A Framework for Wireless Ad Hoc Routing Protocols

Tao Lin, Scott F. Midkiff, Jahng S. Park
Bradley Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061 USA
[taolin, midkiff, jahng]@vt.edu

***Abstract*—Many wireless ad hoc routing protocols have been proposed in the literature. Researchers are comparing and improving these protocols before standard routing protocols are defined. In this paper, we present a framework for wireless ad hoc routing protocols based on the concept of a relay node set (RNS). This framework facilitates the comparison, design, and improvement of wireless ad hoc routing protocols. We briefly present an analytical model for comparing protocols using this framework with packet overhead as the metric. We also apply the framework to show how to improve a routing protocol. Future work using the framework is also discussed.**

Keywords—Ad hoc networks, framework, overhead, relay node set, wireless routing protocols

I. INTRODUCTION

A wireless ad hoc network routing protocol is usually classified as a pure proactive protocol, a pure reactive protocol, or a hybrid of the two. A proactive routing protocol periodically maintains routes to all possible destinations, while a reactive protocol builds a route on demand when there is no known route. Many researchers have studied these protocols using simulations of arbitrary networks with certain traffic profiles [1-11]. This past work focuses on comparing existing protocols, designing new protocols, and improving protocols before standard wireless ad hoc routing protocols are defined. A framework to characterize mobile ad hoc network routing protocols can aid these efforts. In this paper, we present such a framework using the concept of a *relay node set* (RNS). The framework characterizes routing protocols using four modules. The framework provides a new view of wireless ad hoc routing protocols and highlights relations among different protocols. The framework provides the following capabilities.

- We can formally describe wireless ad hoc routing protocols with the framework so that researchers can understand the protocols more easily.
- Analytical models drawn from the RNS framework can be used to analyze different wireless ad hoc routing protocols. The framework allows comparison of routing protocols by analytical models coupled with network parameters and traffic profiles. These parameters and profiles could come from simulations or measurements.
- Ideas for possible improvements of proposed wireless ad hoc routing protocols can be found using the RNS framework. The modularized framework allows a subprotocol to be replaced by another subprotocol to form a better wireless ad hoc routing protocol as long as these two subprotocols have the same functionality.
- The RNS framework can aid the design, evaluation and validation of new wireless ad hoc routing protocols.

In this paper, we concentrate on the explanation of this framework and provide examples of the first three capabilities. (Due to page limitations, we do not compare different protocols in this paper.) Section II presents the RNS framework in detail, together with a prototype of an analytical model with emphasis on the control overhead. Using the framework, we give descriptions of the Ad Hoc On-Demand Distance Vector (AODV) [12], Optimized Link State Routing (OLSR) [13], and Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) [14] routing protocols as examples in Section III. Section III also illustrates use of the protocol for analysis. The final section gives a summary and describes potential future work.

II. RELAY NODE SET FRAMEWORK

We define two terms before we present the framework: *cover* and *relay node set* (RNS). When a node broadcasts, all of its neighbors should be able to receive that message. (In this paper, we assume that the medium access control layer protocol can guarantee delivery.) Referring to this property, we say that the sender *covers* all of its neighbors. A set of nodes, say set *M*, is covered by another set of nodes, say *N*, when any node in *M* is covered by at least one node in *N*. Fig. 1 shows an example. Node 2 covers nodes 1, 3, and 5. Also, the set of black nodes covers the set of white nodes. All nodes in a wireless ad hoc network act as routers and are willing to forward data packets received from their neighbors. The relay node set is a set of nodes that have the capability to retransmit control messages in the wireless ad hoc network. Nodes that are not in an RNS, called non-RNS nodes, always stay silent

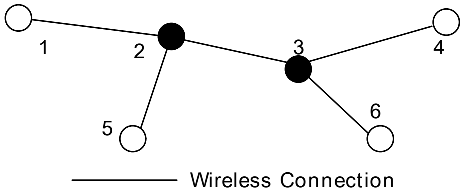

Figure 1. Example of a relay node set and for the definition of cover.

This research is supported in part by the Office of Naval Research through the Navy Collaborative Integrated Information Technology Initiative (NAVCIITI).

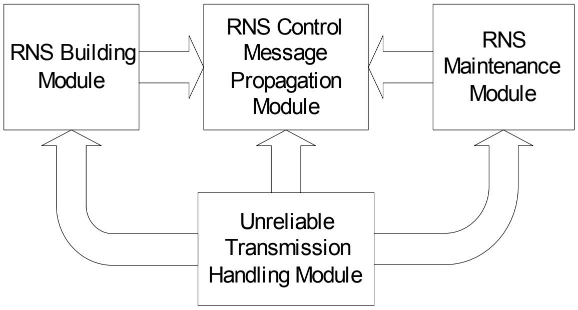

Figure 2. The four modules in the RNS framework.

after they have processed control messages broadcast by their neighbors. Fig. 1 also illustrates the role of an RNS. For example, assume that the control message is a list of neighbors for each node. Every node broadcasts to nodes within range. We also assume in this figure that black nodes form a RNS. According to the definition, only these two nodes, nodes 2 and 3, retransmit the first received control messages from their neighbors. In this example, all nodes are able to receive neighbor node lists from others. Thus, all nodes can build routing tables according to the topology indicated by neighbor lists.

The framework presented in this section is built using the concept of the RNS. It can describe a broad class of wireless ad hoc routing protocols. As shown in Fig. 2, this framework contains four modules: RNS building, RNS control message (unicast or broadcast) propagation, RNS maintenance used when the topology changes, and unreliable (control message) transmission handling. (In this paper, we only consider the basic routing function for these protocols, not other issues such as quality of service and security.) The arrows in Fig. 2 imply the dependency relationships between pairs of modules. For example, the arrow from the module that handles unreliable transmissions to the module that handles the building of the RNS implies that the latter relies on the service of or information from the former.

Different schemes used in these modules result in different wireless ad hoc routing protocols, including both reactive and proactive routing protocols. In the RNS building module, different routing protocols use different methods to build the RNSs and the result could be one RNS for the entire network or multiple copies of RNSs for the network at a given time. An RNS can be associated with a certain node, a certain link, or a certain pair of nodes when multiple copies of RNSs are allowed to exist in a network at a given time. An RNS may or may not cover all nodes in the network, depending on the algorithm that builds it. For example, AODV [12] builds an RNS when an initiator starts the route request procedure by broadcasting a request message. This RNS is associated with a pair of source and destination nodes and may not cover all nodes in the network. This is a typical feature of reactive routing protocols. RNSs built by proactive routing protocols usually cover the entire network. Routes can be built based on the RNSs. For example, in AODV, a route request reply message is sent back to the initiator by unicast. Thus, all intermediate nodes and the initiator can build the routing entry for that particular destination.

Control messages sent in an RNS are different for different routing protocols. RNS nodes may unicast or broadcast control messages. A control message can be a list of visited nodes for the control message, a request reply from a destination to the corresponding initiator, default gateway information for each node, or a link state record. These control messages are used to build routing entries. Generally speaking, the quality of the routing information is proportional to the size of the control message. And, exchanging more information leads to more overhead.

In general, reactive routing protocols do not have the capability to maintain RNSs. Once an old path is broken due to the movement of intermediate nodes, the RNS building procedure starts again and new routing paths generated by the new RNSs replace the old ones. Proactive routing protocols usually have some special schemes to maintain RNSs when an old one becomes invalid. (Examples are given in Section III.) Therefore, the frequency of building RNSs in reactive routing protocols is higher than in proactive protocols for a particular environment. We can assume that the total rate of RNS building and maintenance functions in a certain network environment for any protocol is a constant. This is because when the topology changes, protocols will either maintain old RNSs or rebuild new RNSs.

In a wireless network, a transmission may fail due to the poor quality of the wireless channel. Control messages are damaged or lost due to transmission errors or buffer overflows. Therefore, a routing protocol usually has a separate module to handle unreliable transmissions. If the medium access control (MAC) protocol can guarantee delivery, e.g., as in IEEE 802.11 [15], we can ignore this module.

Our framework can describe a broad class of routing protocols for wireless ad hoc networks. It also illustrates some internal similarities among protocols. We are able to analyze protocols with this framework. For example, according to the four-module RNS framework, the total control overhead for a wireless ad hoc routing protocol is formed by four overhead components. They are the overhead to build or rebuild the RNS, the overhead to maintain the RNS, the overhead to propagate control messages in the RNS, and the overhead to handle unreliable transmissions. Therefore, we can develop an analytical model for overhead. Equation (1) represents the total control overhead for a wireless ad hoc routing protocol in a certain amount of time t.

$$\begin{aligned} Overhead &= O_{const} + O_{maint} + O_{prop} + O_{retrans} \\ O_{const} &= \sum N_{const} \sum S_{RNS} P_{control\_const} \\ O_{maint} &= \sum N_{maint} \sum S_{maint} P_{control\_maint} \\ O_{prop} &= \sum N_{prop} \sum S_{prop} P_{control\_prop} \\ O_{retrans} &= \sum N_{retrans} \sum S_{retrans} P_{control\_retrans} \end{aligned} \tag{1}$$

Here O stands for the overhead, P represents the control packet sizes, S represents the size of the relay node set, N is the number of control operations that occur in time period t, and the subscripts indicate the corresponding modules associated with the variables. Note that the node sets in maintenance, propagation, and retransmission modules are a subset of the corresponding RNSs built in the construction module. In (1), the overhead for the RNS building module and the RNS control message propagation module can be considered as the static overhead of a routing protocol. The overhead for the maintenance module is due to the dynamic changes in the network topology. The overhead for the retransmission module provides robustness for a routing protocol. Generally speaking, if one of the terms becomes smaller, some of the others may become larger. Proactive routing protocols have large overheads in the RNS building module and the RNS control message propagation module when compared to reactive protocols. Proactive protocols have small overhead to maintain the RNS so that the number of rebuilding operations is small. Protocol developers must balance these factors for a given environment so that routing protocols have optimal overall overhead. Descriptions of these routing protocols with this analytical model can help to identify overhead factors. Comparison of results among protocols in this framework can be used to choose, improve, and design wireless ad hoc routing protocols. To summarize, the overhead model based on the RNS framework can be used to define multi-objective optimization problems for MANET routing protocols. In the next section, we show simple examples of analysis of typical routing protocols using the framework.

III. WIRELESS AD HOC ROUTING PROTOCOLS IN THE RNS FRAMEWORK

We present application examples of the RNS framework in this section. Due to space limitations, we only briefly describe three routing protocols, AODV [12], OLSR [13], and TBRPF [14]. We also show how to improve a protocol through an example using OLSR.

A. *Description of AODV*

AODV is a typical on-demand reactive routing protocol. In AODV, a route request is sent when there is no known route to the expected destination. A node rebroadcasts the route request if it does not know the route or it is not the destination. The nodes that rebroadcast the route request form the RNS set associated with a particular pair of source and destination nodes. This procedure is the RNS building procedure in AODV, which stops when a route is found. An RNS in AODV usually does not include all nodes in the network, except in the case that the route does not exist or the route is discovered when the destination joins the RNS as the last node in the network. AODV only unicasts replies to the corresponding route requests via the RNS it just built. This is the propagation of control message module in AODV. Routes could fail due to changes in the network's topology. AODV does not have any scheme to maintain RNSs. Therefore, a route request must be re-sent when a known route fails. This implies that AODV rebuilds the RNS for a pair of source and destination nodes when the old one fails. If the underlying MAC protocol cannot guarantee delivery, AODV may send multiple transmissions or use acknowledgments to correct for missing control packets, e.g., due to buffer overflow at the receiver or corruption during transmission. This is the module to handle unreliable transmissions.

Equation (2) gives the control overhead in AODV.

$$\begin{aligned} Overhead &= O_{const} + O_{prop} + O_{retrans} \\ O_{const} &= \sum_{i=1}^{N_{Pair}} \sum_{j=1}^{S_{i,RNS}} P_{i,j,request} \\ O_{prop} &= \sum_{i=1}^{N_{Pair}} \sum_{j=1}^{S_{i,reply}} P_{i,j,reply} \\ O_{retrans} &= \sum_{i=1}^{N_{Pair}} \sum_{j=1}^{S_{i,retrans}} P_{i,j,retrans} \end{aligned} \tag{2}$$

The variables used in (2) are defined below. We assume that AODV uses notifications from layer 2 to detect link up and link down events.

- N_{pair} is the total number of pairs of source and destination nodes.
- $S_{i,RNS}$ is the size of the corresponding RNS for the *i*th pair of source and destination nodes.
- $P_{i,j,request}$ is the size of the *i*th route request packet forwarded by node *j*. This is a constant value since the size of the number of hops field in the route request is fixed.
- $S_{i,reply}$ is the size of the subset of the corresponding RNS for the *i*th pair of source and destination nodes that forwards the route reply message.
- $P_{i,j,reply}$ is the size of the *i*th route reply packet sent back to the initiator by node *j*. This is also a constant value.
- $S_{i,retrans}$ is the size of the subset of the corresponding RNS for the *i*th pair of source and destination nodes that needs to retransmit the control messages. It can be zero if the underlying MAC layer can guarantee delivery.
- $P_{i,jretrans}$ is the size of the *i*th route request or reply packet retransmitted by node *j*. This is a constant value.

To summarize, as a typical on-demand reactive protocol, AODV does not have any control overhead in the RNS maintenance module. It may have a small amount of overhead in the propagation module. The overhead in the building RNSs module and handling unreliable transmissions module depends on the network profile and traffic profile used in the applicable network environment. In other words, it may suffer a large overhead penalty to build or rebuild RNSs in a high-mobility network. It also requires significant overhead to retransmit packets in a poor communication environment.

B. *Description of OLSR*

OLSR is a typical proactive routing protocol. OLSR uses "hello" messages to exchange neighbor lists between neighboring nodes. By using the neighbor list information, multiple relay (MPR) node sets are built for all nodes with a simple algorithm. An MPR node set is a small subset of neighboring nodes that covers all of two-hop neighbors. This is the RNS building procedure in OLSR. RNSs in OLSR are associated with certain source nodes. The RNS for a source node is the MPR set of that node, combined with the MPR sets of nodes in the MPR set of that source node, and so on until this RNS covers all the nodes in the network. Therefore, multiple copies of RNSs exist in OLSR at any given time. Information about MPR sets is also sent to neighbors via the "hello" messages. All nodes generate their own MPR selector (MPRS) sets. The MPRS for a node is the set of neighboring nodes that select this node as a member of their MPRs. Nodes with non-empty MPRSs broadcast the MPRSs in the corresponding RNSs. So, the number of RNSs is equal to the number of nodes with non-empty MPRS in the network. Each node rebroadcasts control messages sent by nodes that are in its MPRS set. Therefore, information about all MPRS sets can be propagated to all nodes in the network with a small number of retransmissions. Routes can be built based on the information about MPRS sets of other nodes. MPRS nodes act as gateways and pairs of MPR and MPRS nodes form routes. This is the RNS propagation procedure for OLSR. When the topology changes, OLSR uses local modifications of the MPR and MPRS sets to maintain RNSs at nodes within a two-hop range of that changed link. This is the RNS maintenance module for OLSR. OLSR periodically broadcasts control messages. Therefore, it has the capability to handle unreliable transmissions.

The total overhead in OLSR is shown in (3). Note that since OLSR periodically broadcasts control messages, we can assume that the module to handle unreliable transmissions is included in the propagation module.

$$\begin{aligned}
Overhead &= O_{const} + O_{maint} + O_{prop} \\
O_{const} &= \sum_{i=1}^{N_{hello}} \sum_{j=1}^{N} P_{i,j,hello} \\
O_{maint} &= \sum_{i=1}^{N_m} \sum_{j=1}^{N_{i,adjust}} (P_{i,j,MPRS} S_{i,j,RNS}) \\
O_{prop} &= \sum_{i=1}^{N_{update}} \sum_{j=1}^{N_{i,MPRS}} (P_{i,j,MPRS} S_{i,j,RNS})
\end{aligned} \tag{3}$$

The variables are defined below.

- N_{hello} is the total number of periods in which hello messages are sent.
- N is the total number of nodes in the network.
- $P_{i,j,hello}$ is the size of the ith hello packet sent by node j. The actual overhead in the RNS building phase can vary since the size of the hello messages varies.
- N_m is the number of RNS maintenance operations due to link changes. The RNS maintenance operation is invoked each time the topology changes. Thus, N_m depends on random changes in the topology.
- $N_{i,adjust}$ is the number of nodes that need to adjust MPRs in the ith maintenance operation. When the topology changes, the MPRSs of some nodes within a two-hop range may change. This parameter depends on the exact MPR selection algorithm that is used and on the network profile.
- $P_{i,j,MPRS}$ is the size of the MPRS declaration packet (topology control message) for node j in the nodes that adjust their MPRS sets when the ith link changes. The size of the MPRS declaration packet may vary, so the actual overhead due to MPRS declaration varies.
- $S_{i,j,RNS}$ is the size of the RNS for node j in the ith maintenance operation.
- N_{update} is the total number of periods in which broadcast topology control messages are sent.
- $N_{i,MPRS}$ is the number of nodes with a non-empty MPRS at the ith periodic broadcast.

Parameters in OLSR's overhead are independent of traffic profiles. This is a common feature for proactive routing protocols. They can have a fixed upper bound on overhead in a network with any type of traffic. This can be an advantage for proactive protocols.

C. Description of TBRPF

TBRPF is another promising proactive protocol. Each node uses "hello" messages to detect links to its neighbors. Based on the local link state database, each node first builds a shortest path tree to all possible destinations. A node decides whether or not to report links in its shortest path tree to its neighbors by an estimation algorithm based on its local link state database. Information that is shared with neighbor is considered to be reportable. Basically, a neighbor node is added to a reportable node set if this node has at least one neighbor that is not connected to this neighbor. Links in the shortest path tree are added if one end point is in the reportable node set. Therefore, selected links in the shortest path tree form a reportable tree. Each node broadcasts its reportable tree. For any link, there is a set of nodes that broadcasts that link to their neighbors. Therefore, an RNS is built for each link in TBRPF. The construction of reportable trees for all nodes is the RNS building module in TBRPF. The control messages sent in TBRPF are reportable trees. The protocol guarantees that all nodes have enough information to build proper shortest path trees based on reportable trees from neighbors. When the topology changes, the maintenance module uses online computation to update corresponding RNSs and the link state update propagates to all related nodes in the associated RNSs. Similar to OLSR, TBRPF uses periodic broadcast messages to handle unreliable transmissions. The overhead for TBRPF is presented in (4).

$$\begin{aligned}
Overhead &= O_{const} + O_{maint} + O_{prop} \\
O_{const} &= \sum_{i=1}^{N_{hello}} \sum_{j=1}^{N} P_{i,j,hello} \\
O_{maint} &= \sum_{i=1}^{N_m} \sum_{j=1}^{S_{i,RNS}} P_{i,j,LS} \\
O_{prop} &= \sum_{i=1}^{N_{update}} \sum_{j=1}^{E_i} \sum_{k=1}^{S_{i,j,RNS}} P_{i,j,k,LS}
\end{aligned} \tag{4}$$

Some of these variables are the same as those in (3). Others are defined as follows.

- $S_{i,RNS}$ is the size of the RNS for link i.
- $P_{i,j,LS}$ is the size of a link state description for link i at node j.
- E_i is the number of edges in the ith periodic broadcast in TBRPF.
- $S_{i,j,RNS}$ is the size of the RNS for link j.
- $P_{i,j,k,LS}$ is the size of a link state description for link j at node k in the ith periodic broadcast.

D. Example Application: Improving OLSR

We now show an example of using the analytical model to improve OLSR. Equation (3) shows that an algorithm with smaller average RNS size should have less overhead for OLSR. This observation is mentioned in the OLSR draft [13]. Here we propose a modified OLSR that can build one copy of the connected dominating set as the RNS for the network. It uses similar "hello" messages to detect link status and exchange neighbor information. Two-hop neighbor information is used to select the connected dominating set in the first phase. When a node i has a neighboring node that is not covered by the sender node k, node i checks whether there is a common neighbor between i and j, say node x, that covers that neighbor. Node i rebroadcasts the message if node x does not exist or if x has a smaller degree. In the case of a tie, the node with the smallest ID wins. If a node is selected as a connected dominating node in the first phase, it checks its connected dominating neighbors with larger IDs. If those nodes are connected and cover all of its other neighbors, this node is removed from the connected dominating set.

We used simulation to determine the parameters for the size of the RNSs for the original OLSR and the modified OLSR. Since dynamic topologies can be considered as a sequence of static topologies, we implemented these protocols in C++ with sequences of static topologies. Simulations were done in a 100×100 unit square map. Two nodes are assumed to be able to communicate with each other if the distance between them is less than the given maximum radio range. Three maximum radio range values, 25, 50, and 75 units, were used. The number of nodes ranged from 2 to 100. OLSR and modified OLSR were simulated. The latest OLSR draft states that a node "should select an MPR set such that any two-hop neighbor is covered by at least MPR_COVERAGE MPR nodes" [13]. We assume that the minimum MPR set is used in OLSR, i.e., MPR_COVERAGE equals 1. We generated 1000 randomly connected topologies for each set of parameters and obtained the average size of the RNSs. Simulation results in Fig. 3 show that the average RNS sizes in this modified OLSR are smaller than the average RNS size in OLSR. Therefore, it should reduce overhead compared to the algorithm used by OLSR.

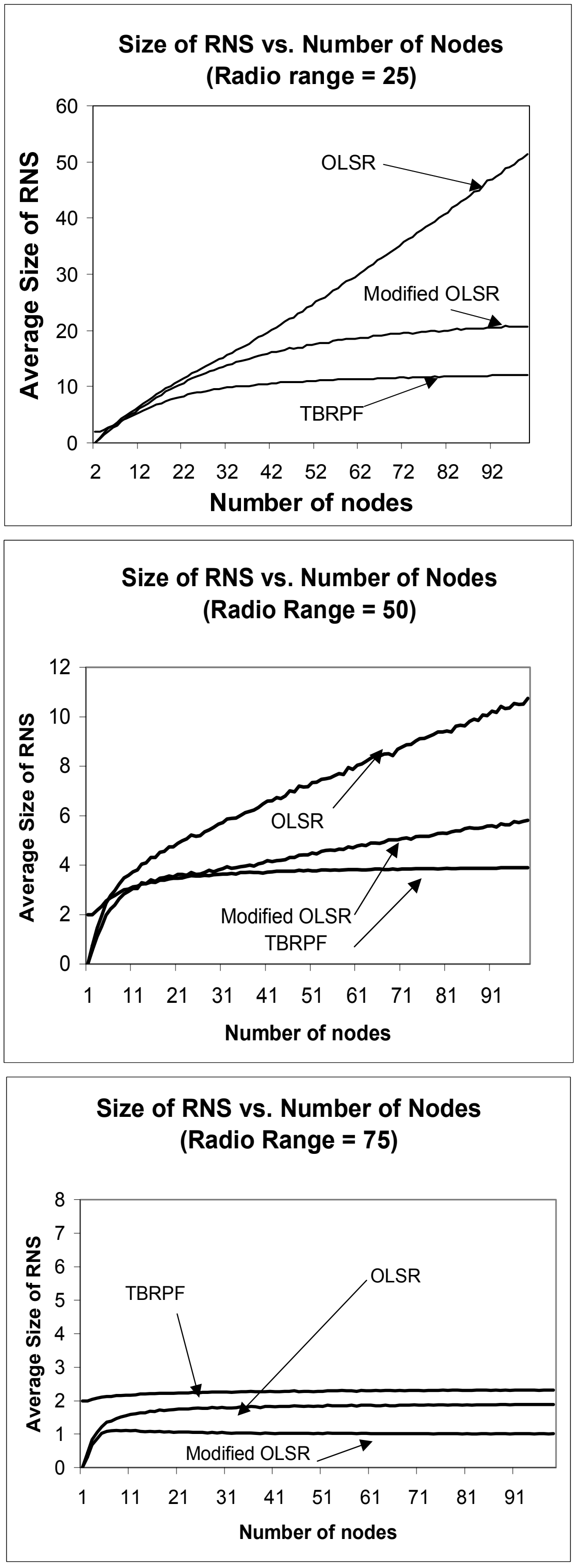

Figure 3. Average RNS size for OLSR, modified OLSR, and TBRPF.

Similar simulations are used to compute the average size of the RNS for TBRPF. The results are also presented in Fig. 3. Note that the RNS for TBRPF is associated with links. Therefore, without the applicable network profiles, we cannot simply compare the average RNS size for TBRPF with the one for OLSR. Due to the page limitation, we are not going to compare AODV, OLSR, and TBRPF with different assumptions of network and traffic profiles.

IV. CONCLUSIONS AND FUTURE WORK

We presented a framework based on the concept of a Relay Node Set that can characterize wireless ad hoc routing protocols. We developed a prototype of an analytical model based on the RNS framework for control overhead for wireless ad hoc routing protocols. A simple example was illustrated in this paper showing how to improve routing protocols with the RNS framework.

Potential future work includes describing a broader class of routing protocols with this framework and analyzing the associated control overhead, extending the analytical models to other metrics, e.g., power consumption, and developing new wireless ad hoc routing protocols. Furthermore, we can generalize the framework by including additional factors that more accurately represent the real operating environment.

ACKNOWLEDGMENT

The authors wish to thank Luiz DaSilva, Nathaniel Davis IV, Michael Christman, George Hadjichristofi, Kaustubh Phanse, and John Wells of Virginia Tech for their contributions to this research.

REFERENCES

[1] C. Ying, Q. Lv, Y. Liu, M. Shi, "Routing protocols overview and design issues for self-organized network," In *Proc. of International Conference on Communication Technology (WCC - ICCT 2000)*, Vol. 2, 2000, pp. 1298–1303.

[2] E.M. Royer, C.K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Communications*, vol. 6, no. 2, Apr. 1999, pp. 46–55.

[3] S.R. Das, C.E. Perkins, E.M. Royer, "Performance comparison of two on-demand routing protocols for ad hoc networks," In *Proc. 19th* INFOCOM, vol. 1, 2000, pp. 3–12.

[4] C.E. Perkins, E.M. Royer, S.R. Das, M.K. Marina, **"**Performance comparison of two on-demand routing protocols for ad hoc networks," *IEEE Personal Communications*, vol. 8, no. 1, Feb. 2001, pp. 16–28.

[5] A. Boukerche, "A simulation based study of on-demand routing protocols for ad hoc wireless networks," *Proc. 34th Annual Simulation Symposium,* 2001, pp. 85–92.

[6] T. Ogawa, E. Kudoh, H. Suda, "Multi-routing schemes for ad-hoc wireless networks," *Proc. IEEE International Symposium on Circuits and Systems,* vol. 4, 2001, pp. 866–869.

[7] A. Boukerche, "Performance comparison and analysis of ad hoc routing algorithms," *Proc. IEEE International Conference on Performance, Computing, and Communications,* 2001, pp. 171–178.

[8] J. Raju, J.J. Garcia-Luna-Aceves, "A comparison of on-demand and table driven routing for ad-hoc wireless networks," *IEEE International Conference on Communications*, vol. 3, 2000, pp. 1702–1706.

[9] M.E. Steenstrup, "Routing under uncertainty: A comparative study," *Proc. IEEE Wireless Communications and Networking Conference*, vol. 1, 2000, pp. 112–116.

[10] I. D. Aron and S. K. S. Gupta. "On the Scalability of On-Demand Routing Protocols for Mobile ad hoc Networks: An Analytical Study," *Journal of Interconnection Networks,* vol. 2, no. 2, 2001, pp. 5–29.

[11] V.D. Park and M.S. Corson, "A performance comparison of the temporally-ordered routing algorithm and ideal link-state routing," *Proc. 3rd IEEE Symposium on Computers and Communications, 1998*, pp. 592–598.

[12] C. E. Perkins, E. M. Belding-Royer, and Samir R. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," Internet Engineering Task Force (IETF) draft, November 2002. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-12.txt

[13] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot, "Optimized Link State Routing Protocol," Internet Engineering Task Force (IETF) draft, Mar. 2002. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-manet-olsr-06.txt.

[14] R. G. Ogier, M. G. Lewis, F. L. Templin, and B. Bellur, "Topology Dissemination Based on Reserve-Path Forwarding," Internet Engineering Task Force (IETF) draft, November 2002. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-manet-tbrpf-06.txt.

[15] IEEE Computer Society LAN MAN Standards Committee, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std. 802.11-1997*, The Institute of Electrical and Electronics Engineers, New York, New York, 1997.