather remarkable in that they are, in some sense, worst case results when boot-strapping off of spanning trees due to our assumption of a uniform traffic matrix. That is, the computation of the Stretch in these results assume each node is equally likely to generate a traffic flow randomly distributed to any destination node. Thus, the uniformly distributed traffic flows, assumed for these computations, bare no relationship to the instances of the Beacon based spanning trees used to bootstrap the routes.

IV. CONCLUSIONS AND FUTURE WORK

This this paper we continue our expanded analysis of a new and exciting class of routing protocols for MANETs.

This class of routing protocols are referred to as Beacon-Based Routing protocols. In this paper we present a first analysis of the impact of supporting multiple (versus a single) spanning tree structures for the purpose of boot-strapping paths through the highly, dynamic MANET environment. We focused on Routing Stretch as the performance metric of interest. Our first topology models discussed are those of a random overlapping disk model. We developed simulation models to assess the Stretch performance for Root, Tree, Short-Cut and Gratuitous path routing for this simple topology model. Our results showed that multiple trees do in fact improve the Stretch performance of BBR protocols. Further, we showed that optimal placement strategies of the multiple root nodes can be effective in further performance improvements.

Our future work will fall into several areas of investigation. We plan to expand the type of topology models investigated, beyond that of a homogeneous overlapping disk model. We plan to investigate different types of multiple tree models, distinct from the multiple spanning tree models described here. We plan to investigate traffic aware placement algorithms of the location of multiple trees. Finally, we are developing a full, event driven simulation model of the extensions to the BBR protocols within the NS2 simulation platform.

REFERENCES

[1] Awerbuch, B., Holmer, D. and H. Rubens, *The Pulse Protocol*, IEEE Infocomm 2002, August, 2002.

[2] Awerbuch, B., Holmer, D. and H. Rubens, *The Pulse Protocol: Mobile Ad Hoc Network Performance Evaluation* Infocomm'05, San Francisco, CA, USA, October 2004.

[3] Awerbuch, B., Cole, R.G., Holmer, D. and H. Rubens, *Beacon-Based Routing for Tactical Mobile Ad Hoc Wireless Networks* submitted to IEEE MILCOM 2007, Orlando, FL, USA, October 2007.

[4] Clausen, T. and P. Jacquet, *Optimized Link State Routing Protocol (OLSR)* IETF Request for Comment 3626, October 2003.

[5] Johnson, D.B., Maltz, D.A. and J. Broch, *DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks* in “Routing in Mobile Ad Hoc Networks”, C. Perkins, Ed., John Wiley and Sons Press, New York, September 2001.

[6] Moy, D., *OSPF - Anatomy of an Internet Routing Protocol* Addison-Wesley Press, New York, January 1998.

[7] Perkins, C.E. and E.M. Royer, *Ad-hoc On-Demand Distance Vector Routing* Infocomm'99, San Francisco, CA, USA, September 1999.

[8] Santivanez, C.A., Ramanathan, R. and I. Stavrakakis, *Making Link-State Routing Scale for Ad Hoc Networks* MobiHOC'01, Long Beach, CA, USA, September 2001.

[9] http://wireless.cs.jhu.edu