A Link Distance Ratio Based Stable Multicast Routing Protocol for Mobile Ad Hoc Networks

Natarajan Meghanathan

Jackson State University
Jackson, MS 39217, USA
nmeghanathan@jsums.edu

Abstract. We present the design and development of a new multicast routing protocol, referred to as the Multicast Link Distance Ratio (MLDR) routing protocol, which yields stable trees with longer lifetime and without incurring any substantial increase in the number of edges and the hop count per source-receiver path. The proposed multicast protocol is based on the idea of assigning each link a weight, called the Link Distance Ratio (LDR), corresponding to the ratio of the actual physical Euclidean distance between the constituent nodes of the link to that of the maximum transmission range per node. The multicast tree construction procedure of MLDR focuses on discovering source-receiver paths that have the lowest sum of the LDR values of the constituent links. An aggregate of all such source-receiver paths yields the MLDR multicast tree. The lifetime of MLDR multicast trees is 25% - 63% longer than that of the well-known minimum hop based Multicast Ad hoc On demand Distance Vector (MAODV) routing protocol and at the same time the number of edges per tree and hop count per source-receiver path are slightly larger than that of MAODV, by factors of 11% and 8% respectively.

Keywords: Multicasting, Routing Protocol, Link Distance Ratio, Simulation, Stability, Mobile Ad hoc Networks

1 Introduction

A Mobile Ad hoc Network (MANET) refers to wireless networks whose topology dynamically changes with time owing to node mobility, bandwidth and energy constraints. MANETs are deployed for military, mission-critical, disaster-relief and emergency management applications. One characteristic nature of all of these applications is one-to-many communication, referred to as multicast, between the participating nodes. Multicasting can be more formally defined as the communication between a source node and a set of receiver nodes, the latter constituting what is called a multicast group. The source node need not be a member of the multicast group, as is the case in this paper. MANET multicasting is done via a tree or a mesh, determined in an on-demand fashion (i.e., only when a source node has data to be sent to the receiver nodes of the multicast group) through a global broadcast query-reply cycle, often called flooding. A multicast tree connects the source node to all of the

N. Meghanathan et al. (Eds.): CCSIT 2012, Part I, LNICST 84, pp. 253–262, 2012.

receiver nodes of the group such that there is exactly one path between the source node and each of the receiver nodes; some receiver nodes could also end up serving as intermediate nodes of the multicast tree. A multicast mesh connects the source node to the receiver nodes such that there are often more than one source-receiver paths. A multicast mesh is typically an extended multicast tree wherein all the links that exist among the tree nodes are considered to be part of the mesh.

Multicast trees are considered more efficient with respect to link usage, bandwidth and energy consumption as only one copy of the data packet reaches each receiver node of the multicast group and there are no redundant transmissions, unlike meshes. However, a multicast tree is considered broken even if one of the constituent links of the tree is broken. Frequently reconfiguring a communication structure using flooding is an expensive operation in MANETs, owing to their resource constraints. The advantage with meshes is that they are more robust to link failures and provide prolonged connectivity between the source node and the receiver nodes without requiring to be frequently rediscovered. But, there will be redundant transmissions of data packets through more than one path from the source to each receiver node.

The motivation for this research is to determine stable multicast trees that exist for a relatively longer time so that the number of tree discoveries in the network can be minimized and at the same time the link efficiency advantage with trees is retained. Multicast trees have been traditionally determined to be minimum-hop trees connecting the source node with each of the receiver nodes through minimum hop paths. When we analyze for the critical factors that trigger link failures (leading to tree failures) in such minimum hop-based multicast trees, the edge effect [6] has been observed to be a significant factor. In order to connect the source and receiver nodes with the minimum number of hops, the number of intermediate nodes added to the source-receiver path is as minimal as possible; however, this leads to a longer physical distance between the constituent end nodes of the links. As a result, for any given link on a minimum-hop path, the probability that the two end nodes of the link would move away from the transmission range of each other at any time is quite high. To counter the edge effect problem, it would be more prudent to construct the source-receiver paths by including those links whose constituent end nodes are not close to the boundary of the transmission range of each other. Accordingly, we define the Link Distance Ratio as the ratio of the physical Euclidean distance separating the two end nodes of a link and the transmission range of the nodes. Smaller the LDR value, we conjecture the link will be more stable. Likewise, a path with the minimum sum of the LDR values of the constituent links is likely to be more stable than minimum hop paths. This forms the hypothesis of our paper and our hypothesis has been proven to be correct through extensive simulation analysis.

The rest of the paper is organized as follows: Section 2 discusses related work. Section 3 describes in detail, the working of the proposed LDR-based stable multicast routing protocol (MLDR), including the packet structures and the sequence of different phases of tree construction and maintenance. Section 4 describes the simulation environment and presents the simulation results obtained when the MLDR is implemented and run in the ns-2 simulator [7]. We also compare the performance of MLDR with the well-known minimum-hop based Multicast Ad hoc On-demand Distance Vector (MAODV) routing protocol [9]. Section 5 concludes the paper and also outlines future work planned for extending our research on MLDR.

2 Related Work

In this section, we discuss related work in the literature on signal strength based routing in conjunction with stable path routing. The Signal Stability-based Adaptive (SSA) unicast routing protocol [1] characterizes the MANET links into two classes: strong and weak links. Nodes are required to periodically exchange beacons in the neighborhood. The network operates with two thresholds for signal strength: threshold for strong links P_{th}^{strong} and threshold for signal reception P_{th}^{rec}, with $P_{th}^{strong} > P_{th}^{rec}$. If the strength of beacon signal received from a neighbor node exceeds P_{th}^{strong}, then a node categorizes the link with the neighbor as a strong link. If the strength of the beacon is below P_{th}^{strong}; but above P_{th}^{rec}, the link is characterized as a weak link. SSA attempts to discover a stable route comprising only of strong links and if not successful, determines a route considering all the links in the network.

The Route-lifetime Assessment Based Routing (RABR) unicast protocol [2] works by computing a metric called "link affinity" for each link based on the average change in the signal strength of the beacons received within a time window during the recent past. If the average change in the signal strength is positive, then the nodes are assumed to be approaching each other and the affinity of the link is assigned to a high value (theoretically, ∞). If the average change in the signal strength is negative, then the affinity value of the link is the ratio computed by dividing the difference between the minimum threshold for the signal strength required for a link between two nodes to exist and the signal strength of the most recently received beacon with the average change in the signal strength. The affinity value for a path is the minimum of the affinity values of its constituent links.

In [5], the authors propose a signal strength-estimate driven stable path routing protocol wherein the estimated signal strength of the Route Request (RREQ) packets is recorded in the RREQ packets itself at each forwarding node. The estimated signal strength of a path is the minimum of the estimated signal strength of the constituent links on the path as included by the forwarding nodes. The destination chooses the path with the largest estimated signal strength and sends back a Route Reply (RREP) packet on the chosen path. A similar Min-Max approach for stable path routing based on the predicted link expiration time (LET) has been proposed in [3] and [4].

3 Design of the Multicast Link Distance Ratio (LDR)-Based Routing Protocol

The objective of the multicast link distance ratio (MLDR) based routing protocol for MANETs is to determine stable multicast trees that have a longer lifetime and at the same time incur a minimal increase in the number of edges per tree and hop count per source-receiver paths as part of the tree. The key assumptions behind the design and working of MLDR are as follows: (i) MLDR assumes the network is homogeneous in nature and that all nodes operate with an identical and fixed transmission range; (ii) MLDR requires nodes to periodically exchange beacons in the neighborhood so that a node can estimate the distance between itself and each of its neighbors by measuring the strength of the signal received from the neighbor. The signal propagation model used is the "two-ray ground reflection" model [8]; (iii) The Link Distance Ratio (LDR) is computed as the distance between a node and its neighbor node divided by the transmission range of the nodes. At any moment, every node maintains a LDR-table comprising of estimates of the LDR values to each neighbor node based on the latest beacons received from the node; (iv) The LDR of a path is sum of the LDR values of the constituent links of the path and (v) An aggregate of all the paths, with the least sum of the LDR values, connecting a source node to the receiver nodes leads to the desired stable multicast tree.

3.1 Propagation of the Multicast Tree Request (MTREQ) Messages

When a source node has data to send to the multicast group and is not aware of the next hop downstream nodes that are part of the multicast tree, the source node broadcasts a Multicast Tree Request (MTREQ) message to all of its neighbors as an attempt to reach out to the receiver nodes of the multicast group. The structure of the MTREQ message is shown in Figure 1. The sequence number field is used to avoid any loops in the broadcast of the MTREQ message and is a monotonically increasing quantity, incremented by 1, for every MTREQ message originating from the particular source node. The Route Record field stores the IDs of nodes through which the message has propagated, starting from the source node. The Link Distance Ratio field stores the cumulative value (sum of the LDR values) of the constituent links through which the MTREQ has propagated, starting from the source node. The source node initializes the LDR value in the MTREQ message to zero and inserts its own ID in the Route Record field. When an intermediate node receives a MTREQ message of a particular broadcast tree construction process (identified using a combination of the Source Node ID and the Sequence Number fields) for the first time, the intermediate node updates the LDR value in the MTREQ by adding to it the LDR value of its link to the upstream neighbor node from which the message was received. The intermediate node then inserts its node ID to the Route Record field and the MTREQ message is further broadcast to all the neighbor nodes. When an intermediate node receives a MTREQ message that it has already seen, the message is dropped.

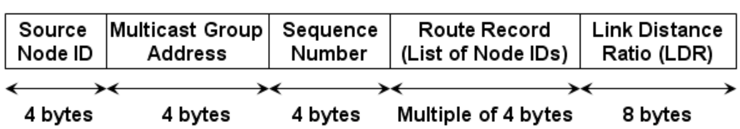


Fig. 1. Structure of the Multicast Tree Request (MTREQ) Message

3.2 Route Selection and Propagation of Multicast Tree Reply (MTREP) Messages

When a member node (of the multicast group) receives a MTREQ message for a particular broadcast multicast tree construction process, the node updates the LDR value in the message by adding to it the LDR value of the link to the upstream node from which the message was received. After waiting for a certain amount of time to receive the MTREQ messages from one or more paths, the node selects the MTREQ message that has the minimum LDR value. The receiver node then generates a Multicast Tree Reply (MTREP) message (shown in Figure 2) that propagates on the reverse path of the sequence of node IDs listed in the Route Record field of the chosen MTREQ message. The MTREP message propagates from the receiver member node to the source node of the multicast process.

Source Node ID	Originating Receiver	Multicast Group Address	Sequence Number	Route Record (List of Node IDs)	Link Distance Ratio (LDR)
4 bytes	4 bytes	4 bytes	4 bytes	Multiple of 4 bytes	8 bytes

Fig. 2. Structure of the Multicast Tree Reply (MTREP) Message

When the MTREP message reaches an intermediate node, the intermediate node checks whether it has an entry for the <*Source Node ID*, *Multicast Group Address*> in its multicast routing table, which is an ordered entry of <*key*, *value*> pairs, where the <*key*> is the tuple <*Source Node ID*, *Multicast Group Address*, *MTREP Sequence Number*> and the <*value*> is the tuple <*Upstream Node*, *List of Downstream Nodes*>. The structure of the multicast routing table maintained at an intermediate node is shown in Figure 3. The *Upstream Node* and the *List of Downstream Nodes* are part of the multicast tree rooted at the Source Node ID. After the <*key*> part of the multicast route entry is properly created or updated based on the most recent value of the MTREP Sequence Number, the intermediate node updates the <*value*> part of the multicast route entry by including the neighbor node from which the MTREP message was received into the *List of the Downstream Nodes* and the next hop neighbor node (that has been listed as the next hop node on the Route Record from the receiver node towards the source node) is included as the *Upstream Node*. If the *Upstream Node* is already listed in the multicast route entry, the MTREP message is just dropped and not forwarded as it would be only tracing a sub-path of the already established optimal path from the intermediate node to the source node. If the next hop neighbor node has been just then updated as the *Upstream Node* in the multicast routing table, the intermediate node sends the MTREP message to that upstream node.

Key			Value	
Source Node ID	Multicast Group Address	MTREP Sequence Number	Upstream Node ID	List of Downstream Nodes

Fig. 3. Structure of the Multicast Routing Table at an Intermediate Node

The source node maintains a multicast routing table of <*key*, *value*> pairs, where the <*key*> is the tuple <*Multicast Group Address*, *MTREQ-MTREP Sequence Number*> of the latest tree discovery process; the <*value*> is the *List of Downstream Nodes* that includes the neighbor nodes that sent it the MTREP messages.

3.3 Multicast Tree Acquisition, Data Transmission and Maintenance

After broadcasting the MTREQ messages, the source node waits for a certain time, called the Tree Acquisition Time, to receive the MTREP messages (originating from the multicast group members) through one or more neighbor nodes. If no MTREP message is received within the Tree Acquisition Time, the source node broadcasts the next MTREQ message (Sequence Number incremented by 1) to its neighborhood. If one or more MTREP messages are received within the Tree Acquisition Time, the source node starts transmitting the data packets through the multicast tree established as part of the MTREQ-MTREP cycle. After the first successful tree discovery procedure, the Tree Acquisition Time is dynamically reset depending on the time incurred to receive the MTREP messages from the multicast member nodes.

A multicast tree is broken even if one of the constituent links of the tree is broken. When an intermediate node could not forward a data packet to even of its downstream nodes in the tree, the intermediate node generates a Multicast Tree Error (MTERR) message and sends it to the source node of the multicast session. In this pursuit, the intermediate node sends the MTERR message (structure shown in Figure 4) to the immediate upstream node in its routing table entry for the particular source and multicast group address. The entry is also then removed from the table. The above process is repeated at every intermediate node (starting from the upstream node of the broken link all the way to the source node) in the tree, as the MTERR message propagates all the way back to the source node. The multicast routing table entries at nodes starting from the downstream node of the broken link, all the way to one or more receiver nodes of the multicast group, are flushed during the propagation of the MTREP message as part of the next broadcast tree construction process.

Source Node ID	Multicast Group Address	MTERR Originating Intermediate Node ID	MTREP Seq. Number	Seq. Number of Dropped Data Packet
4 bytes	4 bytes	4 bytes	4 bytes	4 bytes

Fig. 4. Structure of the Multicast Tree Error Message (MTERR)

4 Simulations

The performance of MLDR has been compared with that of the well-known minimum-hop based Multicast Extension of the Ad hoc On-demand Distance Vector (MAODV) routing protocol [9]. We implemented both the multicast routing protocols (MLDR and MAODV) in the ns-2 simulator (v. 2.32) [7]. The network dimensions are 1000m x 1000m. The transmission range per node is 250m and is the same for all the nodes in the network. The network density is varied by conducting simulations with 50 nodes (low density) and 100 nodes (high density). The nodes are initially assumed to be uniform-randomly distributed in the network.

Nodes move according to the Random Waypoint mobility model [10] with each node moving independent of the other nodes in the network. A node starts moving from an arbitrary location to a randomly chosen destination location within the range [0…1000m, 0…1000m], and moves to the chosen location at a speed uniform-randomly chosen from the range [0,…, v_{max}] where v_{max} represents the maximum node velocity. The v_{max} values used in the simulations are 5 m/s, 25 m/s and 50 m/s representing scenarios of low, moderate and high node mobility respectively. Pause time is 0 seconds. For a given condition of network density and v_{max} values, 5 different mobility profiles were generated. Simulation time is 1000 seconds.

Simulations are conducted with a multicast group size of 3 (small size) and 18 (larger size) receiver nodes. The source node is not part of the multicast group. For each group, we generated 10 lists of receiver nodes and conducted simulations with each of these 10 lists. So, basically, 10*5 = 50 multicast session simulations were run for every combination of network density, mobility (v_{max}) and multicast group size values. Each data point in the plots for the performance metrics illustrated in Figures 5 through 7 are an average of the metric values obtained for these 50 simulations. The traffic model assumed is Constant Bit Rate (CBR); the size of the data packets is 512 bytes and the source sends 4 data packets per second to the multicast group.

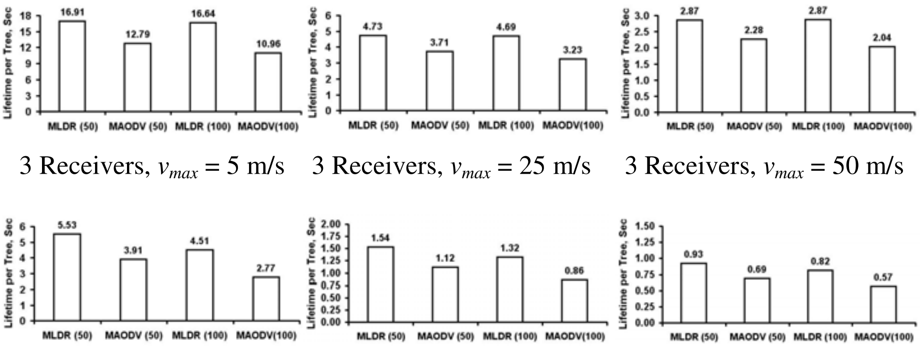
The performance metrics evaluated through the simulations are the following:

- ***Lifetime per Multicast Tree*:** For every multicast tree used during the simulation session, we keep track of the duration the tree exists. The lifetime per multicast tree is the average value of the duration of the multicast trees, over the entire simulation time, across all the simulation conditions corresponding to a particular combination of network density, node mobility and multicast group size.
- ***Number of Edges per Multicast Tree*:** This is the time averaged value of the number of edges in the multicast trees discovered and used over the entire simulation session (i.e., taking into consideration the duration of the trees).
- ***Hop Count per Source-Receiver Path*:** This is the time averaged value of the hop count of the paths from the source node to each of the receiver nodes of the multicast group, computed over the entire multicast simulation session.

4.1 Lifetime per Multicast Tree

For a fixed multicast group size, as the node velocity increases, the gain in the multicast tree lifetime incurred with MLDR over MAODV, decreases. However still, the lifetime of multicast trees incurred with MLDR is at least 25% more than that of the lifetime per multicast tree determined using MAODV. Thus, MLDR yields stable multicast trees compared to the minimum hop based well-known MAODV under all the simulation conditions tested. For a given v_{max}, the gain in the lifetime of multicast tree determined using MLDR compared to MAODV increases with multicast group size. For a given group size, the lifetime per multicast tree determined using both MLDR and MAODV decrease with increase in the v_{max} value. For a given v_{max} value, the multicast tree lifetime for both protocols decreased with increase in group size.

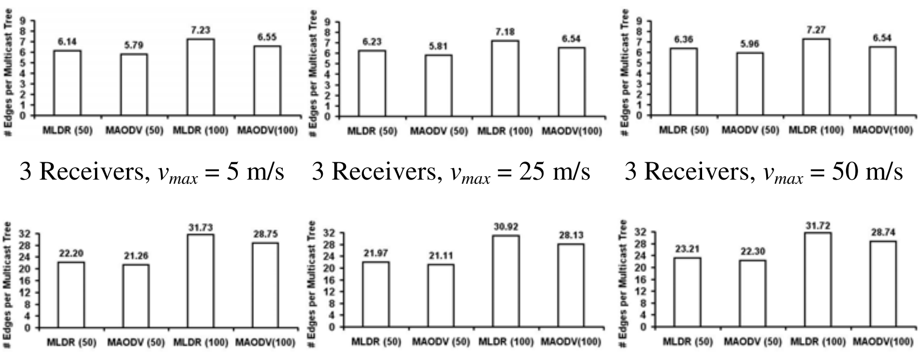
For a fixed v_{max} value and multicast group size, the lifetime per multicast tree determined using MLDR decreases slightly when the network density is doubled. The lifetime of MAODV multicast trees decreases rather more aggressively when the network density is doubled. In the case of both MLDR and MAODV, for fixed node mobility, the decrease in the multicast tree lifetime with increase in network density is more dominant when the multicast group size is larger. But, for both the protocols, for fixed multicast group size, the decrease in the multicast tree lifetime with increase in network density is more dominant in the presence of low node mobility (v_{max} = 5 m/s).



3 Receivers, v_{max} = 5 m/s 3 Receivers, v_{max} = 25 m/s 3 Receivers, v_{max} = 50 m/s

18 Receivers, v_{max} = 5 m/s 18 Receivers, v_{max} = 25 m/s 18 Receivers v_{max} = 50 m/s

Fig. 5. Average Multicast Tree Lifetime: MLDR vs. MAODV



3 Receivers, v_{max} = 5 m/s 3 Receivers, v_{max} = 25 m/s 3 Receivers, v_{max} = 50 m/s

18 Receivers, v_{max} = 5 m/s 18 Receivers, v_{max} = 25 m/s 18 Receivers v_{max} = 50 m/s

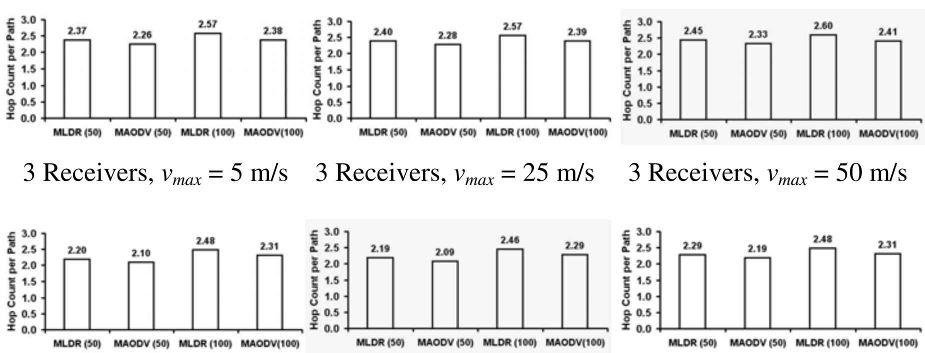
Fig. 6. Average Number of Edges per Multicast Tree: MLDR vs. MAODV

4.2 Number of Edges per Multicast Tree

The tradeoff that we observe for the gain obtained with multicast tree lifetime is a slight increase in the number of edges per multicast tree determined using MLDR compared to that of MAODV. But, the increase is very minimal and when considered over all the simulation conditions, the increase in the number of edges is not beyond 11%. Actually, for low-density networks the difference in the number of edges incurred by the multicast trees of both the protocols is not beyond 7%. For a fixed multicast group size and v_{max} value, both MAODV and MLDR incur lot more edges when operated in networks of high density (100 nodes) compared to that of low density (50 nodes). This can be attributed to the larger connectivity obtained in high-density networks and to connect the multicast source nodes to all the receiver nodes at the maximum percentage of connectivity, more edges are required. For larger multicast group sizes, there is a larger increase in the number of edges per multicast tree with increase in network density. For a given multicast group size, the number of edges incurred by both the MLDR and MAODV multicast trees does not significantly change with node mobility. The increase in the number of edges per multicast tree with increase in the multicast group size is sub-linear and actually the rate of increase in the number of edges gets reduced for larger values of multicast group sizes.

4.3 Hop Count per Source-Receiver Path

The tradeoff that we observe for the gain obtained with multicast tree lifetime is a slight increase in the hop count per source-receiver path on the multicast trees determined using MLDR compared to that of MAODV. But, the increase is very minimal and when considered over all the simulation conditions, the increase in the hop count per source-receiver path is not beyond 8%. Actually, for low-density networks the difference in the hop count per source-receiver path incurred with the multicast trees determined using MLDR and MAODV is not beyond 5%. For a given multicast group size, the hop count per source-receiver path incurred for the multicast trees determined using both MLDR and MAODV does not significantly change with node mobility. For fixed multicast group sizes and maximum node velocity, the hop count per source-receiver paths on the multicast trees slightly increase with increase in network density.



3 Receivers, v_{max} = 5 m/s   3 Receivers, v_{max} = 25 m/s   3 Receivers, v_{max} = 50 m/s

18 Receivers, v_{max} = 5 m/s   18 Receivers, v_{max} = 25 m/s   18 Receivers v_{max} = 50 m/s

Fig. 7. Average Hop Count per Source-Receiver Path: MLDR vs. MAODV

5 Conclusions and Future Work

Stability of the communication structures is critical to reduce the control overhead of the MANET routing protocols. In this pursuit, we have proposed a stable multicast routing protocol based on the Link Distance Ration (MLDR). The LDR of a link is the ratio of the distance between the constituent end nodes of the link and the transmission range per node. MLDR connects the source node to each of the receiver nodes of the multicast group through paths that have the lowest sum of the LDR values of the constituent links. MLDR has been observed to yield a 23-62% longer lifetime than the well-known minimum hop based MAODV routing protocol. At the same time, MLDR does not incur any significantly higher values for the number of edges per tree (at most 11% more edges) and the hop count per source-receiver path (at most larger by 8%), compared to those incurred with MAODV. MLDR requires periodic beacon exchange in the 1-hop neighborhood of nodes and this is a commonly used mechanism in MANETs for nodes to learn about their neighbor nodes. MLDR does not require any additional information to be included in these beacon messages. As part of future work, we intend to study the performance of MLDR, MAODV and some of the other stability-based multicast routing protocols under different mobility models [11] for ad hoc networks.

References

1. Dube, R., Rais, C.D., Wang, K., Tripathi, S.K.: Signal Stability-based Adaptive Routing for Ad hoc Mobile Networks. IEEE Personal Communications, 36–45 (1997)
2. Agarwal, S., Ahuja, A., Singh, J.P., Shorey, R.: Route-lifetime Assessment Based Routing Protocol for Mobile Ad hoc Networks. In: International Conference on Communications, vol. 3, pp. 1697–1701. IEEE, New Orleans (2000)
3. Su, W., Lee, S.-J., Gerla, M.: Mobility Prediction and Routing in Ad hoc Wireless Networks. International Journal of Network Management 11(1), 3–30 (2001)
4. Meghanathan, N., Thomas, D., Addison, E.S.: Multicast Extensions to the Flow-Oriented Routing Protocol and Node Velocity-based Stable Path Routing Protocol for Mobile Ad hoc Networks. In: International Conference on Ultra Modern Telecommunications and Workshops, St. Petersburg, Russia (2009)
5. Mamoun, M.H.: A Novel Technique for the Route Selection in DSR Routing Protocol. International Journal of Video and Image Processing and Network Security 11(3), 1–4 (2011)
6. Jiang, S., He, D., Rao, J.: A Prediction-based Link Availability Estimation for Mobile Ad hoc Networks. IEEE Transactions on Networking 13(6), 1302–1312 (2005)
7. Fall, K., Varadhan, K.: NS Notes and Documentation. The VINT Project at LBL, Xerox PARC, UCB, and USC/ISI (2001), http://www.isi.edu/nsnam/ns
8. Goldsmith, A.: Wireless Communications. Cambridge University Press (2005)
9. Royer, E., Perkins, C.E.: Multicast Operation of the Ad hoc On-demand Distance Vector Routing Protocol. In: The 5th International Conference on Mobile Computing and Networking, pp. 207–218. ACM, Seattle (1999)
10. Bettstetter, C., Hartenstein, H., Perez-Costa, X.: Stochastic Properties of the Random-Way Point Mobility Model. Wireless Networks 10(5), 555–567 (2004)
11. Camp, T., Boleng, J., Davies, V.: A Survey of Mobility Models for Ad hoc Network Research. Wireless Communication and Mobile Computing 2(5), 483–502 (2002)