

Beacon-Based Routing for Tactical Networks

Baruch Awerbuch*, David Holmer, Herbert Rubens

Department of Computer Science
Johns Hopkins University
Baltimore, Maryland, USA

Robert G. Cole

Applied Physics Laboratory
and Department of Computer Science
Johns Hopkins University
Laurel, MD, USA

Abstract—The U.S. Department of Defense’s (DoD) warfighter is reliant upon the development of a reliable, resilient communications capability under harsh, battlefield environments. Due to high mobilities and the nature of the various terrains, the dynamics of the communications links is extremely erratic and rapidly changing. This results in extreme strain on the performance of routing protocols attempting to find and maintain viable communications paths.

In this paper we discuss and analyze a new class of routing protocols which we refer to as Beacon-Based Routing protocols. Beacon-Based Routing protocols proactively build a small number of, typically one or two, spanning trees in the network and use these trees to discover paths on demand. The existence of one (or more) spanning tree(s) ensures full network connectivity and hence can be used to find network paths without the need for network-wide broadcast of discovery messages, as in other on-demand routing protocols. This class of routing protocols represents the generalization of the Pulse Protocol [1], originally developed in 2002 for applications to Internet access networks. The performance of the Pulse Protocol was analyzed in [1] and [2] in various applications including general Mobile Ad-Hoc Networks (MANETs) and in sensor networks.

As we discuss in this paper, the Beacon-Based Routing protocol class has optimal behavior with respect to the communications overhead required to run the protocol. As such, its scaling behavior is superior to other existing routing protocol classes when assessed in the context of a MANET. Further, the performance of the protocol is discussed in the specific context of traffic flows within battlefield scenarios. We show how the protocol can be design to be traffic-aware and how this further improves the scalability of the routing overhead for tactical MANET scenarios.

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 routing 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 which was named 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 [14].

Link State protocols [7], and their numerous MANET variants, such as Hazy Link State [11], Optimized Link State [8], and others, attempt to maintain $O(N)$ spanning trees in the network, where N is the number of network nodes. Because the maintenance cost of each spanning tree structure is also $O(N/t_{link})$ where t_{link} is the average link lifetime in the network, the overall cost to proactively maintain the Link State structures in the network scales like $O(N^2/t_{link})$. 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 [6] and Ad-hoc On-demand Distance Vector [9] 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 sourced 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 $t_{pulse} \leq t_{link}$ is the period over which the single spanning tree is refreshed. It further avoids per flow broadcasts, instead sending zero, one or two unicast messages for flow establishment. Hence, this protocol appears to have very attractive scaling properties when compared to all other classes of MANET routing protocols. Thus, we are interested to initiate a further investigation of the performance and potential extensions to this protocol.

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 $B \ll N$ 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*. We use the term Beacon, based upon the general definition of a beacon as

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

beacon *n.* **1.** A signal fire, esp. one used to warn of an enemy’s approach. **2.** A lighthouse or other signaling or guiding device on a coast. **3.** A radio transmitter that emits a characteristic signal as a warning or guide. **4.** Something that warns or guides.

We feel that this class of routing protocols represent an attractive alternative to the majority of the routing protocols developed and analyzed in the context of MANETs. Further, this class of routing protocols has received little attention in the Internet community and, as such, we believe they merit further attention. There are numerous design and algorithmic questions that should be addressed in order to better understand the potential of this class of routing protocols. We initiate these investigations in this paper and in a companion paper [4].

Specifically, in this paper we investigate several important topological considerations with respect to BBR protocols. Here we study the following aspects of these systems:

- What is the typical stretch, S , in a 1-Tree routing system and how does it scale with the number of nodes, N ?
- What is the impact of tree Short-Cuts and Gratuitous shortcuts on the typical stretch for a 1-Tree routing system and how does it scale with the number of nodes, N ?
- Are there “optimal” strategies for the placement of the Beacon in a 1-Spanning Tree BBR system?

Issues and results with respect to utilization of multiple spanning tree implementations are discussed in a companion paper [4].

The rest of this paper is organized as follows: In the next section we present a brief overview of the Pulse Protocol as defined in [1] and [2]. This is important because the Pulse Protocol forms the foundation for the analysis of extensions represented in this paper. We then define routing overhead in Section III and discuss the importance of minimizing Stretch. We next discuss the study topologies and present some examples through illustrative plots. In Section V we investigate the performance characteristics of a 1-Tree routing system in the context of random topologies and strategies for placement of the Beacon. In Section VII we present our conclusions and thoughts for future work.

II. PULSE PROTOCOL

In this section we present a brief overview of the Pulse Protocol [1] [2] along with a set of extensions for routing in Mobile Ad Hoc Networks (MANETs). The Pulse Protocol was originally developed in the context of access networks and of networks of sensors, where power constraints were of paramount concern. However, it has evolved to handle general routing within a MANET environment and has been deployed in a test environment [14]. What we describe here is a flavor of the original Pulse Protocol with the concerns for power consumption and time synchronization removed and with additional mechanisms, e.g., Short-Cut and Gratuitous routing, added for improved MANET efficiency. Here we describe a base version of the Pulse Protocol for the purpose

of analyzing characteristics of the base protocol and new extensions.

The Pulse Protocol is typically represented as a hybrid, i.e., proactive and reactive, routing protocol. It is an on-demand, reactive routing protocol in the sense that it builds and maintains forwarding paths for active data flows within the network. However, unlike traditional on-demand routing protocols which resort to the worst-case discovery method, i.e., a network flooding algorithm for each active flow, the Pulse Protocol pro-actively maintains a minimal network structure for the purpose of route discovery without individual source node flooding.

The Pulse

The Pulse Protocol dynamically elects a Beacon source. The Beacon periodically floods a pulse message throughout the network. The pulse message contains the root node label, the distance to the root node, a sequence number and a list of nodes that the Beacon is currently paging. As a node receives the pulse message, it updates the distance metric and if the metric is lower than previously received it also updates its forwarding table entry point towards its new parent node (up the spanning tree) and rebroadcasts the pulse message. Else, it discards the message. Once the message has propagated throughout the network, the network has built a minimal spanning tree connecting all nodes up to the Beacon node.

If the cluster of nodes split, then the nodes within the “disconnected cluster” (disconnected in the sense that they are no longer connected to the original Beacon node), can nominate themselves as a new Beacon and broadcast a pulse packet. As the multiple pulse packets travel through the new cluster, each node decides to accept the new Beacon based upon a common algorithm which distinguishes between the Beacon nodes based upon the node label or other information. If it later receives a preferred pulse packet, it can update its forwarding table entry pointing to the new parent and rebroadcast the new pulse packet. Quickly, the nodes in the new cluster will have all converged onto a single new Beacon, and further cluster operation proceeds as above.

Flow Establishments

If a node receives a request from an application to communicate with a distant, destination node, the source node unicasts a *route_req* up to the Beacon¹ unless it already has a forwarding table entry to the destination. The *route_req* packet, addressed to the root node, contains the destination label and the metric from the source. Each node along the path to the root, a) installs a reverse path (towards the source) in its forwarding table, b) updates the metric from the source, and c) transmits the updated *route_req* packet up the tree towards the root node. Other nodes, one hop from the source to root path and which operate in promiscuous mode, can install reverse path entries in their forwarding tables based upon overhearing the

¹Note: this is completely different from traditional on-demand protocols which flood the network with the *route_req* message. Hence the reason for maintaining a minimal path discovery structure within the network.

route_req packet traveling up the tree towards the root node.

The Beacon, when it receives the route_req packet, first checks if it has a forwarding table entry towards the destination node. If it does, it responds to the source. If it does not, it waits till the next pulse broadcast and it places the destination label within the pulse packet (essentially paging for the destination nodes). When the destination nodes receives the pulse packet with its label in the paging field, it unicasts a route_req up to the root node towards the source. This installs the reverse routes in the nodal forwarding tables up the tree and results in an end-to-end Short-Cut Path (see discussion below on routing mechanisms and in particular on Short-Cut Path routing) between the source and the destination.

Flow Management

The source begins sending data packets to the destination. The data packets contain the source and destination labels in their headers, along with a sequence number and the distance metric from the source. Each node along the path updates the distance metric and forwards the data packets along the path. If a path becomes broken, the path may be reestablished by the source and destination nodes issuing route_req packets up to the root node.

Path Optimization

The beauty of the BBR class of routing protocols is that they use a minimal structure, a spanning tree, to bootstrap routes, but do not rely upon the spanning tree to provide data forwarding paths. Instead, various path optimization techniques are described and implemented. These techniques include:

- *Root* – restricting all paths to flow through the root of the tree. This is known to be highly inefficient due to a high resultant stretch. Further, it is simple enough to improve on path discovery by implementing Tree, Short-Cut and Gratuitous-based routing techniques.
- *Tree* – restricting all paths to flow on the tree, however if the nodal pairs have a common parent on the tree which is not the root, then the route_req packets sent up to the root would naturally install forwarding paths which would take, at a minimum, the tree-based paths.
- *Short-Cut* – restricting all paths to flow, in some sense, “close” to the tree. By having nodes operate in promiscuous mode, nodes which are one hop from the tree-based paths will also install forwarding entries to the source and destination nodes associated with the flow. This will naturally result in cases where the actual data flow will shortcut off the tree-based routing path and can result in significant reductions in the cost of forwarding paths.
- *Gratuitous* – allowing nodes, who have promiscuously learned of shorter paths through themselves for the network flows, to transmit one-hop Gratuitous route modification messages to their neighbors. Nodes which receive these Gratuitous advertisements can modify their forwarding tables if the Gratuitous message is advertising

shorter cost routes to the destination nodes².

The Pulse Protocol [14] implements Tree, Short-Cut and Gratuitous-based path discovery.

Metric Measurements

The nodes within the Pulse Protocol rely upon data packet transmissions to sense/measure the status of their various links. For each active link, each node maintains a link quality metric which is updated based upon each received (either transmitted to the node or by promiscuously overhearing) packet. The nodes use the link metrics to update the distance metrics contained in the route_req and data packets flowing through the node. Specifically, the Pulse Protocol relies upon the Medium Time Metric [3] as the cost metrics for the active links in the network.

Multiple tree Extensions

In a companion paper, we discuss and analyze the multiple tree extensions for BBR protocol systems [4].

III. ROUTING OVERHEAD AND STRETCH

In this section we discuss the types of routing overhead and focus upon “Stretch” as a critical metric of study within this paper. We follow the suggested routing overhead metric found in [11]. Thus, the routing overhead is comprised of three components:

- *Proactive Overhead* (P_{OH}) - 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. This overhead is independent of the network traffic and is generated even if no traffic load is placed on the network. Hence the name *Proactive Overhead*.
- *Reactive Overhead* (R_{OH}) - this component incorporates all control messages generated within the network based upon the establishment and maintenance of data flows and traffic load within the network.
- *Stretch Overhead* (S_{OH}) - 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 minimum hop based routing³.

So we write the total routing overhead as

$$OH = P_{OH} + R_{OH} + S_{OH} \quad (1)$$

where OH represents the additional packets transmitted due to the particular routing protocol over that of a network which

²For a static topology and a uniform traffic matrix, i.e., all nodes are equally likely to establish flows to any other node, the Gratuitous routing technique will eventually converge to Shortest Path Routing as it represents an implementation of the Bellman-Ford algorithms for finding shortest paths. However, due to the dynamics in a MANET and/or the non-uniform nature of the traffic matrix, in general, the resulting paths based upon the Gratuitous advertisement technique will be somewhat longer than the shortest cost paths.

³Minimum-hop routing is not the preferred routing metrics for MANETs. The Stretch is in fact to be defined in more general terms of the best path routing metric. However, in this paper, we focus solely on minimum-hop routing metrics

always performs minimum cost routing without having to transmit control packets.

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, 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 $\tilde{T}_{i,j}^{(u)} = 1/[N(N-1)]$. If the network traffic is totally focused into a single node, say node r , then we can write the relative traffic matrix as $\tilde{T}_{i,j}^{(u)} = \delta_{j,r}/(N-1)$ where $\delta_{j,r}$ is the Kronecker Delta function which is unity if $j = r$ and zero otherwise. Finally, we define a hybrid relative traffic matrix, which is a linear combination of the uniform and the focused matrices, as

$$\tilde{T}_{i,j}^{(h)} = \frac{h\delta_{j,r}}{(N-1)} + \frac{(1-h)}{[N(N-1)]} \quad (2)$$

where h represents the proportion of the traffic which is focused into a single node r .

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

$$S_{OH} = T \times \sum_i \sum_{j \neq i} \tilde{T}_{i,j} \tilde{S}_{i,j} \quad (3)$$

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

$$S_{OH}^{(u)} = T \frac{1}{N(N-1)} \sum_i \sum_{j \neq i} S_{i,j} = T \tilde{S}^{(u)} \quad (4)$$

For the hybrid traffic matrix, we define an associated stretch as

$$\begin{aligned} S_{OH}^{(h)} &= T \left\{ \frac{h}{(N-1)} \sum_i \sum_{j \neq i} \delta_{j,r} S_{i,j} \right. \\ &\quad \left. + \frac{1-h}{[N(N-1)]} \sum_i \sum_{j \neq i} S_{i,j} \right\} \\ &= T \{ h + (1-h) \tilde{S}^{(u)} \} \end{aligned} \quad (5)$$

We rewrite this as

$$S_{OH}^{(h)} = T \{ h + (1-h) \tilde{S}^{(u)} \} \quad (6)$$

This last form for the Hybrid Stretch assumes that the focused portion of the traffic is routed over a shortest path as would be the case in the event that we place the Beacon, or tree root node, at the node r which is the focal point for the focused traffic flows. When we perform our Stretch analysis below, we will be referring to the relative Stretch which does not include the total traffic factor of T .

In our studies below, we analyze the relative Stretch under assumptions of a uniform traffic matrix first. We then will take those results, and estimate the Stretch under assumptions of a Hybrid Traffic Matrix using Eq.(6). The relative Stretch is always greater than or equal to unity; approaching unity in the case of perfect path routing. The absolute Stretch (incorporating the total traffic volume factor T) can be quite significant due to the factor of T . Hence the reason why the Stretch is such an interesting metric for assessing routing methods, particularly in MANETs where channel resources tend to be rather scarce.

IV. SCENARIOS

In this section we briefly discuss the topology scenarios studied within this paper. As a first investigation, 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 various instantiations in Figure 1. Here the plot on the upper left represents a random lay down of nodes on the grid. The plot on the upper right shows the resulting network connectivity map based upon radio range considerations. The bottom two plots show example, randomly chosen Beacon nodes and their associated spanning trees.

The objective of this study is to determine the effectiveness of using a spanning tree structure(s) 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

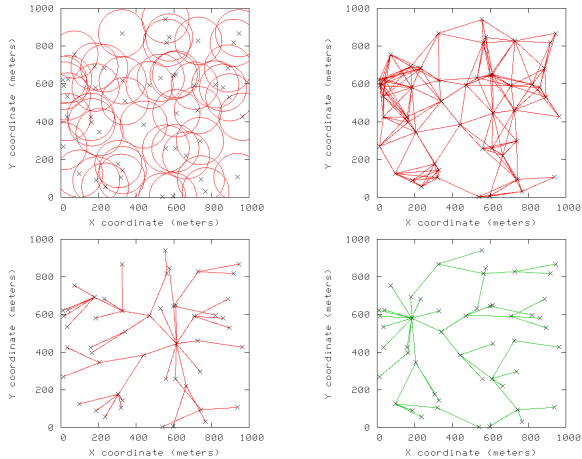


Fig. 1. Example nodes, connectivity and 1-tree in random topology.

large distances on the bootstrap tree structure(s). Further, due to the local heterogeneity in the graphs, we suspect that the specific choice of a root node can make significant differences in the effectiveness in building paths between nodal pairs. We will investigate these questions and observations below.

V. 1-TREE ROUTING SYSTEMS

Our first investigations relate to the effectiveness of relying upon a single tree structure within the network to bootstrap forwarding paths between nodal pairs. Previous simulation studies have compared the effectiveness of the Pulse Protocol in terms of packet delivery ratios compared with other well know MANET-based routing protocols [3]. That study showed that the Pulse Protocol performed well. Here we wish to perform a detailed studied of the resulting stretch obtained through various bootstrapping techniques possible within the context of a 1-Tree network structure discussed in the next section.

A. Stretch Computations

A simulation model was developed to assess the effectiveness of the above mentioned routing techniques within the context of a single spanning tree BBR system, i.e., the Pulse Protocol. In a companion paper [4], we investigate the impact of multiple spanning tree implementations within the context of BBR systems. The simulation was developed in C++. The simulation randomly places N nodes on a square grid. A uniform distribution was used for the nodal placement within the grid. Each node is assumed to have connectivity to another node if it lies within “radio range” of the node. The root node can be randomly placed within the grid as well, or it can be specifically placed at a given grid location. We use this capability to analyze optimal placement strategies of the root node. Once the nodal placement is complete, the simulation computes the connectivity map, i.e., the network graph, then it builds the spanning tree from the root node. Once the spanning tree is built, the simulation computes the Root Stretch, which assumes all paths are routed through the root node. The

stretch computation is that identified as the relative Stretch under a uniform traffic matrix assumption. The simulation then computes the Tree Stretch, which assumes all path remain on the spanning tree, but do not have to transit the root in the event that the source and destination nodes share a common ancestor on the tree below the root. The simulation then computes the Short-Cut Stretch, assuming that one-hop shortcuts will be taken to develop a shorter routing path. Then the simulation computes the Gratuitous Stretch, assuming a single traffic flow exists between the source and destination nodes in question and finds the shortest Gratuitous Path between the source and destination based upon this conservative assumption (see the discussion on the relationship between Gratuitous Routing and Bellman Ford forwarding below).

The base simulation case we study consists of 50 nodes placed in a 1000 meter by 1000 meter (m) grid and each node has a radio range of 250m to develop the network graph. Each data point shown on the graphs below represents the average over 100 independent simulation runs. Based upon the standard deviation of the 100 runs we compute estimates of the 95th percentile confidence intervals which are typically shown on the graphs as well.

Our first set of simulation studies are shown in Figures 2 and 3. Here, we picked three different nodal densities, i.e., 50 nodes per $10^6 m^2$, 75 nodes per $10^6 m^2$ and 100 nodes per $10^6 m^2$. We then varied the number of nodes for each case (by scaling the grid size) from a low of around 20 nodes to a high of 500 nodes. Figure 2 presents the results for the Path Lengths for the various path optimization techniques. Clearly, as we move from Root-based, to Tree, Short-Cut and Gratuitous-based techniques, the mean path lengths decrease dramatically. In Figure 3, for each density, we present two graphs; the bottom of the pair presenting the results for the mean (shortest) path length in the network along with a fitting of the well known $N^{1/2}$ scaling law and the upper graph of the pair presenting the Stretch results. The figure shows the computed stretch for Tree-based, Short-Cut-based and Gratuitous-based routing obtained by bootstrapping off a single randomly chosen spanning tree for the three nodal densities. The lowest nodal density is shown at the top of the figure, the mid-density in the middle of the figure and the highest nodal density at the bottom of the figure. The results show encouraging results with respect to the Gratuitous routing stretch. It appears that all stretch results show a weak dependence on the size of the network. The values for the Gratuitous stretch for the case of 200 node networks are 1.2344 ± 0.24 for the density of $5.0 \times 10^{-5} nodes/m^2$ case, 1.1701 ± 0.23 for the density of $7.5 \times 10^{-5} nodes/m^2$ case, and 1.1413 ± 0.22 for the density of $1.0 \times 10^{-4} nodes/m^2$ case. The comparable results for the Short-Cut routing cases for networks of 200 nodes were roughly 1.6 for the relative Stretch. Again, these results are for the case where the root of the spanning tree, from which routing is bootstrapped, is randomly placed within the grid.

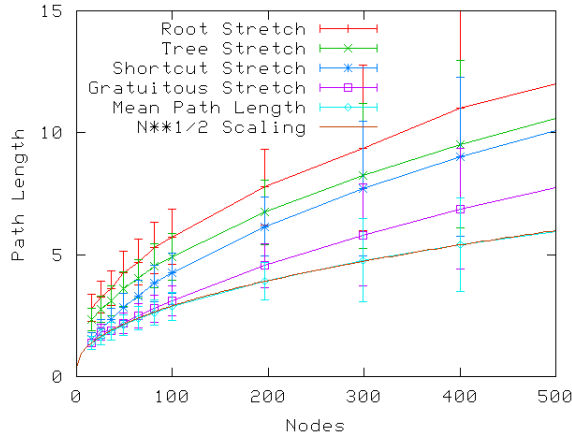


Fig. 2. path length results for the bootstrapping techniques applied to the 1-Tree structure, for a density of 1×10^{-4} .

B. Optimal Beacon Placement

In this section, we address the question of optimal placement of the root of the single spanning tree. Various placement strategies are possible for optimal root-node placement. In this section we consider two strategies; one is topological based and the other is traffic flow based.

1) *Topology Based Placement:* We first present the results for the optimal topology based placement of the root-node. In the next section we address the issue of traffic flow-based strategies. Our topology-based placement studies consist of looking at the stretch performance results as a function of placement of the root-node at various points along a diagonal running from one corner of the square grid to the mid-point of the grid. We placed the root-node at six points along this diagonal at points 0, 20, 40, 60, 80 and 100% from the corner of the grid along the diagonal. Hence, the results for the 0% placement always had the root node at the corner of the grid. The result for the 100% placement of the root node always had the root placed at the center of the grid. The simulation runs made for this section were for a 1000 by 1000 m^2 grid with 49, 74 and 99 nodes randomly placed within the grid and the root-node strategically placed within the grid.

Figure 4 shows the results of the simulation runs. Note that the scale for the Stretch axis differs for each plot. Again, each point on the graphs is the result of averaging over 100 independent runs. There were occasions, especially when the root node was placed in the very corner of the grid, that the root node was totally disconnected from all other nodes. These cases were rare and were excluded from the averaging process. The top graph presents the results for the Tree-based routing, the middle graph presents the results for the Short-Cut-based routing, and the lower graph presents the results for the Gratuitous-based routing. The results show that strategic placement of the root node close to the center of the grid significantly improves the stretch results for all three routing methods, i.e., Tree, Short-Cut and Gratuitous-based. In fact

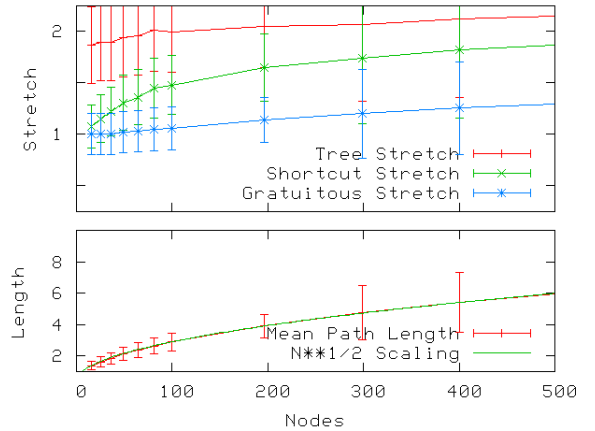
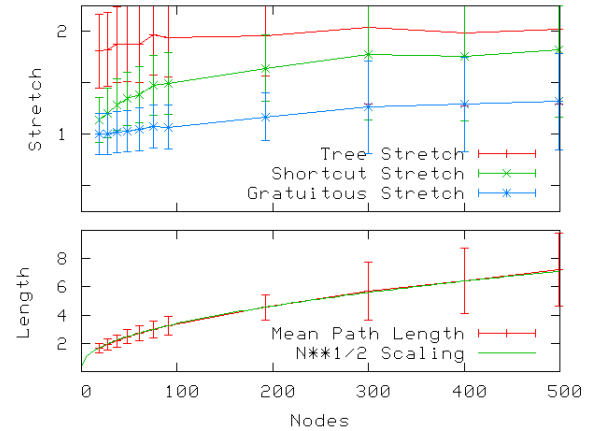
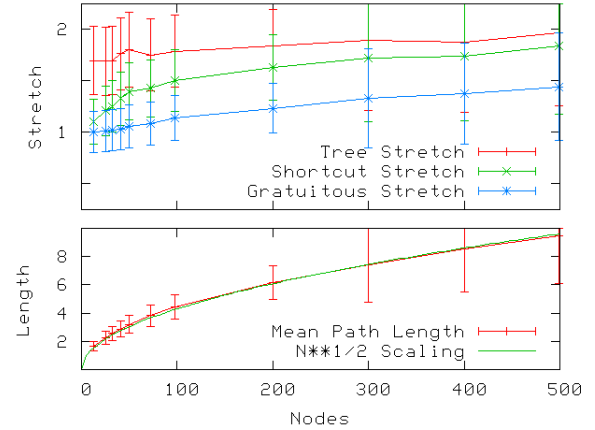


Fig. 3. Stretch for the bootstrapping techniques applied to the 1-Tree structure, where the top figure is for a density of 5×10^{-5} , the middle plot is for a density of 7.5×10^{-5} , and the bottom is for a density of 1×10^{-4} .

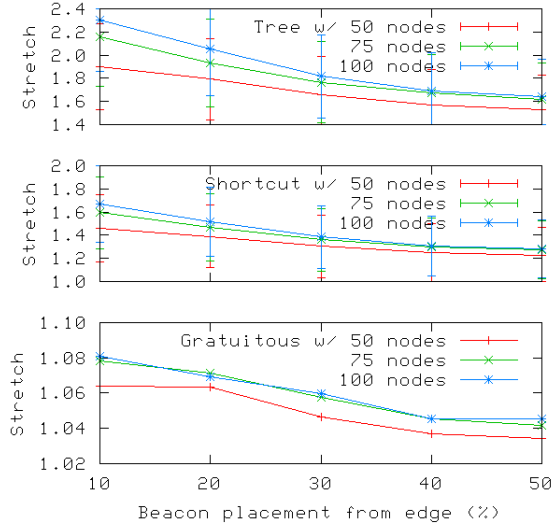


Fig. 4. The effects of root node placement for a 1-Tree structure, where the top figure gives results for Tree routing, the middle plot gives results for Short-Cut Routing, and the bottom gives results for Gratuitous Routing.

the best results for these studies yielded a relative Stretch of 1.0342 ± 0.20 for the fifty node case using Gratuitous-based routing, 1.2264 ± 0.24 for the fifty node case using Short-Cut-based routing, and 1.5300 ± 0.30 for the fifty node case using Tree-based routing.

2) *Traffic Based Placement*: We now consider the effects of a non-uniform traffic matrix and Beacon node placement on the relative Stretch. In Section III we discussed tree types of traffic matrices, i.e., Uniform, Focused, and Hybrid. The Uniform Traffic Matrix, sometimes referred to as peer-to-peer, has a uniform distribution of traffic between all nodal pairs. The Focused Traffic Matrix has a single node which acts as a traffic sink for all sourced traffic. The Hybrid is defined here as a linear combination of the Uniform and the Focused.

The Focused Traffic Matrix is very interesting in the context of commercial wireless access networks and in the context of military tactical wireless networks. In both these environments, a significant amount of the data traffic flows up towards a common location, e.g., the wireless access point in commercial networks or the center of command in military networks. Hence, the bulk of the traffic tends to flow up a tree towards the focus node. In these scenarios, in order to reduce Stretch overhead, it makes perfect sense to locate the Beacon at the node which acts as the traffic sink in the Focused Traffic Matrix. Therefore, placement strategies which place the Beacon node at the highest traffic volume node(s) in the network can achieve scale efficiencies in the context of Beacon-Based routing. Further, the fact that BBR can optimize itself to the natural traffic flows in the network demonstrates an advantage of Beacon-Based Routing over other routing schemes such as link state routing which expends needless resources in maintaining N spanning trees in the network when only one or a few are ever used.

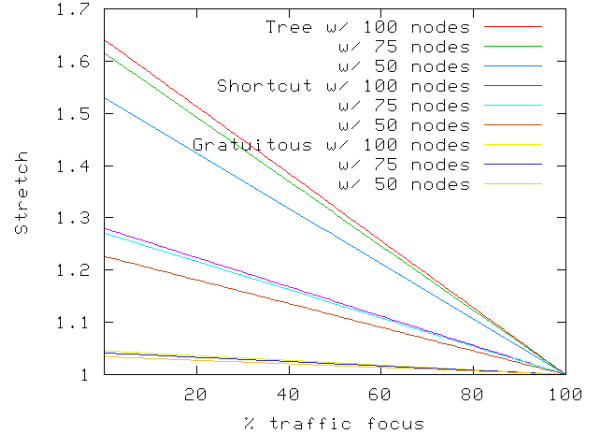


Fig. 5. The effects of a more focused traffic matrix on the relative Stretch results for the case of a single Beacon and Tree, Short-Cut and Gratuitous-based routing.

Eq.(6) gives the relationship between our previous relative Stretch results for the case of a uniform, peer-to-peer traffic matrix, and a more focused traffic matrix described by our Hybrid Traffic Matrix. Clearly, the BBR scheme can exploit a linear improvement in Stretch performance with respect to the relative volume of traffic that is focused versus uniformly distributed. Figure 5 takes the computed relative stretch values from the case of placing the Beacon node in the topological center of the grid and with Eq.(6) demonstrates the improvement in stretch in the event that a portion of the traffic flows are focused to the beacon node. Obviously, the larger the proportion of the traffic which is focused to the Beacon node, the greater volume of traffic which is routing over the shortest path and hence the closer the stretch becomes to unity.

In the limit that all traffic flows into the Beacon node location, Beacon-Based routing as implemented in the Pulse Protocol becomes the optimal routing protocol architecture. It achieves a scalability which is linear in N , versus N^2 as observed in all link state and on-demand routing protocols. Further, its on-demand overhead scales as number of flows times $N^{1/2}$ versus number of flows times N as observed in other on-demand MANET routing protocols, e.g., AODV and DSR.

Finally, we are currently in the process of analyzing the U.S. Army TRADOC's tactical battlefield scenarios including their Information Exchange Requirements (IERS). These scenarios are developed for the purpose of tactical system performance characterization. We hope to develop improved models of traffic flows and characteristics and improved mobility models based upon this analysis. These models will be used to assess the performance of our BBR protocols in the context of these model tactical scenarios.

C. Gratuitous Routing and Bellman-Ford

Finally, it is worth mentioning that, in some sense, our computations of the Gratuitous routing path lengths are conser-

vative. Our computations assume a single source-destination flow in the network when computing the Gratuitous path. In reality, there will be many simultaneous flows in the network, each resulting in Gratuitous update messages. As the number of simultaneous flows increase the collective effect of the Gratuitous updates will be that the Gratuitous-based routing which bootstraps off the spanning tree will converge to Bellman-Ford shortest path routing. This is a strong statement with respect to a routing mechanism which bootstraps off a single spanning tree.

VI. RELATED WORK

There exists an extensive literature on the design and development of routing protocols for MANETs. Primarily these break out into “Proactive” and “On-Demand” protocols, although there exist some “Hybrid” approaches.

The Proactive routing protocols tend to be based upon traditional link state routing protocols for wired networks, e.g., the Open Shortest Path First (OSPF) protocol [7], and variants which attempt to minimize the proactive routing overhead. Example, link state derivatives include the Hazy Link State routing protocols [11] and the Optimized Link State Routing (OLSR) protocol [8]. While these optimized MANET routing protocols improve the constant multiplier with respect to the scaling behavior as a function of the number of nodes, they are still link state based and fundamentally scale like $O(N^2/t_{link})$.

On-Demand routing protocols only build routing paths based upon traffic requirements. Examples include the Dynamic Source Routing (DSR) protocol [6] and the Ad-hoc On-demand Distance Vector (AODV) protocol [9]. As previously discussed, these protocols flood the network with route discovery packets in order to find paths between source and destination nodes. Hence their scaling is fundamentally $O(fN^2/t_{link})$.

There exist some Hybrid approaches with proactively build up some limited routing structures and use these for packet forwarding. One example of this type of routing is found in [5]. The Pulse protocol represents another Hybrid approach [1] [2]. Here a single spanning tree is proactively maintained within the network. The source then uses this spanning tree to bootstrap path discovery, thus avoiding the need for costly network flooding for reactive path discovery.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we present an initial analysis of the effectiveness of Beacon-Based Routing (BBR) protocols in path discovery for tactical networks. We described the BBR protocols and discussed their scalability with respect to the Proactive and Reactive routing overheads in comparison to other MANET routing protocols. We developed a simulation model to investigate the routing Stretch, which is a measure of the ability of a routing protocol to approach the idealized, best path routing. Our simulation results show that BBR protocols can be effective in tactical MANET environments. We believe that this class of routing protocols for MANETs merits much

attention in the research and development community. We are very excited about this class of routing protocols and will continue to investigate their performance. In a companion paper [4], we investigate the effectiveness of running multiple, simultaneous Beacons within the network. Paths can then be bootstrapped off of any number of these spanning trees.

We have made several assumptions in this study. They include omni-directional antennas, homogeneous radios, promiscuous mode operation, bi-directional channels, and topology-based simulation modeling. Our future work will address these assumptions. Specifically, we hope to further investigate the performance of BBR protocols for MANETs which include heterogeneous channels/networks, non-broadcast channels, and packet level simulation studies with nodal mobility models.

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., Holmer, D. and H. Rubens, *The Medium Time Metric: High Throughput Route Selection in Multi-rate Ad Hoc Wireless Networks* in “Mobile Networks and Applications”, Vol.11, issue 2, pp. 253-266, October 2006.
- [4] Awerbuch, B., Cole, R.G., Holmer, D. and H. Rubens, *Analysis of Multiple Trees on Path Discovery for Beacon-Based Routing Protocols* IEEE PacRim'07 Conference on Communications, Victoria, BC, Canada, August 2007.
- [5] Caesar, M., Castro, M., Nightingale, E.B., O'Shea, G. and A. Rowstron, *Virtual Ring Routing: Network Routing Inspired by DHTs* SIGCOMM'06, Pisa, Italy, September 2006.
- [6] 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.
- [7] Moy, D., *OSPF - Anatomy of an Internet Routing Protocol* Addison-Wesley Press, New York, January 1998.
- [8] Clausen, T. and P. Jacquet, *Optimized Link State Routing Protocol (OLSR)* IETF Request for Comment 3626, October 2003.
- [9] Perkins, C.E. and E.M. Royer, *Ad-hoc On-Demand Distance Vector Routing* Infocomm'99, San Francisco, CA, USA, September 1999.
- [10] Ramanathan, S. and M. Steenstrup, *A Survey of Routing Techniques for Mobile Communications Networks* BBN Corporation Report, September 1996.
- [11] 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.
- [12] Sterberz, J., et.al., *Survivable Wireless Mobile Networks: Issues, Challenges and Research Directions* WiSe'02, Atlanta, Georgia, September 2002.
- [13] Private communications with D. Holmer and H. Rubens, Department of Computer Science, JHU November, 2006.
- [14] <http://wireless.cs.jhu.edu>