BRA: A Bidirectional Routing Abstraction for Asymmetric Mobile Ad Hoc Networks

Venugopalan Ramasubramanian and Daniel Mossé

***Abstract*—Wireless links are often asymmetric due to heterogeneity in the transmission power of devices, non-uniform environmental noise, and other signal propagation phenomenons. Unfortunately, routing protocols for mobile ad hoc networks typically work well only in bidirectional networks. This paper first presents a simulation study quantifying the impact of asymmetric links on network connectivity and routing performance. It then presents a framework called BRA that provides a bidirectional abstraction of the asymmetric network to routing protocols. BRA works by maintaining multi-hop reverse routes for unidirectional links and provides three new abilities: improved connectivity by taking advantage of the unidirectional links, reverse route forwarding of control packets to enable off-the-shelf routing protocols, and detection packet loss on unidirectional links. Extensive simulations of AODV layered on BRA show that packet delivery increases substantially (two-fold in some instances) in asymmetric networks compared to regular AODV, which only routes on bidirectional links.**

***Index Terms*—Ad hoc network, asymmetry, routing, unidirectional.**

I. Introduction

A FUNDAMENTAL problem in mobile ad hoc networks is asymmetry. Asymmetric or unidirectional links arise in the network for several reasons: Devices transmitting with different powers explicitly cause unidirectional links. Even when the devices are transmitting at the same power, noise sources near a device that affect packet reception at that device more than others may create unidirectional links. Finally, other intractable factors such as barriers and environmental conditions that affect signal propagation also lead to asymmetry.

Recent real-world deployments of ad hoc networks indicate a significant presence of asymmetry in the network. Ganesan *et al.* report that up to 15% of the links in their deployment are unidirectional even when all nodes are transmitting at the same power and no external radio sources exist [5]. Similarly, De Couto *et al.* report that up to 30% of links have asymmetric delivery rate in an indoor deployment of wireless nodes [3]. Finally, Zhao *et al.* find that more than 10% of links have significant asymmetry in their packet delivery rates [18].

Manuscript received April 8, 2002; revised December 15, 2004; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor M. Grossglauser. The work was done at Cornell University, supported in part by DARPA/AFRL-IFGA grant F30602-99-1-0532 and in part by NSF-CISE grant 9703470, with additional support from the AFRL-IFGA Information Assurance Institute, from Microsoft Research, and from Intel Corporation, and in part by the DARPA PARTS (Power-Aware Real-Time Systems) project under Contract F33615-00-C-1736 and through a NSF-ITR medium grant.

V. Ramasubramanian is with Microsoft Research Silicon Valley, Mountain View, CA 94043 USA (e-mail: rama@microsoft.com).

D. Mossé is with the Department of Computer Science, University of Pittsburgh, Pittsburgh, PA 15260 USA (e-mail: mosse@cs.pitt.edu).

Digital Object Identifier 10.1109/TNET.2007.912064

Network asymmetry adversely affects routing in several different ways:

1) Connectivity: Asymmetric networks have fundamentally different connectivity than bidirectional networks. Two nodes may be connected through one or more unidirectional links requiring an alternative path in the reverse direction. Or worse, they may be connected in only one direction with no route in the reverse direction. Ignoring the unidirectional links, and routing solely on the bidirectional links, as many conventional routing protocols do, mitigates the problem but may instead prevent several connected nodes from communicating with each other.

2) Routing Protocols: Standard routing protocols often fail to function or function inefficiently in an asymmetric network. Some routing protocols (e.g., TORA [12]) were primarily designed for bidirectional networks and hence break down in the presence of unidirectional links. Several others (e.g., AODV [14]) function by avoiding the unidirectional links and routing data only along the bidirectional links. A few other protocols (e.g., DSR [8]) have the capability to include unidirectional links in their routes through expensive mechanisms that provide significantly decreased throughput in asymmetric networks.

3) Link-layer Services: In addition to the routing layer, unidirectional links also pose several problems at the lower layers such as the data link and the MAC layers. Common MAC-level schemes for congestion avoidance (RTS-CTS) and packet loss recovery (ACKs) fail for unidirectional links. Moreover, other useful services such as detection of link breaks and discovery of new neighbors provided by some MAC protocols become unavailable to the routing protocols.

This paper first presents a simulation study to quantify the impact of asymmetry on network connectivity and routing performance. We model three different types of asymmetric networks based on the cause of asymmetry: (a) regular links becoming unidirectional due to random irregularities in signal propagation, (b) unidirectional links created by external noise sources, and (c) nodes transmitting at different power.

Our study on asymmetry reveals several surprising insights about connectivity in asymmetric networks. First, we find that routing solely on bidirectional links is highly unreliable; while bidirectional connectivity can often be quite good, it may deteriorate suddenly and cut-off several bidirectional routes leading to a poorly-connected network. Second, a substantial percentage of unidirectional links have short (one to three hop) paths connecting them in the reverse direction. Finally, inclusion of such unidirectional links with short reverse paths significantly increases the stability of the routes and leads to better connectivity overall, without significant overhead.

The key contribution of this paper is a framework to improve connectivity on asymmetric networks and support off-the-shelf routing protocols. This framework, called BRA, uses the insights mentioned above to provide a bidirectional abstraction of the underlying asymmetric network to routing protocols.

BRA takes the approach of discovering and maintaining reverse paths for unidirectional links. Its core is a novel algorithm called Reverse Distributed Bellman–Ford Algorithm (RDBFA), which efficiently searches for reverse routes in a bounded search region around each node. BRA keeps the overhead of maintaining reverse routes low by dynamically adjusting the size of the search region, and thereby the length of the reverse routes, independently at each node based on the prevailing extent of asymmetry around that node.

BRA provides three critical functionalities to facilitate routing in asymmetric networks. First, it improves connectivity between nodes by finding new or better routes through unidirectional links. Second, it provides reverse-route forwarding for unidirectional links, which makes them appear as bidirectional links. This abstraction enables routing protocols to send control packets (such as notifications about discovered routes and detected errors) in the reverse direction as it would on symmetric networks. Finally, it implements critical functionalities that MAC and link layers are often unable to provide in asymmetric networks; namely, recovery of lost packets sent across unidirectional links, proactive detection of new neighbors, and notifications about failed links.

BRA supports conventional off-the-shelf routing protocols with little or no modifications. In this paper, we do not consider the details of integrating BRA within the current protocol stack. Rather, we focus on the principles that are needed to provide such an abstraction but we also provide a proof of concept: the implementation of the well-known AODV routing protocol using BRA; other implementations can be carried out in a similar manner. Extensive evaluation of AODV layered on BRA shows that it obtains a significant increase in the number of reachable destinations (double in some instances) in typical asymmetric networks compared to regular AODV, which only routes using bidirectional links. Moreover, the improved connectivity is obtained at a modest cost and little difference in loss rate and network delay.

This paper has the following organization: Section II provides some background on routing in asymmetric networks and summarizes related work. Section III presents an extensive simulation study of asymmetric network topologies. Section IV discusses the key design features of BRA while Section V describes how BRA enables asymmetric routing. Section VI presents an evaluation of AODV over BRA, and Section VII concludes.

II. Background and Related Work

We first provide a brief overview of routing in asymmetric networks and currently-proposed approaches to handle network asymmetry.

A. Routing Protocols and Network Asymmetry

Routing protocols are typically designed for efficient operation on bidirectional networks. Consequently, they either totally fail on asymmetric networks or operate with high overhead and low throughput. This section describes how the well-known routing protocols deal with network asymmetry.

1) Proactive Link-State Protocols: Link-state protocols such as OLSR [2] maintain a view of the network topology at each node; nodes broadcast their views of the topology to their neighbors and in turn update their topology views based on their neighbors' state. Clearly, with a complete view of the network, link-state protocols do not have a problem finding routes in an asymmetric networks. Practical implementations of link-state protocols, however, maintain partial views in order to reduce the $O(n^2)$ worst-case message complexity, where n denotes the number of nodes; the partial views may not have sufficient information to handle unidirectional links.

Bao and Garcia-Luna-Aceves propose a link-state protocol with sufficient view of the topology to handle asymmetry; their protocol maintains an *inclusive cycle* (formed by the link and its reverse route) for each unidirectional link; nodes forward their view of the topology periodically to each upstream neighbor using the reverse routes. However, larger views and extended communication impose a high overhead.

2) Proactive Distance-Vector Protocols: Distance-vector protocols such as DSDV [13], with a worst-cast message size of $O(n)$ in an n-nodes network, are more efficient than link-state protocols. They maintain at each node a *distance vector* consisting of the length and the first-hop neighbor of the shortest path to other nodes; a node broadcasts its distance vector periodically to its neighbors and in turn updates its distance vectors based on its neighbors' distance vectors. Clearly, this protocol assumes that links are bidirectional and fails in asymmetric networks.

Prakash proposes a modified protocol where a node's distance vector is broadcast over multiple hops so that it can reach the nodes that have a unidirectional but direct link to it. However, this protocol increases the worst-case message size from $O(n)$ to $O(n^2)$[15].

3) On-Demand Protocols: On-demand protocols such as AODV [14] and DSR [8] further decrease the routing overhead by maintaining routes only when required for communication. In typical on-demand protocols, a source node S that requires to communicate with a destination node D first initiates a route discovery process, where a route request packet (RREQ) is broadcast typically to the entire network. The destination, or another intermediate node that knows a route to the destination, sends a route reply (RREP) back to the source upon receiving the RREQ. Typically, the RREP is sent along the discovered path in the reverse direction. The state about the discovered path is either retained at each intermediate node (as in AODV) or carried along with each packet (as in DSR). If the current route to the destination breaks, a process similar to route discovery is performed to repair the route.

In an asymmetric network, the route reply packets, and similarly, the route error packets (RERR) used to notify the source about route breaks, cannot be sent along the reverse path if the path has unidirectional links. AODV and DSR handle this situation differently.

AODV avoids any unidirectional links in its paths. It achieves this by tracking the unidirectional links in a *black list*. Nodes discover unidirectional links and add them to the black list whenever an RREP sent along the reverse path fails to return an expected acknowledgment. Once added, a unidirectional link is retained in the black list for a short lifetime. Nodes do not forward RREPs to any node in their black list. While the black-list mechanism enables AODV to approximately identify unidirectional links and avoid them, it poses fundamental limitations.

First, it limits connectivity by precluding communication between nodes that can be reached only through unidirectional links. Second, the black list may prevent discovery of existing bidirectional routes due to stale entries of links not currently unidirectional. Finally, it increases the latency of route discovery since RREPs sent on yet-to-be-discovered unidirectional links cause timeouts.

Marina and Das propose an alternative technique called *reverse path search* to avoid unidirectional links in AODV [9]. While the reverse path search is more efficient than the black-list technique employed by AODV, it does not enable routing protocols to use routes with unidirectional links.

DSR, on the other hand, supports routing on unidirectional links by sending the RREPs on a separate route back to the source. This typically requires an additional route discovery from the destination to the source. However, when intermediate nodes try to send route replies back to the source, the route discoveries often trigger a storm of broadcast packets in the network. Moreover, DSR relies on hop-level acknowledgments for discovering route errors on unidirectional links, for which, it maintains a reverse route at each hop using the DSR protocol itself. Overall, DSR's approach to handle network asymmetry has high overhead and severely limits the throughput of the network.

4) Hybrid Protocols: Finally, hybrid protocols such as ZRP [6] combine both proactive and on-demand routing methods. ZRP uses a proactive routing technique for nodes in a small region called *zone* around each node and an on-demand routing technique for inter-zone routing. Sinha *et al.* extend ZRP to handle unidirectional links by discovering reverse routes for links inside a zone [16]. The resulting protocol has a higher control overhead as it needs to perform proactive routing in an extended zone of twice the earlier size.

B. General-Purpose Support for Network Asymmetry

In contrast to the above protocol-specific solutions for handling network asymmetry, a general-purpose framework such as BRA has the following advantages: First, it enables routing protocols not equipped to handle asymmetry to function with minimal changes. Second, it eases protocol design by removing the responsibility of handling asymmetry from the designers of new routing protocols. Finally, it creates an opportunity for a more suitable and efficient solution to handle asymmetry; for instance, BRA optimizes its cost-benefit tradeoff by supporting asymmetric routing only in regions of poor connectivity and only to the extent required to improve connectivity.

Prior to BRA as well, a few general-purpose frameworks to handle network asymmetry have been proposed. The IETF working group on Unidirectional Link Routing (UDLR) proposes a protocol [4] that invokes tunneling and encapsulation to send multi-hop acknowledgments at the link layer. However, the protocol does not specify what routes are used for the multi-hop tunnels. Nesargi and Prakash propose a similar tunneling-based protocol where control packets are tunneled through multi-hop reverse routes to the upstream nodes of unidirectional links [11]. Their approach uses the same protocol involved in routing data packets to discover and maintain the reverse routes.

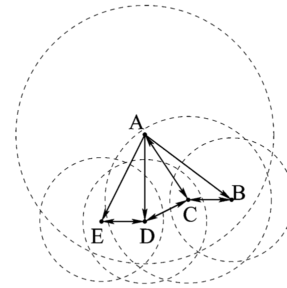

Fig. 1. A unidirectional ad hoc network. $A \rightarrow B$ is a unidirectional link, and $B \rightarrow C \rightarrow A$ is its reverse route.

C. Analysis of Asymmetric Topologies

Jetcheva [7] as well as Marina and Das [9] study connectivity and routing performance of asymmetric networks created by nodes with heterogeneous transmission powers. The study presented in this paper broadly agrees with the observations made in the above studies. Our study provides additional insights into the characteristics of asymmetric networks. First, it simulates network asymmetry due to other causes in addition to heterogeneous transmission powers. Second, it provides a deeper analysis of network connectivity in addition to the average behavior. Finally, it characterizes the effectiveness of using reverse routes of different lengths to improve network connectivity. These observations inspire an efficient and scalable framework for routing in asymmetric networks.

III. Study of Asymmetric Network Topologies

We studied the impact of network asymmetry on connectivity and routing performance through extensive simulations of randomly generated asymmetric network topologies. Each of these topologies represents an instantaneous snapshot of the network; mobility-induced topology changes can be conceived as a series of such snapshots.

A. Notation and Definitions

We first define a few terms before presenting the results of this study: The topology of a network is a directed graph, $D = (V, E)$, where V is the set of nodes and E the set of links in the network. A *link* $A \rightarrow B$ exists between two nodes A and B if B is within the transmission range of A. A link $A \rightarrow B \in E$ is *bidirectional* if $B \rightarrow A \in E$ and *unidirectional* if $B \rightarrow A \notin E$. If $A \rightarrow B$, then A is called an *in-neighbor* of B, and B an *out-neighbor* of A.

The *reverse route* of a link $A \rightarrow B$ is the shortest directed path from B to A, and the length of this shortest path is the *reverse route length* of the link. For example, in Fig. 1 the path $B \rightarrow C \rightarrow A$ forms the reverse route for unidirectional link $A \rightarrow B$. By this definition, bidirectional links have a reverse route length of one hop. If no path exists between B and A, the reverse route and the reverse route length are not defined. The notation $A \rightsquigarrow_n B$ means a path of length n hops exists from A to B in the unidirectional network. The network is *strongly connected* if every link has a reverse route.

We analyze network topologies based on reverse route lengths. A link with reverse route length r is an *r-link*. Thus, *1-links* represent the set of all bidirectional links in the network. The *r-graph* of a network is the sub-graph consisting of the

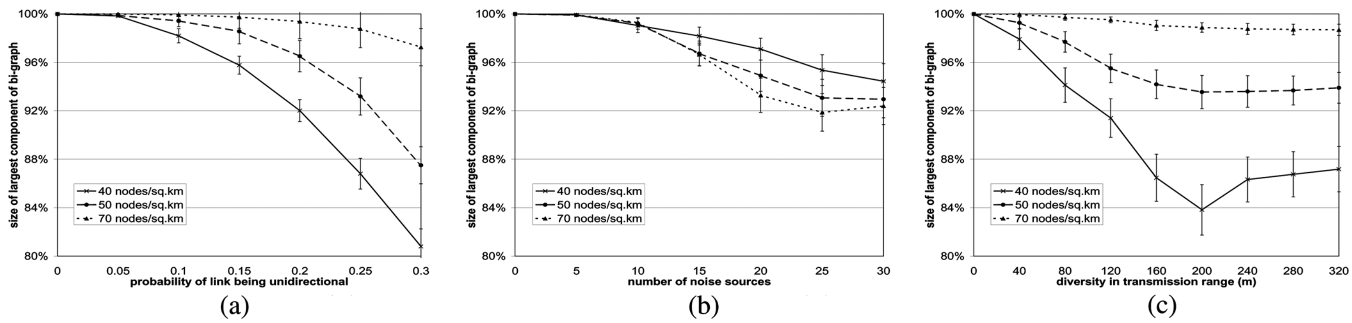

Fig. 2. Normalized size of the largest connected component in the 1-graph for different asymmetry models. These graphs show that asymmetry affects network connectivity significantly. (a) P-Model. (b) N-Model. (c) D-Model.

links with reverse route of length at most r. By this definition, the ∞-*graph* of the network consists of all the links that have a reverse route making each component in the ∞-graph strongly connected. Similarly, the *1-graph* of a network includes only the bidirectional links and represents the network connectivity of routing protocols that route using only bidirectional links.

B. Topology Models

We randomly generated network topologies with parameters commonly used in simulations of routing protocols. Each topology consisted of 100 nodes placed in a square field at random, uniformly distributed locations. We varied the density of the nodes from 30 nodes/km^2 to 100 nodes/km^2 in steps of 10. Similar simulations performed at other scales (50, 200, and 400 nodes) and on rectangular fields yielded results comparable to those presented here, and we omit them.

We modeled unidirectional links for three different causes of network asymmetry:

P-model (probabilistic model): This model simulates unidirectional links created by random irregularities in signal propagation due to ambient conditions. This model takes a basic, bidirectional topology with a transmission range of 220 m for each node and then probabilistically converts links to become unidirectional. We varied the probability of making links unidirectional from 0 to 0.3 in steps of 0.05.

N-model (noise model): This model simulates unidirectional topologies created by external radio sources that increase noise and congest some nodes. In this model, the noise sources are randomly distributed throughout the network with uniform probability. Each noise source has a power 50 times smaller than the nodes in the network. We varied the number of noise sources from 0 to 30 in steps of 5.

D-model (diversity model): Finally, the D-model simulates unidirectional topologies caused by diversity in the transmission power of nodes. We define the *diversity* D of a topology as the difference between the maximum and the minimum transmission ranges of the nodes in the network. We then assign each node a transmission range picked randomly (uniform distribution) from a set of transmission ranges in the interval $[N - \frac{D}{2}, N + \frac{D}{2}]$, where N is the nominal transmission range (set to 220 m). We varied the value of the diversity between 0 m and 320 m. Since, radios typically have discrete levels of transmission powers, we picked the transmission ranges from a discrete set as well. For example, a diversity of 80 m with discreteness of 40 m means the transmission ranges are randomly drawn from the set: 180 m, 220 m, and 260 m.

For each set of parameters described above, we analyzed 500 randomly generated topologies. The Sections III-C– describe the observations made from this analysis. In the graphs we present, each data point represents the average over 500 samples. Error bars represent 99% confidence intervals. However, since the results were not sensitive to the choice of discreteness, we present here only the results for picking transmission ranges in steps of 40 m.

C. Bidirectional Connectivity

We first examine the network connectivity provided by bidirectional links. In this analysis, we use the size of the largest strongly connected component as a measure of connectivity in a network. Fig. 2(a)–(c) show the average size of the largest connected component of the 1-graph at different densities for the P-model, N-model, and D-model respectively. The values plotted are normalized to the average size of the largest connected component in the ∞-graph.

These graphs quantify the drop in connectivity, as unidirectionality increases. We note that the largest connected component becomes smaller in the bidirectional sub-graph. For example, at a density of 50 nodes/km^2, the average size of the largest component in the 1-graph drops to 93% in the N-model and the D-model, and even lower in the P-model. Moreover, the decrease in bidirectional connectivity due to asymmetry is sensitive to the density of the network; the bidirectional connectivity is affected less in denser networks, except in the N-model, where the connectivity appears to be more affected because the noise sources also become denser. However, at lower densities, the bidirectional connectivity can be quite adversely affected; for instance, more than 15% decrease at 40 nodes/km^2) in the D-model and the P-model. While the average decrease in the bidirectional connectivity of asymmetric networks may not appear alarmingly significant, a deeper analysis of the topologies paints a much grimmer picture. Fig. 3(a) shows a histogram of the size of the largest component in the 1-graph[1] of the 500 randomly generated topologies with diversity 200 m and density 50 nodes/km^2. The x axis gives the size of the largest component, and the y axis shows the number of topologies with the corresponding size for the largest component.

[1]We will discuss the 2- and 3-graph results below; we focus on the bidirectional network here.

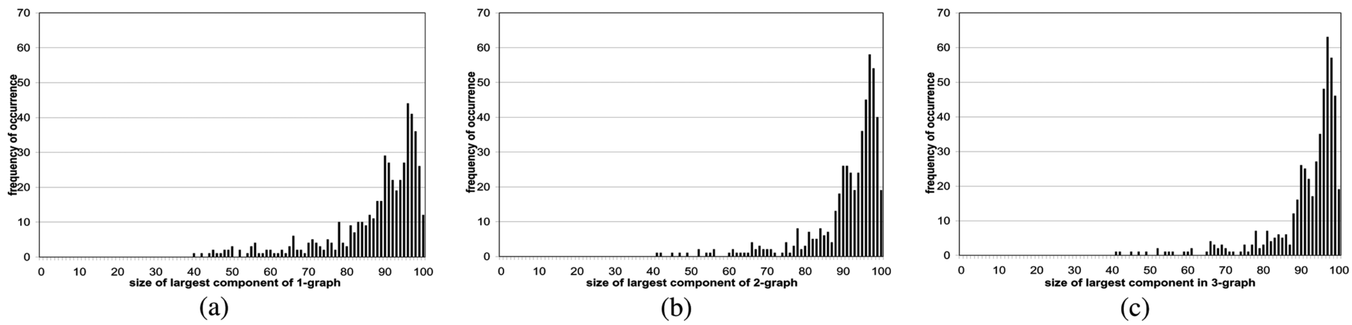

Fig. 3. Histogram of the size of the largest component for diversity 200 m and density 50 nodes/km^2. These graphs show that connectivity is heavy-tailed, but the tail gets lighter as asymmetric links are included. (a) 1-Graph. (b) 2-Graph. (c) 3-Graph.

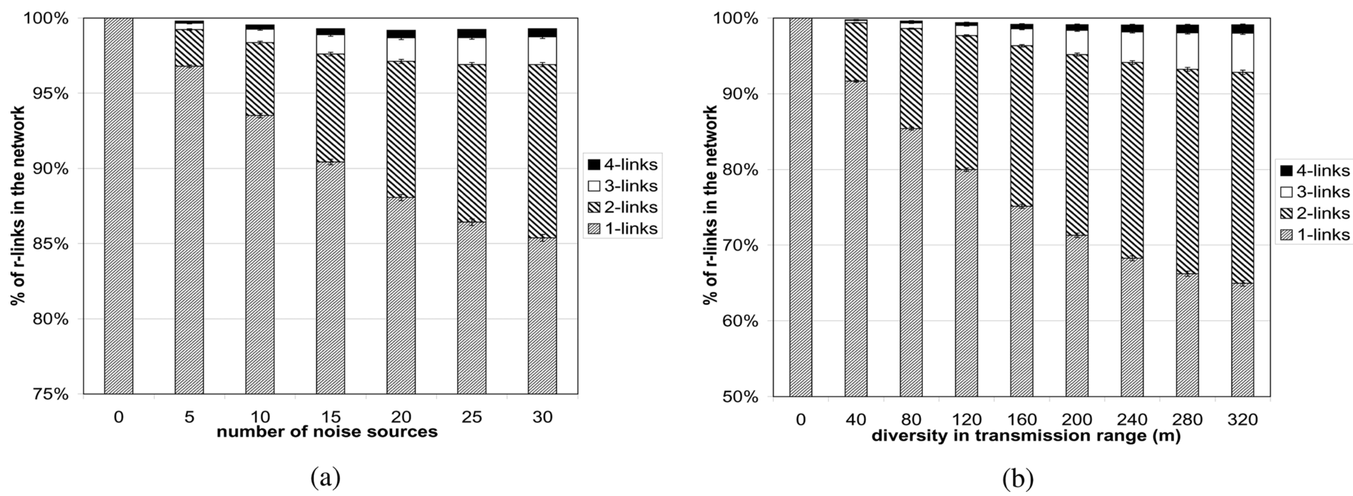

Fig. 4. Average distribution of r-links for a density of 50 nodes/km^2. These graphs show that the number of r-links in the network decreases with r. (a) N-Model. (b) D-Model.

Fig. 3(a) indicates a clustering of values towards the right implying that the size of the largest component is usually very high (> 90). However, the discerning feature of the graph is the heavy tail of the distribution, which goes to very low sizes for the largest component. This figure suggests that while the bidirectional connectivity is mostly tolerable, it could suddenly turn into a scenario of really poor connectivity. The frequency distributions for the N-model and P-model (not shown here) show a similar heavy-tailed distribution of bidirectional connectivity.

D. Link Asymmetry

Before we examine how unidirectional links might help improve network connectivity, we analyze the prevalence of unidirectional links in typical ad hoc networks. The graphs in Fig. 4(a) and (b) show the percentage of r-links in the network for different values of r in the N-model and D-model respectively for topologies with density of 50 nodes/km^2. The Y axis shows the percentage contribution of each r-link (for $r = 1$, 2, 3, 4) in the total number of links with reverse routes in the topology. The contribution of r-links for $r > 4$ is negligible and hence not shown.

These graphs indicate two important trends in asymmetric topologies. First, a significant percentage of links in these topologies are unidirectional $(r > 1)$; Fig. 4(a) shows an increase in the percentage of unidirectional links from 0% to 15% as the number of noise sources varies from 0 to 30, while Fig. 4(b) shows an increase in the percentage of unidirectional links from 0% to 33% as the diversity increases from 0 m to 320 m. Second, the percentage of r-links decreases sharply with increasing r; in both figures, up to 97% of the links have short (i.e., ≤ 3) reverse routes.

E. Connectivity With Unidirectional Links

Finally, we examine the benefit of including unidirectional links in routing. Fig. 5(a) and (b) show the average size of the largest connected component for different values of r in the N-model and D-model respectively, for a density of 50 nodes/km^2. The sizes shown in the figure are normalized with respect to the size of the maximum component in the 1-graph.

Not surprisingly, these figures show that inclusion of unidirectional links increases the size of the largest component. As earlier, the figures show the diminishing returns: as r increases, the connectivity gets better, but on an average the improvement is smaller for higher values of r.

Moreover, including unidirectional links also makes connectivity less heavy-tailed. Fig. 3(b) and (c) show the histogram of the size of the largest connected component in the 2-graph and 3-graph of topologies with diversity 200 m and density 50 nodes/km^2. Compared to Fig. 3(a), which shows the histogram for the 1-graph, more values shift to the right as r increases; that

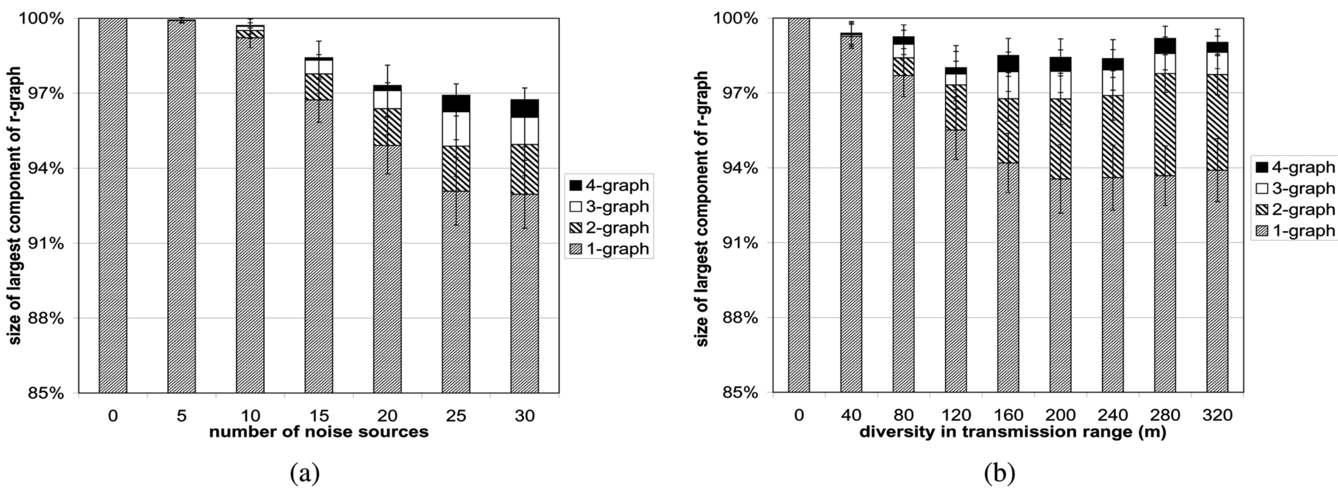

Fig. 5. Normalized size of the largest connected component in r-graphs for a density of 50 nodes/km^2. These graphs show that inclusion of unidirectional links improves network connectivity. (a) N-Model. (b) D-Model.

is, the tail in the distribution is lighter indicating that the standard deviation of the size of the largest component gets smaller as r increases.

F. Summary of Observations

The above analysis of asymmetric topologies drawn from typical scenarios provides valuable insights. First, bidirectional connectivity worsens with increasing asymmetry; the average impact is quite adverse in sparse networks. Second, the heavy-tailed nature of connectivity means that it can suddenly change from good to worse due to mobility. Consequently, routing protocols such as AODV that ignore unidirectional links might suffer from poor connectivity from time to time.

The study also shows that inclusion of unidirectional links, and their corresponding reverse routes, improves both the average connectivity as well as the variance in the level of connectivity. We also observed the diminishing returns in the reverse route length. Since the cost of maintaining the reverse routes is proportional to their lengths, using unidirectional links with shorter reverse routes provides better benefit for cost. However, since scenarios of poor connectivity are infrequent, a routing mechanism that includes unidirectional links would benefit from using them only when really required.

IV. BRA Bidirectional Routing Abstraction

Based on the above insights, we present the *bidirectional routing abstraction* (BRA), a framework that provides improved connectivity and routing performance in asymmetric networks. BRA seeks to improve the efficiency of off-the-shelf routing protocols typically designed for bidirectional networks to function efficiently on asymmetric networks. To that end, BRA provides a bidirectional abstraction of the underlying unidirectional links. The central feature of BRA is an adaptive and scalable technique to maintain reverse routes for unidirectional links. The rest of this section describes this central technique and explains how BRA provides the necessary functionality to enable routing protocols to operate on asymmetric networks.

A. Reverse Distributed Bellman–Ford Algorithm

Finding reverse routes for unidirectional links in an asymmetric network is non-trivial. While it may appear that a straightforward application of a standard distance-vector or link-state algorithm will provide the necessary reverse route information, several problems arise while applying them in an asymmetric network.

For instance, the *Distributed Bellman–Ford Algorithm* is a well-known distance-vector algorithm to obtain the shortest routes between pairs of nodes in a bidirectional network. This algorithm has practical advantages because it works asynchronously and is guaranteed to converge eventually if the network is not partitioned and remains stable for sufficient time. In this algorithm, each node B broadcasts its currently known distances to other nodes in the network to its neighbors. When a node A receives this *distance-vector message* from one of its neighbors, it recalculates its minimum distances to other nodes as follows: If the current known shortest distance from A to another node C is more than one hop longer (to include the hop $A \rightarrow B$) than the distance advertised by B to C, then A discovers a new shortest path to C through B.

However, the above algorithm fails in the presence of unidirectional links. For instance, if $A \rightarrow B$ but not $B \rightarrow A$, then A would never receive the distance-vector message from B and thus will never be able to discover the shortest hop path to C through B.

BRA finds reverse routes through a modified version of the above algorithm called the *Reverse Distributed Bellman–Ford Algorithm* (RDBFA). As the name implies, RDBFA operates by reversing the direction of route discovery; that is, each node aims to find the shortest distance **from** other nodes to itself rather than from itself **to** other nodes. In the previous example, node B tries to learn the shortest path through which other nodes can reach it. B achieves this when it hears A's *reverse-distance-vector* broadcast saying that C can reach A in n hops; B discovers that C can reach B through A in $n+1$ hops since $A \rightarrow B$. If, at B, the previous known route from C is longer than $n + 1$ hops, B can now record the new $n+1$ hop route from C. Furthermore, if there is a unidirectional link $B \rightarrow C$, then C can learn

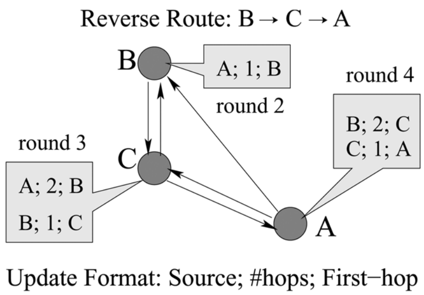

Fig. 6. Reverse Distributed Bellman–Ford Algorithm. This figure illustrates how distance vectors are propagated in RDBFA, enabling nodes to discover reverse routes. In this example, node B discovers the reverse route $B \rightarrow C \rightarrow A$ of unidirectional link $A \rightarrow B$.

about this new reverse route to B from B's next reverse-distance-vector broadcast.

Each entry in the distance vector includes two values: the length of the shortest route from a node and the address of the first hop in the shortest route from that node. Including the first-hop information provides two benefits: 1), it enables a node to compute the reverse route to its in-neighbors based on local state even though the node cannot always compute the reverse route for other nodes' in-neighbors due to the reversed direction of routing state. 2), as in [1] and [10], it enables RDBFA to avoid routing loops and prevent the *counting-to-infinity* problem that affects classical distance vector algorithms; each node uses the first-hop information to detect routing loops and invalidates such routes from its computations.

The following is a precise definition of RDBFA based on the notations defined in Section III.

- At periodic intervals, each node broadcasts to all its out-neighbors a *distance-vector message* containing the shortest paths of its knowledge from other nodes to itself.
- A node B receiving a distance vector from one of its in-neighbors A extracts the following information:
  - — The reverse route from B to A; B obtains this information from the entry for the route from B to A if the distance vector includes such an entry.
  - — If the currently known route from node C to B ($C \rightsquigarrow_i B$) is longer than the route from C to A followed by $A \rightarrow B$ (i.e., $C \rightsquigarrow_j A$ and $i > j + 1$), then B sets the newly found shorter route from C to B through A.

Fig. 6 illustrates the RDBFA algorithm with empty initial state at each node. First, B learns that $A \rightarrow B$ from the first message it receives from A. However, B does not yet know the reverse route to A. In the second round, B broadcasts a distance-vector message, which indicates to C that A can reach B in one hop. The message from C in the third round carries two distance updates: $B \rightsquigarrow_1 C$; $A \rightsquigarrow_2 C$. Finally, A broadcasts the following distance updates: $C \rightsquigarrow_1 A$; $B \rightsquigarrow_2 A$. When B hears this distance update from A, the information cycle is complete, and B discovers the 2-hop reverse route to A. Additionally, B uses the first hop information in the distance-vector to compute the reverse route $B \rightarrow C \rightarrow A$.

B. Reverse Route Maintenance Protocol

BRA uses the above distance-vector algorithm RDBFA for reverse route maintenance. Distance-vector algorithms are more efficient as they exchange an order of magnitude less information ($O(n)$) than link-state algorithms ($O(n^2)$). BRA further reduces the overhead of reverse route maintenance by restricting the spread of distance vectors to a bounded region around each node called *locality*. The locality, defined by a *radius* r, includes all nodes that A can reach in r hops or fewer. For instance, locality radius 1 means the reverse routes found are all of one-hop length and only include bidirectional links. The radius also helps to control the maximum length of reverse routes discovered.

Reverse route maintenance in BRA is proactive. It proceeds through periodic exchanges of distance-vectors between nodes, once every *update_interval*. A full distance vector update from node A as described earlier contains an entry for each node in whose locality A exists. But, unchanged entries are redundant across consecutive updates. BRA optimizes the protocol by letting each node broadcast only the changes in the route information rather than the complete information in each round. However, to account for message loss, nodes broadcast each change in route information multiple times (twice currently).

While the above optimization reduces the average size of an update message significantly, there are cases where broadcasting incremental changes alone is insufficient. For instance, a node might have moved into a new locality and require reverse route information for its new in-neighbors. To help this case, BRA forces each node to broadcast a full distance-vector message infrequently, once every *complete_update_interval*.

In addition to spreading distance-vectors, the periodic messages enable nodes to discover new in-neighbors and loss of existing links. A failure to receive at least *update_loss* consecutive update messages from an in-neighbor A is taken as an indication of link breakage. Link breaks may in turn invalidate existing reverse routes and trigger new distance updates.

Overall, a periodic message in BRA could be one of three kinds: (a) infrequent, complete distance vectors with information of the order of number of nodes in the locality once every complete_update_interval, (b) incremental distance updates containing only the latest changes in the distance-vector once every update_interval, or (c) an empty message called *hello message* to enable link status sensing in the absence of any changed routing information during an update_interval. Note that either message in (b) or in (c) are sent, but not both.

C. Dynamic BRA (DBRA)

The performance and overhead of BRA hinges on the value of the locality radius, because nodes use the radius to prune the size of their updates. If a node A with radius r has a path of length $d > r$ to node B, then node B need not store or broadcast any distance information about A.

The analytical study presented in Section III suggests that a small value of 2 to 3 hops is a reasonable choice for the locality radius. However, the heavy-tailed nature of connectivity suggests that there may be occasions when reverse routes of even longer length might be required to obtain good connectivity. Moreover, using a single global value for the radius entails two high overheads: (a) the size of update packets typically increases polynomially with the radius ($O(r^2)$ in a two dimensional topology), and (b) choosing a single global value for the locality radius in a dynamic manner would be quite expensive.

Instead of picking any single (perhaps ill-suited) value for the locality radius, dynamic BRA (DBRA) sets a locality radius for each node independently and dynamically based on the extent of asymmetry around that node. Picking an independent value for each node keeps the overhead low because asymmetry that affects a local region is taken care of through local reverse route maintenance activity without a global increase in the protocol overhead.

DBRA determines a suitable value for a node A's radius periodically based on two criteria:

a) *Local Criterion* It takes into account the existence and lengths of reverse routes to the node's in-neighbors. If a node does not have reverse routes to more than *asymmetry_tolerance_threshold* percentage of in-neighbors, it chooses the *max_radius*. Otherwise, it chooses a radius of the maximum reverse route length of A's in-neighbors. The tolerance threshold serves as a heuristic to ignore a few asymmetric links from consideration since there may be asymmetric links with no reverse routes in the network.
b) *Global Criterion* DBRA also takes into account the radius of other nodes in a node's locality in order to make sure that nodes help other nodes succeed in reverse route discovery. The radius of a node A must be at least $r - d$ for each node B with radius r and path of length d to A. This condition is necessary for B to obtain reverse routes for its in-neighbors; otherwise, node A might ignore some distance updates crucial to B. For example in Fig. 6, if node B has a radius of 3, then node C must have a radius of at least 2, and node A at least 1; otherwise, B would never learn the path $B \rightarrow C \rightarrow A$.

The final chosen radius of the node is the maximum of the values determined based on local and global criteria. Nodes broadcast their current radius as part of the distance-vector messages.

V. Routing With BRA

BRA provides essential services to enable routing on asymmetric networks. We expect BRA to operate as a sub-layer between the current routing and the MAC layers in the network stack. If the system implementation prohibits modifications to the network stack, then BRA can be integrated with the routing protocol. While this integration will require changes to the routing protocol, the changes are minimal.

BRA should not be completely transparent, that is, a routing protocol layered on top of BRA is expected to be aware of the nature of links they are routing over. This non-transparency allows routing protocols to use the services of BRA intelligently and only upon necessity. For instance, a routing protocol that cannot distinguish whether two nodes are connected through a direct link or a multi-hop reverse route might mistake the reverse route for a fast, direct-hop route and route packets through the longer route; such an action may increase the cost of routing, introduces additional congestion in the network, and decreases the overall throughput of the system.

The services that BRA offers are: *reverse route forwarding*, *reliable packet delivery*, and *link status monitoring*. The rest of this section describes these services in detail.

A. Reverse Route Forwarding

The key service that BRA provides is routing over the reverse routes. This service is intended to enable routing protocols to send control packets over the reverse route of a unidirectional link instead of through an alternative, separately discovered and maintained route. On-demand routing protocols such as AODV and DSR typically send notifications about a newly discovered path through the discovered path itself in the reverse direction. Similarly, the reverse path is also used to send notifications about lost or broken paths. BRA's reverse route forwarding provides the above abilities on asymmetric paths, and helps routing protocols to operate in the same manner on asymmetric networks.

Reverse route forwarding in BRA works as follows: When the BRA sub-layer at node B receives a control packet for node A from the routing layer, BRA computes the reverse route for the link $A \rightarrow B$ from its local state, appends an IP option containing the reverse route to the packet, and transmits the packet to the next node on the reverse route. Intermediate nodes on the reverse route further forward the packets until it reaches the destination. Note that the packet itself carries the entire reverse route since the intermediate nodes cannot always compute the reverse route based on their local state. However, the size of the added IP option is typically small as reverse routes in BRA tend to be short.

B. Reliable Packet Delivery

Packet losses are common in the link layer due to congestion. In bidirectional networks, the MAC layer alleviates this problem either through congestion avoidance (RTS-CTS schemes) or through loss detection and retransmission (ACK-based schemes). BRA provides reliable packet delivery for unidirectional links, where the above MAC layer schemes fail. BRA uses multi-hop acknowledgments and retransmissions for packets sent on both unidirectional links and reverse routes.[2] A packet's destination node sends an acknowledgment back to the source node upon packet reception. The acknowledgement is sent through the reverse routes for packets received on the direct link, while acknowledgments for reverse-routed packets is sent through the direct link. Nodes tag each packet with a sequence number in an IP-option header to identify the packet acknowledged. They maintain separate sequence numbers for each pair of nodes in the locality; the sequence number is set to 0 whenever nodes discover each other in the locality and incremented upon each message transfer.

If acknowledgments are not received in time, nodes retransmit each packet up to *ack_retires* number of times. If a node does not receive an acknowledgment even after retransmissions, it notifies the routing protocol of a packet delivery failure. The time-out value for receiving an acknowledgment is set proportional to the number of links traversed by the packet and the acknowledgment. Thus, if a packet is sent along a direct link $A \rightarrow B$ with an r-hop reverse route, then the time-out is proportional to $r + 1$. Similarly, if the packet is sent along a r-hop reverse route, then the time-out is proportional to $r + 1$. The time-out for a packet sent on a direct link is specified by the *ack_timeout* parameter.

C. Link Status Monitoring

Routing protocols derive significant benefits from proactive discovery of new neighbors and broken links. While MAC pro-

[2]More sophisticated mechanisms such as exponential back-off and simultaneous transmission of multiple outstanding packets, while beyond the scope of this paper, might further improve the throughput of the system.

tocols often provide this functionality for bidirectional links, BRA extends the same ability to asymmetric networks. BRA already performs link status monitoring for reverse-route maintenance through periodic messages as described in Section IV. It simply exports the link status changes discovered this way to the routing protocol.

A routing protocol can use BRA to check whether a link $A \rightarrow B$ is broken. At B, this check is trivial as the BRA already keeps a list of its current in-neighbors. At A, BRA declares the link broken under one of two conditions: 1) A is unaware of any route from B to itself and 2) A packet transmission to B failed. The first condition implies that there is no reverse route from B to A and therefore the link is not suitable for routing. The second condition implies that either the direct link or the reverse route is currently broken; in the latter case, the link is not usable until a new reverse route is discovered. Note that in either case, the node does not definitely know if the direct link ceased to exist. However, just knowing whether the link is useful for routing works well.

D. AODV Over BRA

This section describes how a typical on-demand routing protocol such as AODV would operate on top of BRA. Other protocols can use BRA in a similar manner.

Routing with BRA is straightforward. AODV requires only one key modification to use BRA. It needs to send the control packets such as the route reply packets (RREP) and the route error packets (RERR) through reverse route forwarding. AODV may choose shortest path solely based on the length of the forward path as it would on a bidirectional network. Alternatively, it might benefit from including the length of the reverse routes into path selection metric. Otherwise, route quest broadcast (RREQ) and data forwarding takes place in the same manner as on bidirectional networks. It treats packet-drop and link-break notifications from BRA in the same manner as it treats them from the MAC layer, that is, initiate the route repair process.

AODV derives the following key benefits from BRA: First, it is able to find additional routes not present in the bidirectional view of the network. Second, it does not require the *blacklisting mechanism* to identify and isolate unidirectional links; indeed, BRA with locality radius 1 can provide the same functionality as a black list. Finally, it can reduce data forwarding delay by finding shorter routes including unidirectional links. Note that reverse route forwarding of RREPs may induce an additional delay in receiving route replies at the source and increase the latency for route discovery. The Section VI evaluates the benefits AODV derives from BRA.

VI. Evaluation

We evaluated the routing performance of AODV layered on BRA (BRA-AODV) and DBRA (DBRA-AODV) and compared it with BL-AODV, which uses the black-list technique for identifying and avoiding unidirectional links. This section reports the observed performance in terms of the number of routes established, percentage of successful packet delivery, latency, and the control overhead for routing data packets and for maintaining reverse routes in various network conditions.

We implemented BRA, DBRA, and AODV in GloMoSim [17], a scalable, packet-level wireless simulation environment. Our AODV implementation follows the standard protocol specification [14] ; it implements suggested optimizations such as expanding-ring search, route reply generation at intermediate nodes, and blacklisting for unidirectional links, but excludes gratuitous route reply generation for improved routes and local route repair (route-repairs are currently initiated by the source node).

TABLE I
Simulation for Parameters of BRA and DBRA

parameter	value
update_interval	500 ms
complete_update_interval	4500 ms
update_loss	3
ack_timeout	15 ms
ack_retires	3
asymmetry_tolerance_threshold	15 %
max_radius	8

A. Setup

1) Protocol Parameters: We used the parameters listed in Table I for BRA and DBRA and standard parameters [14] for AODV and AODV-BL (AODV with blacklisting). In addition, we varied the locality radius of BRA from 1 to 8. Note that BRA-1 only routes over the bidirectional links like AODV-BL where as BRA-8 finds almost all the reverse routes that exist in the network; DBRA dynamically adapts the radius at each node based on local asymmetry conditions.

2) Physical and MAC Layers: We set the simulated radio to correspond to the WaveLAN radio hardware with a transmission range of 220 m, bandwidth of 2 MHz, and the two-ray signal propagation model. For the MAC layer, we used IEEE 802.11 for AODV-BL and CSMA for BRA-AODV. Our choice for the MAC protocols is restricted by the functionalities provided by the protocols; IEEE 802.11 is unsuitable for unidirectional routing in BRA-AODV where as CSMA does not provide the necessary collision avoidance or recovery for AODV-BL.

3) Topology and Asymmetry: We simulated topologies similar to those used by the network-asymmetry analysis described in Section III, but only present results for the topology of 80 nodes distributed uniformly at random in an area of 1300 m $\times$ 1300 m. The transmission range of each node was based on the D-model, that is, drawn randomly from the set $[N+\frac{D}{2}, N-\frac{D}{2}]$ for a nominal range $N = 220$ m and diversity D. We simulated six different values of diversity D: 0 m, 80 m, 160 m, 240 m, 280 m, and 300 m.

4) Application: We used a Constant Bit-Rate Generator (CBR) application to initiate data transfers. We set up data transfers between 20 randomly chosen sources and destinations, where each data transfer started randomly between 50 s and 150 s, periodically sent 200 data packets of a size randomly chosen between 64 B and 1024B, and terminated after 200 s. That is, we simulated at most 350 seconds.

5) Mobility: We used the usual random-waypoint model to simulate nodes in motion, where nodes repeatedly move using the following algorithm. A node chooses a random destination point and a random speed (between a *maximum* and a *minimum* value), moves towards that destination with the chosen speed, and, after reaching the destination, waits for a specified *pause time* before repeating this random motion. We simulated two mobility scenarios: a *low-speed or pedestrian-walking* scenario

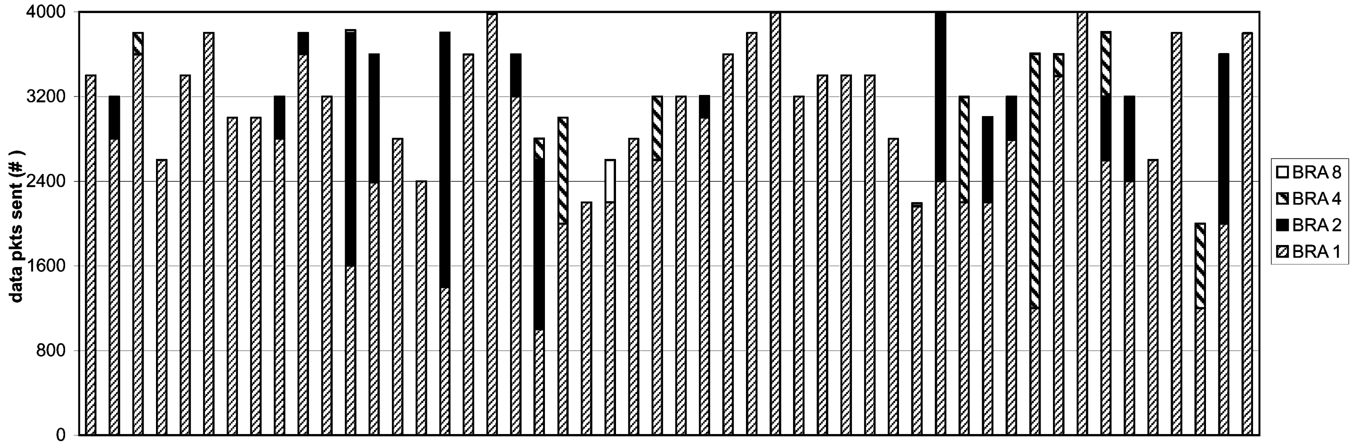

Fig. 7. Number of data packets sent in 50 different trials for diversity 280 m, no mobility, and different radius values. This figure shows that BRA often improves connectivity using unidirectional links.

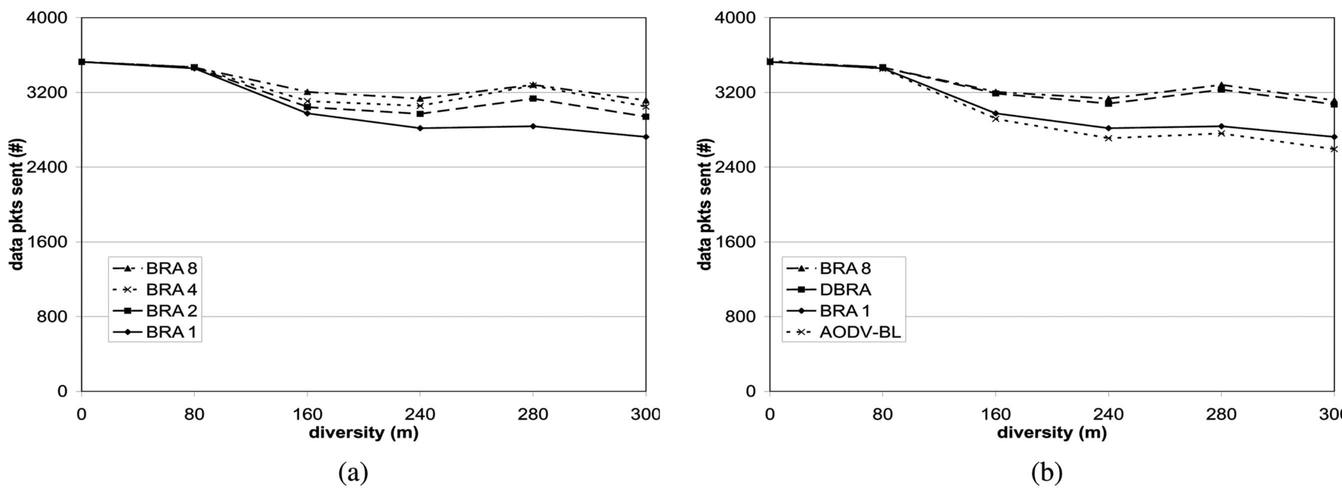

Fig. 8. Average number of data packets sent versus diversity when all nodes are stationary. These figures show that BRA can find more routes using unidirectional links. (a) BRA [1,2,4,8]. (b) BRA-AODV versus AODV-BL.

with speeds between 4 kmph and 8 kmph and a *high-speed or city-vehicular* scenario with speeds between 35 kmph and 45 kmph. For each scenario, we varied the pause time from 0 s to 360 s in steps of 60 s; the pause time of 360 s corresponds to a static environment.

B. Results

The simulations presented in this section were repeated 50 times with different seeds for random number generation in order to account for the high variance in the measured quantities.

1) Connectivity: We quantify the connectivity of a protocol in terms of the number of routes it can establish between nodes in an existing network topology. However, since the number of routes is difficult to count when network topologies are constantly changing due to mobility, we measure connectivity by the number of data packets the source nodes attempt to send. Since AODV sends data packets only after routes are found, this number is a reasonable measure of connectivity.

Fig. 7 shows the number of data packets sent by BRA-AODV in each of the 50 trials for a diversity of 280 m and stationary nodes. It plots cumulative increase in the number of data packets sent as the radius of BRA is increased. This figure highlights the heavy-tail distribution of connectivity in the presence of unidirectional links. In 25 out of 50 trials, using BRA with radius higher than 1 provides no improvement in the number of routes found. However, in the other 50% of cases, using BRA with radius 2 or higher improves the number of data packets sent substantially, sometimes by more than double.

We next present average characteristics of the protocols over the entire simulation. Fig. 8(a) shows the average number of data packets sent by BRA-AODV for different values of radius and diversity when all nodes are stationary. Increasing the radius from 1 to 2 increases the number of routes established substantially, especially when the diversity is large and network asymmetry high. For example when diversity is high, BRA-AODV with radius 2 provides about 10% average improvement over BRA-AODV with radius 1, which only uses bidirectional links. Increasing the radius of BRA beyond 2 further increases the number of data packets sent although the average improvement gets smaller with increasing radius indicating diminishing returns.

Fig. 8(b) shows the average number of data packets sent by BRA-AODV and DBRA-AODV in comparison to AODV-BL

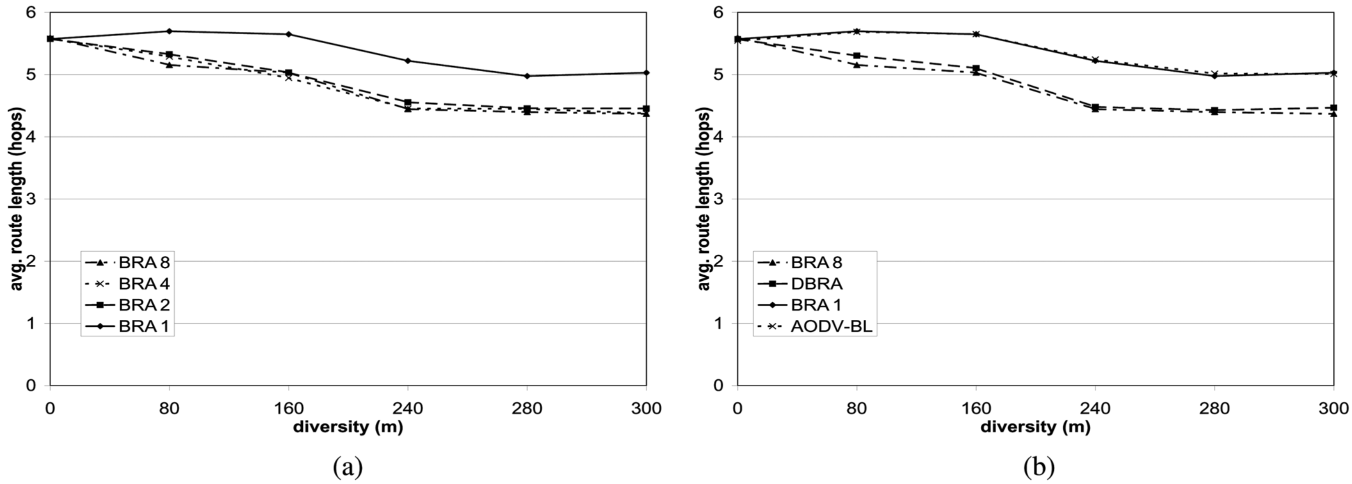

Fig. 9. Average route length versus diversity when all nodes are stationary. These graphs show that BRA finds shorter routes by leveraging unidirectional links. (a) BRA [1,2,4,8]. (b) BRA-AODV versus AODV-BL.

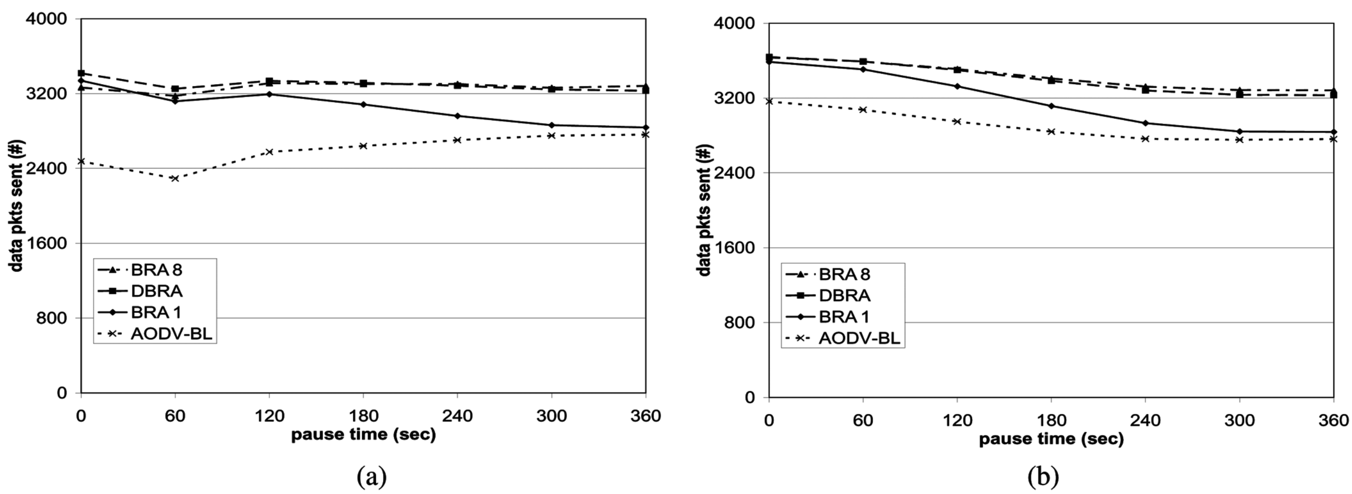

Fig. 10. Average number of data packets sent versus pause time for diversity 280 m. These graphs show that BRA and DBRA can find more routes than AODV-BL in the presence of moderate mobility. (a) High-speed mobility. (b) Low-speed mobility.

for different values of diversity, when all nodes are stationary. As expected, BRA-1 finds roughly the same number of routes as AODV-BL. The marginally lower values for AODV-BL is because AODV-BL initially wastes several route discovery attempts to identify the unidirectional links before it can establish routes correctly. More importantly, the number of data packets sent by DBRA-AODV is comparable to BRA-AODV with radius 8 for all diversity scenarios simulated indicating that DBRA-AODV, which dynamically adjusts the radius depending on the extent of asymmetry in the network, adapts effectively.

While the previous figures showed the number of routes established, Fig. 9(a) and (b) show the average length of the routes found by BRA, DBRA, and AODV-BL. As expected, BRA-1 and AODV-BL have the same average route length since they both use only bidirectional links. However, BRA with higher radius and DBRA find shorter routes using unidirectional links.

Finally, Fig. 10(a) and (b) show the average number of data packets sent in the presence of mobility (high-speed and low-speed scenarios respectively) for a diversity of 280 m. These figures lead to the following observations (1) The number of data packets sent by AODV-BL significantly decreases as the pause time decreases, because the black-list scheme is less effective in identifying unidirectional links in the presence of mobility. Comparatively, even BRA-1 provides a significantly better performance (up to 35% improvement in number of data packets sent). (2) BRA-8 sends more data packets on average than BRA-1, as expected. However, the improvement decreases with increasing mobility. This is a consequence of a cost-performance trade-off in picking the update_interval; a small update_interval would be more effective in maintaining reverse routes in a high mobility environment but at the expense of increased control overhead. (3) Finally, the number of data packets sent by dynamic BRA is comparable to BRA-8, at all mobility and speed scenarios, showing the effectiveness of DBRA.

2) Overhead: Next, we evaluate the overhead incurred by BRA and DBRA. We first discuss the overhead incurred by the periodic update protocol of BRA and DBRA. The total number of periodic packets sent during each trial is a constant (720 per node) since each node transmits at least a hello packet during each update_interval. Hence, we quantify the overhead by the average size of the periodic packets. As a side effect, the size

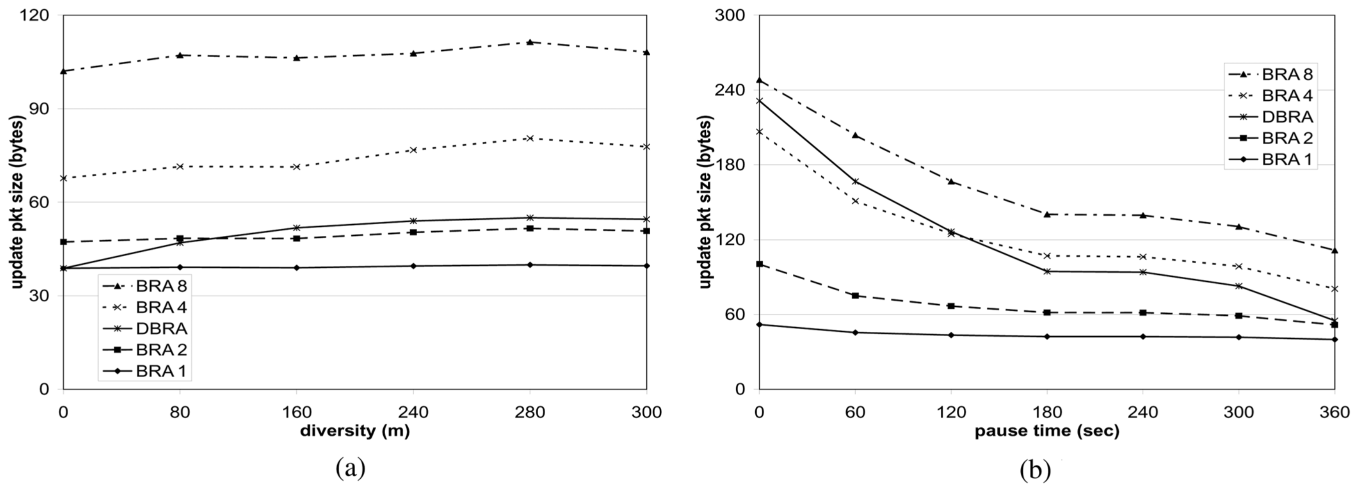

Fig. 11. Average size of update packets. These graphs show that DBRA adapts its overhead based on network conditions. (a) Update Overhead versus Diversity (no mobility) (b) Update Overhead versus Mobility $(\text{diversity} = 280\text{ m})$.

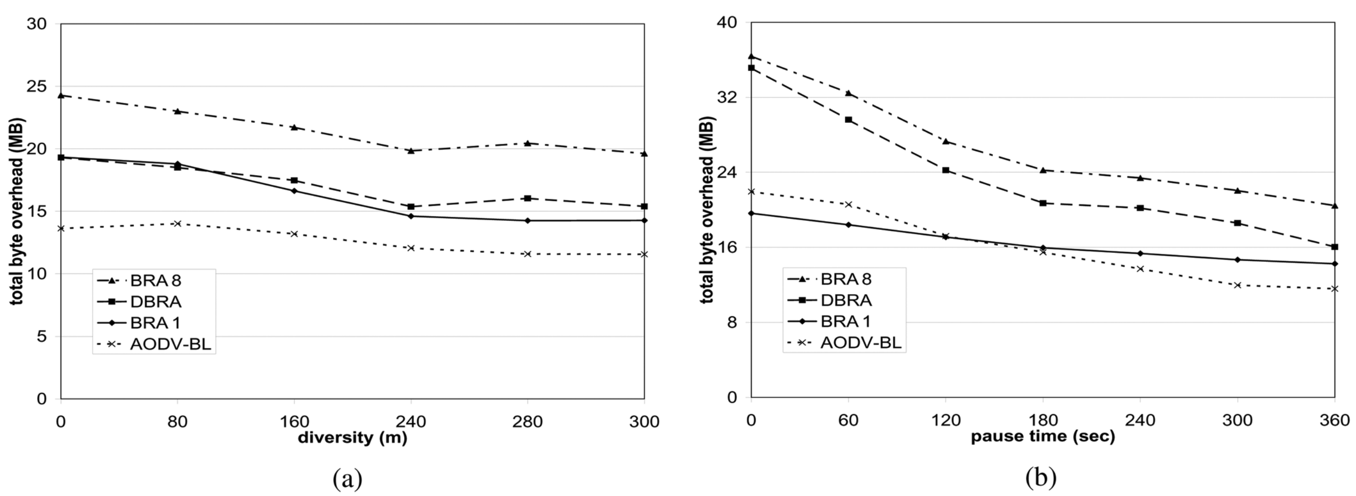

Fig. 12. Total byte overhead. These graphs show that DBRA adapts its overhead based on network conditions. (a) Total overhead versus diversity (no mobility). (b) Total overhead versus mobility $(\text{diversity} = 280\text{ m})$.

of the periodic packet also indicates the extent of congestion introduced by the update protocol.

The average size of a BRA update packet depends on the number of nodes in the locality and, consequently, increases as BRA radius increases. However, it varies little with diversity since the number of nodes in the locality does not change much with diversity. In contrast, for dynamic BRA, the average size of update packets increases with increases in network asymmetry. However, both BRA and DBRA can be expected to have larger update packets with increasing mobility, because more link failures trigger more incremental updates.

Fig. 11(a) plots the average size of a periodic packet (including the MAC layer header) generated by BRA and DBRA for different values of radius and diversity, when all nodes are stationary. DBRA's update packets become larger as diversity, and consequently the asymmetry, increases indicating that it adapts itself to different network conditions.

Fig. 11(b) shows how the average size of a periodic packet generated by BRA and DBRA varies with pause time in the high-speed scenario, for a diversity of 280 m. The average size of periodic packets increases with increased mobility as expected. However, the average size of a DBRA update packet increases substantially more and reaches the average size for BRA-8 at high mobility indicating that DBRA employs larger radius as mobility increases. This is a consequence of DBRA trying to find reverse routes for new unidirectional links for which it needs to adaptively learn the reverse route lengths.

While the previous figures compared the overhead of BRA with its dynamic counterpart, Fig. 12(a) and (b) compare BRA and DBRA with AODV-BL by plotting the total control overhead incurred by these protocols. They plot the total byte overhead counted at the MAC layer since BRA and DBRA use a different MAC layer protocol from AODV-BL.

As expected, in the absence of mobility [Fig. 12(a)], the control overhead of BRA-based protocols are higher than AODV-BL due to the overhead involved in sending periodic packets. However, AODV-BL's overhead increases with mobility [Fig. 12(b)] and, surprisingly, even surpasses BRA-1. While these overhead numbers depend on the data traffic in the network, it is worth noting that the periodic BRA-1 protocol can impose lower overhead than the non-proactive AODV-BL protocol while providing the same level of connectivity.

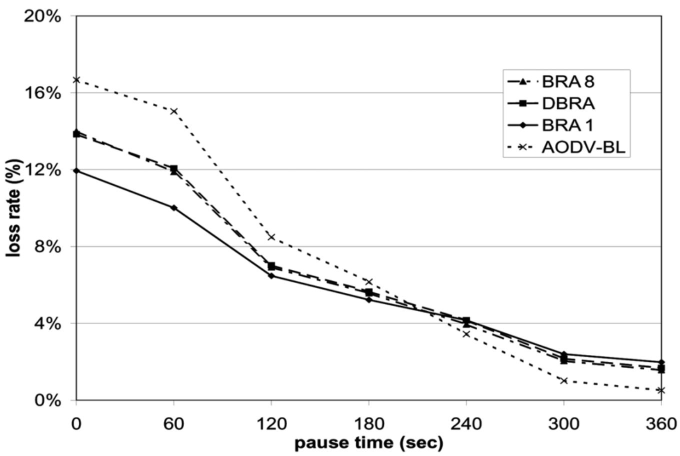

Fig. 13. Average loss rate versus pause time for diversity 280 m and speed 35 to 45 kmph. This figure shows that BRA and DBRA have comparable loss rate as AODV-BL.

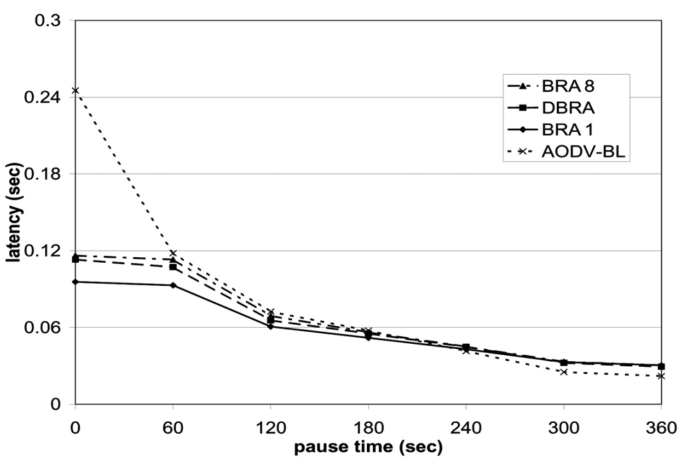

Fig. 14. Average latency versus pause time for diversity 280 m and speed 35 to 45 kmph. This figure shows that BRA and DBRA incur comparable latency as AODV-BL.

3) Loss Rate: Fig. 13 shows that the loss rate incurred by BRA and DBRA are comparable to AODV-BL, in the high-speed scenario for diversity 280 m. At high mobility, AODV-BL has a higher loss rate than BRA due to the inaccurate determination of unidirectional links. For BRA, the loss rate of BRA-1 is lower than the loss rate of BRA with higher radius. Routes in BRA can break due to failures of both the direct links and reverse routes. Loss rate in BRA increases with radius as reverse routes get longer.

4) Latency: Fig. 14 shows the average latency for BRA, DBRA, and AODV-BL in the high-speed scenario, for a diversity of 280 m. The latencies incurred by BRA and DBRA are comparable to AODV-BL. BRA with higher radius incurs more delay than BRA-1 because, increased route breakages (due to reasons mentioned earlier) induce more route discoveries. Overall, the loss rate and latency of BRA and DBRA are as good as AODV-BL, and much better when the nodes are in continuous movement.

C. Summary

This section presented an evaluation of BRA and DBRA in comparison to AODV's black-listing technique for handling network asymmetry. The evaluation indicates that BRA improves AODV's route-finding ability by using unidirectional links across a wide range of network conditions. Even though the average improvement in connectivity is moderate (up to 15% between BRA-1 and BRA-8 for diversity 280 m), in many adverse instances BRA provides a substantial increase (over two times) in the number of routes found. BRA enables routing protocols to effectively compensate for the unpredictable drops in connectivity due to the high variance and heavy-tail in network connectivity.

However, the benefit of using BRA to handle network asymmetry comes with a tradeoff. Increasing BRA radius to find more routes also incurs greater overhead. The choice of BRA radius can be application-specific; an application that can tolerate low connectivity and partitions should chose a small radius whereas an application desiring the best connectivity might pick a larger radius. In general, choosing an appropriate radius for BRA is a nontrivial task and the heavy-tailed nature of connectivity makes it difficult to quantify the cost-benefit tradeoff.

DBRA enables routing protocols to avoid the necessity of choosing the radius *a priori* through an independent and adaptive radius selection heuristic at each node. DBRA provides performance comparable to that of BRA-8 in terms of data packets sent, loss rate and latency, while it adapts its overhead to the extent of asymmetry and mobility in the network: comparable overhead to BRA-1 at low diversity and mobility, but higher overhead as diversity or mobility increases.

Finally, BRA-1 provides an effective mechanism for finding symmetric routes in asymmetric networks. In comparison to the use of black lists for AODV, BRA-1 can identify unidirectional links faster and much more accurately in high mobility scenarios. This leads to an improvement of up to 35% more data packets sent. While BRA-1 achieves improved connectivity with additional control overhead, our simulations showed that in many scenarios BRA-1 also incurs lower overhead than the black list technique.

VII. Conclusion

This paper presented a bidirectional routing abstraction (BRA) to handle unidirectional links that arise frequently in mobile ad hoc networks. BRA provides routing protocols with the familiar bidirectional abstraction that they are typically designed for and thus enables them to operate efficiently on asymmetric networks. Internally, however, it actively uses both unidirectional and bidirectional links to 1) find symmetric routes more effectively than conventional techniques; 2) find new, asymmetric routes substantially increasing the reachability of the network; and 3) find alternate routes with shorter path length.

This paper made three overall contributions: First, it presented a quantitative analysis of how network asymmetry caused by alien radio sources, heterogeneity in transmission power, and random fluctuations in signal propagation that affect conventional MANET routing protocols. Second, it presented the design of BRA, based on a novel protocol to maintain reverse routes for unidirectional links in an efficient and scalable manner. Finally, it showed through extensive evaluation how a typical routing protocol, such as the well-known AODV, layered on BRA achieves superior connectivity in asymmetric networks.

REFERENCES

[1] C. Cheng, R. Riley, S. Kumar, and J. Garcia-Lunes-Aceves, "A loop-free extended Bellman-Ford routing protocol without bouncing effect," in *Proc. ACM SIGCOMM*, Aug. 1989.

[2] T. Clausen and P. Jacquet, "Optimal link-state routing," RFC 3626, Oct. 2003.

[3] D. D. Couto, D. Aguayo, B. Chambers, and R. Morris, "Performance of multihop wireless networks: Shortest path is not enough," presented at the Workshop on Hot Topics in Networking (HotNets), Princeton, NJ, Oct. 2002.

[4] E. Duros, W. Dabbous, H. Izumiyama, N. Fujii, and Y. Zhang, "A link layer tunneling mechanism for unidirectional links," IETF Internet Draft, Network Working Group, 1999.

[5] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "An empirical study of epidemic algorithms in large scale multihop wireless networks," Intel Research, Tech. Rep. IRB-TR-02-003, 2002.

[6] Z. J. Haas and M. R. Pearlman, "The Zone Routing Protocol (ZRP) for ad hoc networks," IETF Internet draft, MANET Working Group, 2001.

[7] J. G. Jetcheva, "Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, 2004.

[8] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*. Norwell, MA: Kluwer, 1996, pp. 153–181.

[9] M. K. Marina and S. R. Das, "Routing performance in the presence of unidirectional links in multihop wireless networks," in *Proc. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*, Lausanne, Switzerland, Jun. 2002.

[10] S. Murthy and J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks," *Mobile Networks and Applications Journal, Special Issue on Routing in Mobile Communication Networks*, vol. 1, no. 2, 1996.

[11] S. Nesargi and R. Prakash, "A tunneling approach to routing with unidirectional links in mobile ad hoc networks," in *Proc. Int. Conf. Computer Communications and Networks (ICCCN)*, Las Vegas, NV, Oct. 2000.

[12] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proc. IEEE INFOCOM*, Kobe, Japan, Apr. 1997.

[13] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proc. ACM SIGCOMM*, Aug. 1994.

[14] C. E. Perkins, E. M. Royer, and S. R. Das, "Ad-hoc on demand distance vector (AODV) routing," RFC 3561, Jul. 2001.

[15] R. Prakash, "Unidirectional links prove costly in wireless ad hoc networks," in *Proc. ACM DIAL-M Workshop*, Seattle, WA, Aug. 1999.

[16] P. Sinha, S. Krishnamurthy, and S. Dao, "Scalable unidirectional routing with zone routing protocol (ZRP) extensions for mobile ad hoc networks," in *Proc. Wireless Communications and Networking Conf. (WCNC)*, Chicago, IL, Sep. 2000.

[17] X. Zeng, R. Bagrodia, and M. Gerla, "Glomosim: A library for parallel simulation of large-scale wireless networks," in *Proc. Workshop on Parallel and Distributed Simulation*, Banff, Canada, May 1998.

[18] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *Proc. Conf. Embedded Networked Sensor Systems (SenSys)*, Los Angeles, CA, Nov. 2003.

Venugopalan (Rama) Ramasubramanian received the B.Tech. degree in computer science and engineering from the Indian Institute of Technology, Madras, and the Ph.D. degree in computer science from Cornell University, Ithaca, NY, in 2007.

He is a Researcher at Microsoft Research Silicon Valley. His research spans the fields of systems and networking.

Daniel Mossé received the B.S. degree in mathematics from the University of Brasilia in 1986, and the M.S. and Ph.D. degrees in computer science from the University of Maryland in 1990 and 1993, respectively.

He joined the faculty of the University of Pittsburgh in 1992, where he is a Professor. His research interest include fault-tolerant and real-time systems, as well as networking. The major thrust of his research in the new millennium is real-time systems, power management issues, and networks (wireless and security). Typically funded by NSF and DARPA, his projects combine theoretical results and actual implementations.