Providing a Bidirectional Abstraction for Unidirectional Ad Hoc Networks

Venugopalan Ramasubramanian Ranveer Chandra
Department of Computer Science
Cornell University
Ithaca, NY 14850

Daniel Mossé
Department of Computer Science
University of Pittsburgh
Pittsburgh, PA

Abstract— **Several routing protocols for mobile ad hoc networks work efficiently only in bidirectional networks. Unidirectional links may exist in a real network due to variation in transmission power of different nodes, noise or other signal propagation phenomena, and heterogeneity in transmission hardware of nodes in the network. We introduce a sub-layer called Sub Routing Layer, SRL, between the network and the MAC layer to provide a bidirectional abstraction of the unidirectional network to the routing protocols. We present a scalable and efficient way to provide this abstraction by finding and maintaining multi-hop reverse routes to each unidirectional link. We simulate SRL and a modified version of AODV that uses SRL to route packets in unidirectional network. We observed that with SRL, the packet delivery of AODV in unidirectional networks increases substantially. Further our simulations indicate that reverse routes are often only a few hops long and hence the overhead of using SRL is very low.**

I. Introduction

A network of mobile nodes using peer-to-peer communication is called an ad hoc network. The nodes in an ad hoc network are limited by power, memory, bandwidth and computational constraints. Such networks have the ability to provide cheap communication without any fixed infrastructure. Hence, they are very useful in disaster recovery, collaborative computing, rescue operations and military surveillance.

Several routing algorithms have been designed to work efficiently in bidirectional networks. However, unidirectional links may be present in a real network due to a number of reasons including diversity in the transmission power of nodes, noise and collisions affecting packet reception at some regions, heterogeneity in the radio hardware of mobile nodes. Presence of unidirectional links renders some of these algorithms such as TORA [7] inoperable while many others including AODV [1] continue to operate, routing only along the bidirectional links. Few protocols such as DSR [5] can also route packets through unidirectional links but face many problems that decrease their efficiency. In this paper, we present an efficient algorithm to enable routing protocols that were primarily designed for bidirectional networks to work with unidirectional links. We propose to introduce a sub-layer between the network layer and MAC layer called Sub Routing Layer (SRL). The Sub Routing Layer presents a bidirectional abstraction of the network to the routing layer.

In Section II we provide the motivation for our work. SRL is described in detail in Section III, its performance measures are presented in Section IV and V, and in Section VI we present a framework to adapt on-demand routing protocols to operate over SRL and discuss performance measures of AODV over SRL.

The authors were 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 the Intel Corporation

Also supported in part by DARPA PARTS (Power-Aware Real-Time Systems) project under Contract F33615-00-C-1736 and through NSH-ITR medium grant.

II. Motivation and Related Work

Power is an extremely critical resource in mobile networks. Each node has a limited battery life which typically allows for a few hours of continuous operation. Since these small devices are often released in very harsh environments, an efficient utilization of limited power supply is necessary. Turning radios on and off provide major power savings, but varying the transmission power (and thus the range of the node transmissions) is a powerful and well-explored technique [12].

An enabling technology is the ability of wireless network cards to transmit at different power levels. Hence, allowing these devices to transmit at lower power levels would help to prolong their lifetime. In most foreseeable environments, handheld devices, laptops running on battery power, laptop hooked to a power supply, base station transmitters are all interacting devices with inherently different power supplies. It is advantageous for these devices to operate at their own optimal power for communication. Further, it may be advantageous to allow these devices to lower their transmission power in a very dense environment to decrease congestion and to increase their transmission power in a sparse environment to increase connectivity

A fundamental problem of allowing nodes to transmit with different transmission powers is the creation of unidirectional links in the network. For example, in Figure 1 node A is transmitting at higher power than other nodes. Node B is within the transmission range of node A while node A cannot hear node B. Thus the link $A \rightarrow B$ is unidirectional. Unidirectional links may also arise when nodes are transmitting at the same power. Increased collisions or noise could affect reception at one node more than the other. However, the nature of unidirectional links created due to the diversity in transmission powers is different from the nature of unidirectional links created by noise, collisions and other propagation phenomena. Unidirectional links created by noise related phenomena are local and temporary and hence may not affect routing protocols to a great extent. Further, the reverse routes back to the source of the unidirectional link tend to get bigger with increased diversity in transmission powers.

Several of the existing routing protocols (e.g., AODV [2] and TORA [7]) were primarily designed only for bidirectional networks. While a few protocols (e.g., DSR [5] and ZRP [8]) have the capability to route packets using unidirectional links many others route packets only along the bidirectional links. Several problems in routing with unidirectional links are examined in

[9]. Below we briefly discuss some of the problems faced by a few routing protocols in this environment.

Ad-hoc On-demand Distanve Vector routing (AODV) [1] is an on-demand routing protocol that has been designed for bidirectional networks. AODV broadcasts route requests (RREQs) and uses the bidirectional links (reverse routes) to transmit route replies (RREPs). In fact, since AODV is unaware of uni- or bidirectionality of the links, it will fail if there are only unidirectional links between a source and a destination. In other words, Reverse routes may not exist in the presence of unidirectional links. Thus, identifying unidirectional links is an important issue for AODV; towards that end, it was recently suggested to maintain a list of nodes called *black-list* to identify unidirectional links. But the black-list only provides an approximate identification of unidirectional links.

Dynamic Source Routing (DSR) [5] is a purely reactive routing protocol that is designed to work even in unidirectional networks. Routes are discovered by broadcasting route requests and route replies. Data packets are then source-routed to the destination. In bidirectional networks several optimizations can be applied to DSR and the control overhead can be reduced significantly (e.g., instead of discovering the route from destination to source, simply follow the reverse route through bidirectional links for replies). However, the performance of DSR in unidirectional networks is limited by the scalability of the protocol. Since intermediate nodes with cached routes to the destination will broadcast route replies back to the source, several nodes might respond to the same route request thus resulting in a large number of route reply broadcasts. We call this the *RREP-explosion* problem. Further, DSR relies on hop level acknowledgments in unidirectional networks for discovering route errors. This requires the discovery and maintenance of additional reverse routes for acknowledgments at every hop. This extra control overhead results in increased congestion and severely limit the throughput of the network.

Zone Routing Protocol (ZRP) [8] is a routing protocol that exploits the effectiveness of both proactive and reactive routing strategies. Each node maintains information about the topology in a small area around it called *zone*. Routing within a zone is proactive (i.e., routes are maintained for all pairs within the zone at all times) while inter-zone routing is done using a reactive protocol. A technique called *border-casting* is used to minimize the broadcast overhead of inter-zone route requests; only nodes at the border of the zone will forward/rebroadcast the RREQ. In [11], ZRP is modified to route in the presence of unidirectional links. Intra-zone routing is used to transmit route replies and route errors thus avoiding the RREP-explosion problem. Reverse routes in this protocol are gathered from the periodic packets broadcast throughout the zone. SRL is also locally proactive, maintaining proactive routes in a small locality. Compared to unidirectional ZRP, SRL's proactive zone is of a much smaller size. Additional optimizations in SRL's proactive routing algorithm helps it to have a considerably lower periodic control overhead.

An alternative approach, based on tunneling, to handle the problem faced by unidirectional links is presented in [10]. Control packets and acknowledgments are tunneled back to the source enabling the routing protocols to use unidirectional links avoiding loops and explosion of acknowledgment packets. Periodic packets containing a list of neighbors are broadcast to help with the task of identifying unidirectional links. Reverse routes needed for tunneling packets are gathered and maintained by the same routing protocol as used for routing data packets. This may lead to inefficiencies because the routing protocol may not be well suited to maintaining several short reverse routes in unidirectional networks. Also existence of unidirectional links may not be known at the source if the network is not strongly connected.

The Internet MANET Encapsulation Protocol (IMEP) [13] also operates between the link layer and network layer providing several services including link-status sensing, broadcast reliability, and control message aggregation. IMEP can detect and monitor the occurences of unidirectional links in the network. However, unlike SRL, IMEP does not attempt to maintain reverse routes and facilitate use of unidirectional links for data traffic. Temporally Ordered Routing Algorithm (TORA) [7] is a routing protocol that employs a link-reversal algorithm to proactively maintain routes to nodes in an ad hoc network. TORA utilizes the services provided by IMEP to route packets along the bidirectional links of the network. TORA's routing algorithm breaks down in the presence of unidirectional links.

Working with unidirectional links is also problematic due to the lack of efficient MAC level protocols that work with unidirectional links. The failure of both RTS-CTS and ACK based schemes make MAC protocols inefficient in a unidirectional environment. Further, some of the services provided by MAC protocols such as detection of link breaks and neighbor discovery may no longer be available. Thus, there is a need to compensate for these drawbacks and make routing protocols adapt efficiently to unidirectional environments.

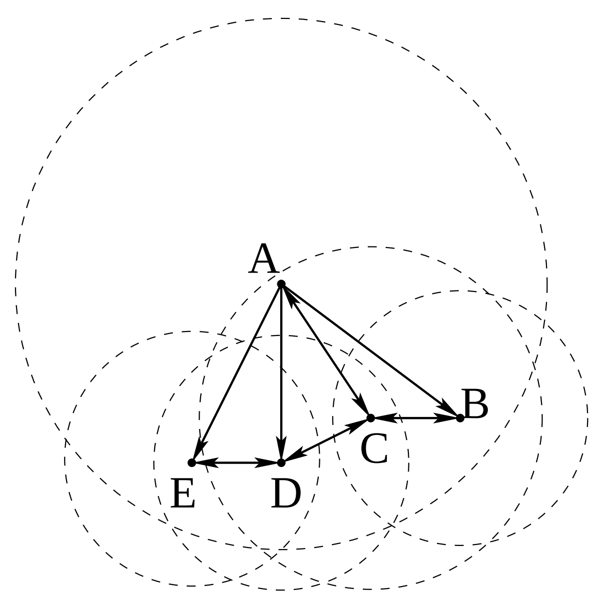

Fig. 1. A network with unidirectional links

III. Sub Routing Layer (SRL)

In the previous section, we described the problems that arise when nodes are allowed to transmit at different power levels. In this section, we present a solution to address the unidirectionality problem. We observed earlier that, in Figure 1, node A transmitting at a power much higher than node B could create a situation where node B can receive packets directly from node A but node A cannot receive packets from node B. However, it is possible that an intermediate node C can receive packets from B as well as directly send packets to A. Thus, B, C and A form a reverse route for packets from B to reach A. Protocols such as AODV that utilize bidirectional links to send route replies and route maintenance packets can now route data packets through link $A \rightarrow B$ while using the reverse route $B \rightarrow C \rightarrow A$ to send route replies or route error packets. Sometimes we may need to traverse more than one intermediate node to reach A. For example, in Figure 1, $E \rightarrow D \rightarrow C \rightarrow A$ forms a 3-hop reverse route from E to A.

Thus, having a reverse route, even if it is few hops long, can make a unidirectional network easily operable. While it is true that the reverse route may be longer (in number of hops and delay) than the forward route, only control packets (e.g., route errors and route replies) are routed through them. This increases only the route discovery latency. In practice, we expect that the length of the reverse route would be a small number. Thus, by incurring a little extra latency for the route replies and route maintenance packets, we can avoid network-wide broadcasts of route replies and acknowledgments for each packet routed. An efficient way to gather this reverse route information is detailed in Section III-B.

Since the reverse route information is a generic information useful to several protocols, we present a uniform abstraction that can be used by several routing protocols either to successfully operate in this unidirectional environment or to optimize its performance specific to this environment. We have designed a sublayer called SRL (Sub Routing Layer) to provide this abstraction. SRL is not strictly a sub layer, that is protocols can bypass SRL and send packets directly to the MAC layer. The function of SRL is to provide certain services and interfaces based on reverse route information for routing protocols to function efficiently in a unidirectional ad-hoc network.

SRL works by broadcasting (more precisely neighborcasting) to its neighbors periodic update messages about the connectivity within a small local region of the network. In addition, periodic hello messages are used to detect link status. The number of update messages sent is of the same order as the number of hello messages sent by the neighbor discovery mechanism. Since the update messages can also play the role of hello messages, the update messages do not increase the number of periodic packets neighbor-cast (broadcast with TTL 1). The update messages differ from hello messages because they contain some extra information about the connectivity of the node sending the update message. Our algorithm explained in the next section attempts to restrict the average size of extra information to be small and at most of the order of the number of nodes in the local region. In our scheme, the radius of the local region is chosen to be of the order of the number of hops required to establish reverse routes. The diversity in the transmission power of different nodes and the density of nodes in the network determine the average number of hops in the reverse route.

A. Nomenclature

Before proceeding with the algorithm, we define the terminology used in this paper. Whenever a node B is in the transmitting range of node A, a link exists from A to B and it will be indicated as $A \rightarrow B$. A link is said to be bidirectional if both $A \rightarrow B$ and $B \rightarrow A$ hold. Whenever $A \rightarrow B$, the node A is said to be an *in-neighbor* of node B and B is said to be an *out-neighbor* of node A. The notation $A \rightsquigarrow_n B$ is used to indicate that a path of length n hops exists from A to B in the unidirectional network. For a link $A \rightarrow B$, the shortest route through which B can reach its in-neighbor A is called the *reverse route*. For example, in Figure 1 the path $B \rightarrow C \rightarrow A$ forms the reverse route for the link $A \rightarrow B$. If a link is bidirectional then the reverse route is of length 1.

The *locality radius* sets the size of the local region called *locality* about which information is collected. For a given node A, every node that can reach A by using a path of length equal to or less than the locality radius would be included in the locality of A. In other words, $locality(A) = \{X | X \rightsquigarrow_i A \text{ and } 1 \leq i \leq r\}$, where r is the locality radius. SRL attempts to find reverse routes only within the locality. As a result, the locality radius also bounds the maximum length of a reverse route found. For example, locality radius 1 means the reverse routes found are all of one-hop length, i.e., only bidirectional links are utilized.

B. Reverse Distributed Bellman-Ford Algorithm

This section describes the algorithm used by SRL to gather reverse routes efficiently. We have already mentioned that the SRL works by sending periodic update messages with local connectivity information. The efficiency of this method depends on the amount of information that is broadcast. An easy way to gather the local connectivity information is to broadcast, at each node, the link status of all nodes within distance of locality radius. With a locality radius of 1, this would be equivalent to broadcasting the list of in-neighbors. However, the amount of information thus broadcast at each node is of the order of square of the number of nodes in the local region. Even with a small local radius and moderate node density, the size of the update message could be very high. In our design, we reduce the information broadcast to the order of the number of nodes in the local region. We modified the well-known Distributed Bellman-Ford algorithm to achieve a low complexity in information gathering.

Distributed Bellman-Ford algorithm is a well-known algorithm to obtain the shortest routes between pairs of nodes in a network. Each node is expected to broadcast its distance to every other node. Whenever a node receives a new update from a neighbor, it recalculates its minimum distances. Let $A \rightarrow B$ and let $B \rightsquigarrow_n C$. If the current known minimum distance from A to C is more than $n + 1$ hops, then a new shortest path from A to C of distance $n+1$, $(A \rightsquigarrow_{n+1} C)$, through B is discovered in the next round. This algorithm is extremely advantageous because it works asynchronously and is guaranteed to converge if the network remains unchanged for sufficient time. However, this algorithm is difficult to implement in the scenarios we consider. For example, in the just described case, if $A \rightarrow B$ but not

$B \rightarrow A$, then A would have no way of knowing that B can reach C in n hops. Thus, A will never be able to discover the $n+1$ hop path to C.

We modify the above algorithm to work in our environment and call it Reverse Distributed Bellman-Ford Algorithm. The difference between the two algorithms is that in our algorithm, each node aims at finding the shortest distance *from other nodes* to itself rather than from itself to other nodes. Thus, in the case described above, A sends a broadcast message saying that C can reach it in n hops. Since $A \rightarrow B$, B discovers that C can reach B through A in $n+1$ hops. If at B, the previous known route from C is longer than n+1 hops, B can now update the new n+1 hop route from C.

Suppose that $B \rightarrow C$, but C does not know the reverse route (i.e., the route from itself to B). Since B would neighbor-cast the information about the n+1 hop path from C to B, C learns that it can reach B in n+1 hops. The information cycle is complete and C has learnt a reverse route to B.

A precise definition of this algorithm is given below.

- At periodic intervals, each node broadcasts to all its out-neighbors an update message containing the shortest paths of its knowledge from other nodes within the locality to itself.
- Whenever an update is received at node B from in-neighbor A, the following information is extracted.
  - The reverse route from B to A; this information is obtained from the entry for the route from B to A in the update received.
  - If the currently known route from node C to node B ($C \rightsquigarrow_i B$) is longer than the route from node C to node A followed by $A \rightarrow B$ (i.e., ($C \rightsquigarrow_j A$ and $i > j+1$), the newly found shorter route from node C to node B through node A is recorded. Note that the path from C to A is obtained from the update message.

In addition to the length of the shortest route from each node in the locality, the address of the first hop in the shortest route also needs to be included in the updated message. Since we know the first hop in the route from B to A (let us assume it is C), we would also know similarly the first hop from C to A (this information would also be broadcast by A). Thus, we can construct the entire reverse route from B to A. Consequently, each node only needs to broadcast an update message with size of the order of number of nodes in the locality. By ignoring routes of length greater than the locality radius, we can restrict the updates to only contain distance from nodes within the locality. Each node however, is expected to store the most recent update messages from each in-neighbor. This is required to recalculate shortest routes whenever new updates are received. Thus, each node stores information of the order of the product of average number of in-neighbors and average number of nodes in the locality.

The above algorithm employed by SRL is not only efficient but also guaranteed to be loop free. The first hop values in the route updates can be used by each node in order to check for routing loops and hence the discovered sub-routes have loop freedom. Further bounding the route updates by a locality radius avoids the counting to infinity problem faced by many distance vector protocols.

C. Incremental Updates

A large amount of information that is being neighbor-cast at each node is redundant. For example, once node A learns of the new route from node B, there is no need for node C to keep broadcasting the shortest route from B. Node C needs to broadcast this information only if the distance from B has changed to a new value or when the first hop of the route has changed. We optimize the operation of SRL by letting each node to broadcast only the changes in the route information rather than the complete information in each round. Since some of these broadcast messages could be lost because of collisions, each change in route information should be broadcast twice.

There are cases in which it is not sufficient to incrementally broadcast the changes. A node might have moved into a new locality and is in need of route information of its new in-neighbors. It is possible, however, in a static or slow moving environment, that the node might not be able to get the needed reverse routing information if the new in-neighbors broadcast the values only when they have changed. Thus, our algorithm forces each node to neighbor-cast all the routes to itself from different nodes in the locality infrequently. This frequency is set lower than the update frequency but adequately high enough for nodes to discover the new locality within a short time of moving to the new location.

Thus, the overall overhead of reverse route discovery consists of infrequent complete updates with information of the order of number of nodes in the locality and periodic update packets containing only the latest changes to the same information. The update messages also play the dual role of hello messages thus, compensating partly for the extra overhead. It takes about as many rounds as the locality radius before reverse route information can be gathered. The periodicity of the updates must be chosen so that the overhead is tolerable while at the same time the reverse routes can be obtained quickly. The frequency of updates may also depend on the mobility pattern of the nodes in the network. This is subject of our future research.

D. SRL Interface and Added Reliability

Efficiency considerations led us to allow the routing protocols to communicate directly with the MAC layer. Often only the control packets are routed using the reverse routes, thus the data packets can be directly transmitted without incurring any delay because of an extra layer. For this reason, we chose not to provide a totally transparent interface to the routing protocols. They are expected to be aware of the unidirectional and power varying nature of the environment. Further, we believe that the routing protocols can be adapted more efficiently if they use only the required services provide by the sub routing layer. For example, a totally unaware routing protocol might mistake the (longer) reverse routes to be fast, direct-hop routes and start routing data packets along a reverse route.

The fundamental bidirectionality abstraction of SRL is provided by a service through reverse routing or sub-routing. For example, an on-demand routing protocol can use the reverse route to send route replies back to the sender of the request. The route reply packet is then sent to SRL for reverse routing. SRL looks up the reverse route information it stores and finds a reverse route. The packet is then sent using this reverse route. Since the locality radius is typically a small number, the length of the source route remains small. The sub-routed packet is then delivered to the routing protocol at its destination.

In a unidirectional environment, the MAC protocol is unable to provide any guarantee for delivery since both RTS-CTS schemes and hop-by-hop ACK-based schemes cannot be implemented at the MAC layer. SRL could be used to provide a reliable routing service below the routing layer. SRL could use the reverse routes to send acknowledgments back to the sender. SRL can, thus, be used to imitate the reliability mechanism of the MAC layer in a bidirectional environment. Packets can be retransmitted by SRL to achieve this reliability. Since the reverse route is a few hops long, a protocol with a window size of the same order as locality radius is useful.

In addition to the routing services, SRL provides additional services that are sometimes provided by the MAC layer in a bidirectional environment. Since SRL has an inherent neighbor discovery mechanism, it can export this mechanism to higher layers. For that, our implementation of SRL raises new *in-neighbor found* and *in-neighbor lost* events. Routing protocols may use these events to initiate route repairs. Many routing protocols designed for bidirectional networks use packet drops at MAC layer to detect link breaks. The SRL can imitate this behavior by raising a packet drop event whenever it knows that the out-neighbor cannot reach back because no reverse route exists from the out-neighbor. It is possible that the out-neighbor can receive the packet and only the reverse route does not exist. This way, SRL may generate a false packet drop event. However, even the MAC layer can falsely raise a packet drop event while in fact only the acknowledgment has been lost because of congestion.

IV. Setup of Simulation

The Sub Routing Layer has been implemented in a simulation environment. We used GloMoSim [3], a scalable simulator for wired and wireless network systems. It uses the parallel discrete event simulation provided by Parsec [4], a C based simulation language. We first describe the details of simulation environment used to study the performance of SRL. In the following sections we discuss the results of the simulations performed to study the behavior of SRL and a modified version of AODV that uses SRL to provide routing in unidirectional environment.

All simulations were executed with 80 nodes randomly distributed in an area of 400m by 400m for 240 seconds. The nodes were allowed to move according to the random way-point mobility model. Following this model, each node chooses a random position within the terrain to move. Each node then moves to that point with a velocity randomly chosen between a maximum and a minimum. The node then waits for a random interval of time and continues its motion as described above. In our experiments, the minimum speed was always maintained at 0 m/sec. The maximum speed was varied from 0.1 m/sec to 10 m/sec, with the following 11 values: 0.1, 0.2, 0.4, 0.6, 0.8, 1, 2, 4, 6, 8, 10 (all values in m/sec). Because of this, the x-axis is not linear in our graphs showing variations of different metrics with values of maximum speed. Random waiting interval was chosen between 0 sec and 80 sec.

The bandwidth of the physical channel was assumed to be 2 MHz. The radio-layer employs a two-ray path propagation model to simulate signal propagation. In order to work with a reasonably efficient MAC protocol in the unidirectional environment, the following modifications were made to IEEE 802.11 MAC protocol. The RTS/CTS exhange was turned off. Acknowledgments are sent at the receiving side but they are received at the sending side only if the links are bidirectional. The receiving side MAC layer continues to deliver the packets to the network layer. Whenever acknowledgments are not received at the sending side, the MAC layer assumes that the link is unidirectional and starts transmitting the next packet without raising any packet-drop or link-break event. Thus, the modified MAC protocol remains efficient when the links are bidirectional but just broadcasts with carrier sensing when links are unidirectional.

We also modified the GloMoSim simulator to accept different values of transmission powers for different nodes. We then carefully chose five scenarios with different power levels for the 80 nodes. Each node has a value of transmission power picked from a discrete set of values. Both the number of different power levels in the discrete set as well as the diversity in transmission range was varied while choosing these scenarios.

Table I lists the power of transmission assigned to different node in each of the 5 power scenarios. The values of transmission power are measured in dbm. In our simulations, a transmission power of -1 dbm corresponds to a transmission range of about 100m and -9 dbm about 40m. The values are in logarithmic scale with respect to power measured in milliwatts. A difference of -3 dbm between two values indicates a ratio of 2 in transmission power in milliwatts. Scenarios 1 and 2 choose values of transmission radius from a set of 2 discrete values, scenarios 3 and 4 use a set of 3 values to pick the transmission power and scenario 5 picks 4 different values of transmission power. The magnitude of transmission power picked by the nodes affects the overall connectivity of the network. For example, nodes with -2 dbm transmission power have a higher reachability than nodes with -3 dbm transmission power. As a result, different scenarios produce topologies of different connectivity in the simulations. We refer to the average difference in the transmission power as the diversity of the scenario. Scenario 4 has the highest diversity among the scenarios used to perform simulations.

The following parameters were used while simulating the sub routing layer. The periodicity of hello messages and incremental update messages were both set to 500 milliseconds. Hello messages is not sent in an interval whenever update message is sent. The frequency of complete updates was set at once in 4.5 seconds. Whenever 3 consecutive hello or update messages are missed, the link from that in-neighbor is declared to be broken. We varied the locality radius of SRL as 1, 2 and 3.

We first describe the observations on two metrics measured during the simulations in order to explain the role of the power scenarios in our experiments. More detailed analysis of the simulations of SRL are presented in the next section.

A. Average Number of Neighbors

Figure 2 shows the variation of the average number of in-neighbors seen by each node under different power scenarios. An increase in average neighborhood increases the connectivity

TABLE I

VALUES OF TRANSMISSION POWER ASSIGNED TO DIFFERENT NODES IN 5 POWER SCENARIOS.

Scenario1		Scenario2		Scenario3		Scenario4		Scenario5	
# nodes	power (dbm)	# nodes	power (dbm)	# nodes	power (dbm)	# nodes	power (dbm)	# nodes	power (dbm)
30	-2	30	-3	20	-2	20	-1	10	-2
50	-8	50	-7	30	-5	30	-5	20	-4
				30	-8	30	-9	20	-6
								30	-8

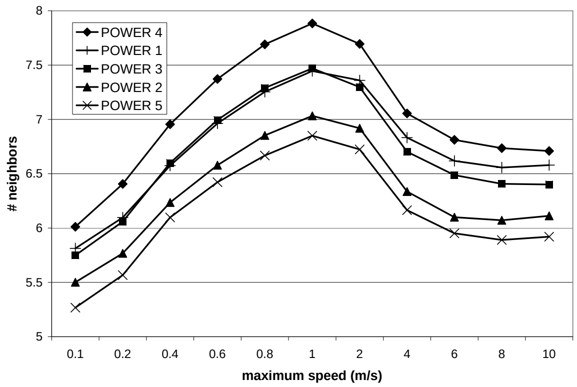

Fig. 2. Average number of neighbors per node.

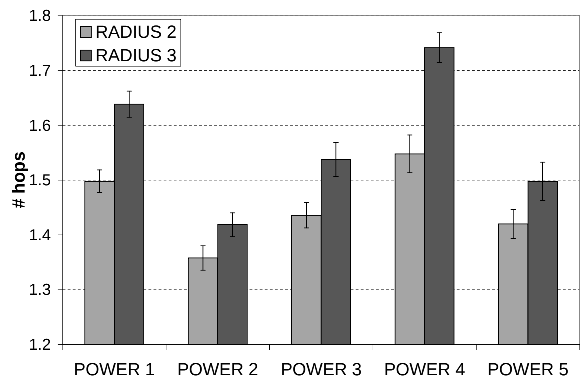

Fig. 3. Average reverse route length.

of the network and hence shortens route lengths. It could also increase packet loss due to collisions. The average neighborhood depends on the average node density and the transmission powers of the nodes in the network. In our experiments, the average node density is not varied as we use a fixed number of nodes in a fixed area. Figure 2 shows that the average neighborhood is the highest for power scenario 4, which has the highest transmission power of -1 dbm. The average neighborhood with scenario 2 is less than that of scenario 1 indicating that the average reachability of nodes in scenario 1 is higher than in scenario 2.

Another interesting feature of these curves is their variation with maximum speed. It can be observed that the values increase with increase in maximum speed between 0.1 m/sec and 1 m/sec while decrease and become steady after that. This observation can be explained based on the random way-point mobility model as follows. At very low speeds the values are steady because of limited movement. As the speed is increased and the nodes move, they make several temporary neighbors during their motion. These transient neighbors increase the average neighborhood seen by a node. However at higher speeds, in general, the time spent near other nodes is too small to form neighborhoods.

B. Average Reverse Route Length

The average reverse route length is the average length of reverse routes established by each node during the simulation of SRL. It affects the average latency seen by reverse routed or sub-routed packets. It must be noted that the links for which reverse routes could not be found are not counted. Thus, the average reverse route length is always less than the locality radius. Figure 3 shows the variation of this quantity for different power scenarios with different locality radius. The error bars on the figure indicates the standard deviation of measured values. For locality radius equal to 1, the measured values of length of reverse route are all 1, because SRL finds the bidirectional links of the network. Hence only the values for locality radius equal to 2 and 3 are plotted in the figure. The average reverse route length increases slightly with the locality radius, because SRL finds more routes if the reverse route is allowed to be longer.

The diversity of transmission power influences directly the average reverse route length; it can be seen that Scenario 4, with the greatest diversity, shows an increased value for average reverse route length. Note that even though scenario 5 has a larger number of distinct power levels, the average reverse route length seems to depend only on the diversity of transmission range. For the same reason, scenario 1 has a higher average reverse route length than scenario 2. It should be noted that when the radius increases from 2 to 3 the increase in the average reverse route length is at most by 0.2, leading us to believe that larger radius values will not significantly improve the number of routes found.

V. PERFORMANCE OF SUB ROUTING LAYER

In this section, we describe the results of the simulation of the sub routing layer. We measured different aspects of performance of SRL.

A. Average Number of Local Nodes

Figure 4 shows the average number of local nodes encountered by each node for different values of locality radius. The

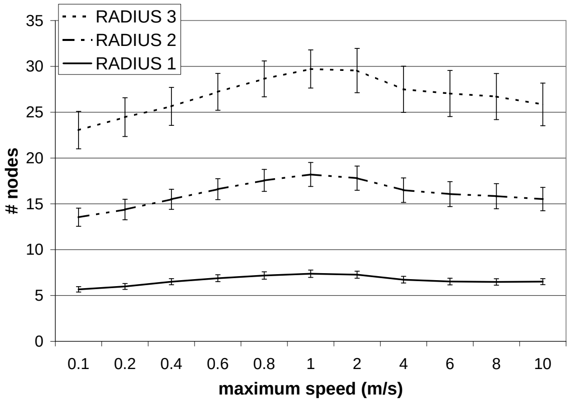

Fig. 4. Average number of nodes in each locality.

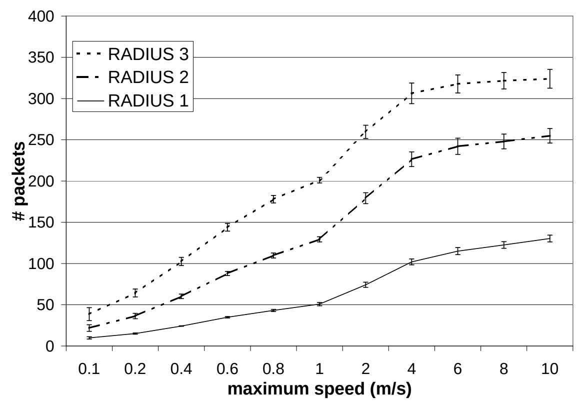

Fig. 5. Average number of incremental update packets sent.

average has been taken from both the values of 80 nodes in each run and the 5 power scenarios, and the standard deviation is indicated by error bars. When the radius is 1, the values are identical to the average neighborhood. We can see the general increase in the values as the locality radius is increased. The curve shows a brief increase for speed between 0.1 m/sec and 1 m/sec for the same reason explained in the description of average number of neighbors (see Section IV-A). The average number of local nodes influences both the amount of state stored in each node as well as the size of update packets.

B. Number of Update Packets Sent

Figure 5 shows the variation of the number of incremental update packets sent from each node with average taken over 5 power scenarios, for different locality radii, and for different values of maximum speed. It can be observed that the number of incremental update packets sent directly depends on the mobility. This is because the higher the speed, the more link breaks there are, and therefore higher number of updates generated. The dependence of number of update packets on the locality radius can also be observed. With a larger radius, the number of local nodes about which information needs to be carried increases, increasing the number of incremental updates sent. The number of complete update packets sent remains more or less a constant over the different executions and hence are not plotted in figure 5. We found that on average 53 complete update packets were sent by each node (min and max values are 52.75 and 53.5 for any one simulation).

C. Size of Periodic Packets Sent

Since the update (incremental or complete) and hello packets are sent periodically, their size forms an important factor affecting the level of congestion in the neighborhood. The information carried by the complete update packets is proportional to the number of nodes in the locality at that time. The average number of updates carried by the incremental update packets however are typically less because only information about the routes affected by a link break or a new link is sent. We tried to measure the average size of the periodic packets sent. Each update sent contributes 9 bytes to the size (IP address of the source node, IP address of the first hop, and the distance). In addition each periodic packet includes an IP header of 20 bytes and a MAC layer header of 12 bytes. Thus a hello packet would be counted to have a size of 32 bytes. In this measurement both update packet and hello packet are counted as periodic packets. Since the hello packets are sent once in 500 milliseconds, totally about 480 periodic packets would be sent in 240 seconds.

The variations of the average size of these periodic packets with maximum speed can be seen in Figure 6. The standard deviation of the measurements are indicated by error bars. When the locality radius is 1, the size of periodic packets is quite small (between 38 and 42 bytes). With locality radius 1, SRL is effectively broadcasting the list of its in-neighbors. We believe that this overhead is acceptable and would not lead to congestion. When locality radius is 2, the average size of periodic packets is still under 60 bytes for most cases. However, for reasonable values of maximum speed this blow-up is quite reasonable and does not create significant congestion. With locality radius 3, the average size increases to about as much as 90 bytes for high speeds and 70 bytes for moderate speeds. Even though we could not directly measure the effects of congestion, we observed (as explained in next section) that the effects of congestion are discernible. Hence, there is a trade-off between choosing higher values of locality radius for finding more reverse routes versus the effects of congestion. We find that for the scenarios we studied, a locality radius of 2 is a good balance. However, we could significantly lower the overhead if each node could dynamically change its locality radius; this is outside the scope of this paper and subject of current research.

VI. Unidirectional Routing with SRL

A number of routing protocols, such as AODV and DSR, have been proposed for mobile ad hoc networks. These protocols are efficient when all the links are bidirectional. However, their performance degrades in ad hoc networks with unidirectional links, especially when nodes change transmission power levels. A qualitative performance analysis of the above protocols in such networks was presented in Section II. SRL provides a bidirectional abstraction of the unidirectional network. The routing protocols could use this abstraction and work effectively even in the presence of unidirectional links. Minimal changes are required in these routing protocols to work in unidirectional networks. In this section, we show how AODV could be used to route over unidirectional links with SRL and present an evalua-

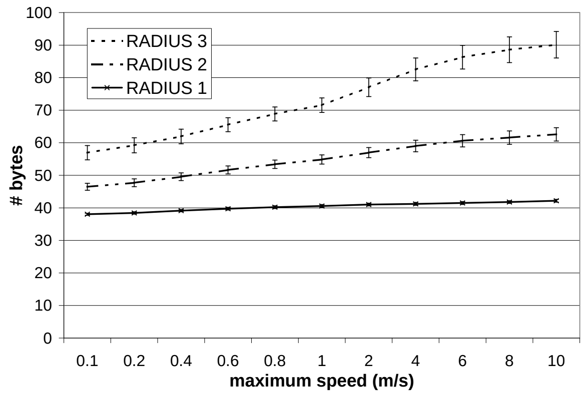

Fig. 6. Average size of each update/hello packet.

tion of this scheme.

AODV is an on-demand routing protocol for bidirectional ad hoc networks. Whenever a packet needs to be sent to the destination node for which a route is not known, the source broadcasts a route request packet. When the route request packet reaches the destination or an intermediate node with fresh enough route to the destination, the destination or the intermediate node sends a route reply back to the source. In a bidirectional network, the reply can be unicast by reversing the request path. However, in a unidirectional network the reverse path may not exist. AODV [1] proposes to solve this problem by keeping track of the unidirectional links in a *black-list*. A node sending a route reply to its neighbor expects to receive an acknowledgment and when the acknowledgment is not received that neighbor is added to the black-list. Future route requests received through this neighbor are ignored. Each entry added to the black-list is deleted after a specified lifetime. The black-list gives an approximate list of unidirectional links. It is possible that in future, the neighbor is actually close enough to hear from this node but the route requests coming from it are ignored. Further, the presence of the unidirectional link is detected only when an attempt is made to send route replies through it. This may be too late and sometimes prevent routes from being discovered. Moreover, with increased mobility the black-list becomes a poor approximation.

The black-list enables AODV to route packets only along the bidirectional links in the network. However, there could be certain nodes reachable only if a few unidirectional links are included in the route. AODV would not be able to discover such routes. Also, a link in a route that is presently bidirectional may become unidirectional. In such cases AODV has to declare that route to be broken and start a fresh route discovery.

We propose and study AODV on SRL, for the following reasons. By using the services of SRL, AODV obviate the necessity for a black-list. Also, because SRL maintains reverse routes, it can identify with good accuracy whether a link along which a route request is received is unidirectional (in fact, SRL with a locality radius of 1 is sufficient to do this).

We modified AODV to route packets through the unidirectional links. Route replies are sent through a multi-hop reverse route. In addition to the route replies, the route errors are also sub routed through this reverse path with the help of SRL. The behavior of AODV remains the same as before except that the route reply and route error packets are sent through the sub routing layer. This may induce an additional delay in receiving route replies but only affects the route discovery latency as the data packets continue to travel the same or shorter (forward) path.

While routing along unidirectional links, the MAC protocol used may not be able to detect link breakages. But the modified AODV can use information from the Sub Routing Layer to detect link breakages. Whenever a data packet is sent along a link to the next hop, the SRL is consulted to see if the next hop node has a reverse route back. If no reverse route is found from the next hop a link break event is detected and the data packet is assumed to be dropped. In addition, each node detecting a break of an in-neighbor locally broadcasts a AODV route repair packet if the in-neighbor is a previous hop in one of the active routes. This method of detecting link breakages does not guarantee reliability against packet loss due to collision or overflow; such reliability guarantees are left for higher layers such as TCP. However, if reliability guarantees are indeed needed at this level, the SRL can also be used to sub route acknowledgments for data packets.

We simulated the modified AODV along with SRL and studied the performance of this protocol. For comparison, we implemented a version of AODV as described in [1]. This implementation was modified to be used for unidirectional routing with SRL. Optimizations specified in [1] such as expanding ring search, and cache based route replies have been incorporated both in AODV and SRL+AODV. However, we chose not to implement local error recovery for simplicity. This would not affect the quality of the comparative results between AODV and SRL+AODV. We conducted the experiments with the same scenarios and parameter values for SRL as described in previous sections. For AODV the values recommended in [1] were used. For AODV executions, we used IEEE 802.11 as the MAC protocol. Since we could rely on either the MAC protocol or SRL to detect link breaks, additional AODV hello messages were not used.

We also tried to simulate DSR as defined in [6] for unidirectional networks. However, we observed that the performance of DSR was very poor for the topologies and route pairs we considered. We also observed that the DSR performs very well when fewer number of routes are used during the simulation. There appears to be a problem with the scalability of DSR in unidirectional networks. We already stated that RREP-explosion problem could significantly affect DSR throughput. In addition to that, hop-level acknowledgments create several short routes to be discovered and maintained. DSR attempts route maintenance only when packets are sent along a route or overheard route errors are being processed. But this form of on-demand route maintenance is not adequate when route replies or acknowledgments are being routed. As a result the route cache stores many broken routes leading to increased loss of both control and data packets. Further, the specification of DSR's operation over unidirectional links is ambiguous at many places. For these reasons, we do not present the results observed from simulations of DSR.

We ran simulations for the same topology and 5 power scenarios described in Section IV. A constant bit rate (CBR) generator was used as test application. All packets generated were of length 512 bytes. The rate at which packets were being generated was different for different routes. During the 240 seconds of simulation active packet generation by CBR lasted from 50 to 200 seconds. 20 pairs of source and destination nodes were picked randomly for each execution. We ran 25 experiments for each data point, with 5 different sets of 20 source-destination pairs and 5 different power scenarios. The data we present is an average of these 25 executions.

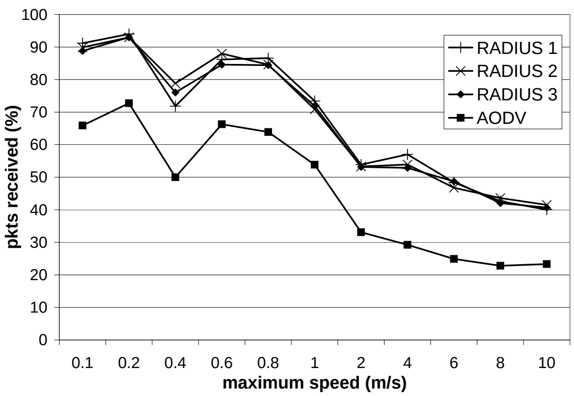

Fig. 7. Percentage of Packets Delivered.

A. Packet Throughput

Figure 7 shows the variation of the packet delivery for different values of maximum speed. The values are plotted as a percentage of packets that were generated by CBR; in cases where routes were not found, all packets for that source were dropped and not counted as delivered but counted as sent. The graph shows a good overall improvement in packet delivery achieved by using SRL and routing along unidirectional links. We see that the modified protocol performs quite well for moderate speeds between 0.1 m/s and 1 m/s. There is an average improvement from about 60% for AODV to about 80% for SRL+AODV. At higher values of maximum speed the packet delivery drops steadily for both versions of AODV, but SRL+AODV performs much better relative to AODV. At higher values of mobility, an increased number of link breakages decreases the average time for which a route remains good and the frequent route repairs increase the loss of packets. We also observe that when maximum speed is 0.4 m/s there is a momentary decrease in the packet delivery. This observation is explained later.

B. Goodput

We define goodput as the ratio (percentage) of the number of packets that were delivered to the number of packets actually sent (i.e., sources with an active route). The packets dropped at the source because no routes could be found to the destination are not counted as sent by the goodput metric. In Figure 8 we see the variation of the goodput for different maximum speeds. We observe that the goodput of AODV is about the same as SRL+AODV. We can conclude that the percentage of packets lost due to route maintenance is approximately similar to both the versions. In comparison with Figure 7, this clearly illustrates that the decreased packet delivery for AODV was because of its inability to find routes to the destinations.

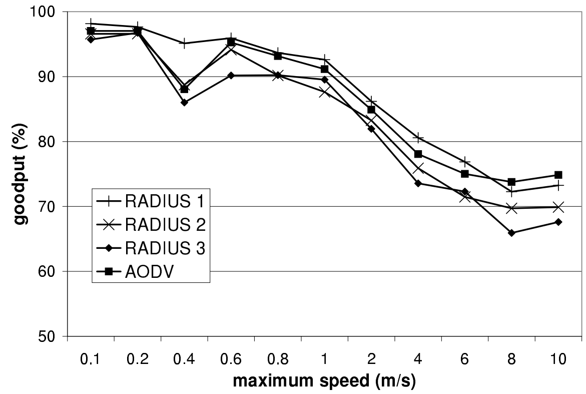

Fig. 8. Average Goodput.

Figure 8 also shows the goodput of the SRL+AODV protocol when the locality radius of SRL is changed. We observe that the goodput drops slightly as the locality radius is increased. Also at higher speeds, the goodput of AODV is slightly higher than that of SRL+AODV. Since SRL uses periodic packets to detect link breaks, there is a slight delay in the detection of link breaks. Further, the news of link-break detection stake a few more updates to reach the upstream node on the link. As the SRL radius increases this delay also increases. This increased delay would mean more packets are lost before the link breakage is detected and the route is repaired. By choosing different values for update and hello frequencies we could alleviate this problem to a certain extent; this is subject of our future research. We also observe that there is a momentary decrease in goodput and packet delivery of all the simulated protocols when the maximum speed is 0.4 m/s. We conjecture that this is because of the periodicity of update messages, which are synchronized with the device mobility. That is, certain link breaks affecting many routes at the same time affects the performance of the protocol. We are running simulations with different topologies and power scenarios to make more accurate measurements.

C. Average Route Length

The average route length is the average number of hops traversed by the packets that were sent from the source and received at the destination. Figure 9 shows the variation of the average route length for AODV, SRL+AODV for different values of locality radius. We observe that, in general, as we increase the locality radius, we find shorter routes to the destination, since SRL+AODV was able to use unidirectional links (with increased locality radius).

The variation of the average route length with respect to power scenarios follows a pattern related to the average neighborhood explained before. Scenarios with higher values of transmission power can be observed to have shorter routes on average. The average route length for AODV, SRL+AODV with radius 1 are approximately the same because SRL with radius 1 only uses bidirectional links. The difference between SRL+AODV with locality radius 2 and 3 is very small indicating that the gain by using locality radius 3 is not very high.

The simulations with SRL+AODV show us how using unidirectional links to route data could increase the packet delivery and also reduce the average route length. Decreasing the route lengths is very advantageous because shorter routes are easier to maintain and have smaller latencies. Our experiments also suggest a small locality of radius is sufficient to use unidirectional links effectively.

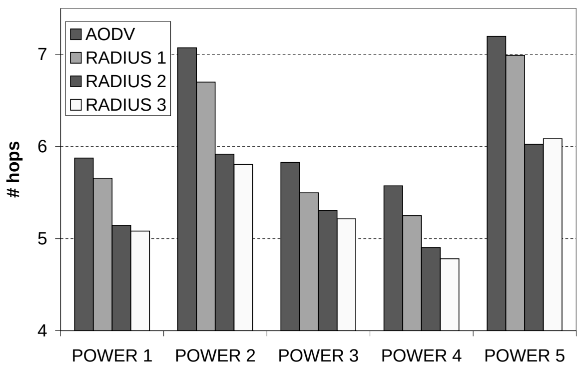

Fig. 9. Average route length.

VII. Concluding Remarks and Future Work

The sub routing layer (SRL) presented in this paper provides an ideal framework for allowing existing routing protocols to function in the presence of unidirectional links and changing power levels. SRL provides a bidirectional abstraction of the network to the routing protocol at low cost. It is extremely scalable due to the locality of control messages, which is parameterized. Further, the algorithm employed to maintain reverse routes does not suffer from counting to infinity problem and avoids formation of loops in the reverse routes. We have shown that AODV profits tremendously from having SRL provide the bidirectional abstraction, and that the cost of such abstraction is low. Our current work is to develop routing protocols over SRL to optimize power consumption. Clearly, SRL can form the basis of an adaptation layer that conserves power without changing existing routing protocols.

References

[1] C. Perkins, E. Royer, S. Das "Ad-Hoc On Demand Distance Vector(AODV) Routing". *IETF Internet Draft, draft-ietf-manet-aodv-08.txt*, March, 2001. Work in Progress.

[2] Elizabeth M. Royer and Charles E. Perkins. "Ad-hoc On-Demand Distance Vector Routing". *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, Feb. 1999, 90-100.

[3] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. "GloMoSim: A Library for Parallel Simulation of Large-scale Wireless Networks". *Proceedings of the 12th workshop on Parallel and distributed simulation*, May 26 - 29, 1998, Banff Canada, 154-161.

[4] R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, B. Park, and H. Song. "Parsec: A Parallel Simulation Environment for Complex Systems". *IEEE Computer*, Vol. 31(10), October 1998, pp. 77-85

[5] D. B. Johnson, and D. A. Maltz. "Dynamic Source Routing in Ad-Hoc Wireless Networks". *Mobile Computing*, edited by T. Imielinski and H. Korth, chapter 5, pp. 153-181, Kluwer, 1996.

[6] D. B. Johnson, D. A. Maltz, Yih-Chun Hu, and Jorjeta Jetcheva. "The Dynamic Source Routing protocol for mobile ad hoc networks". *IETF Internet Draft, draft-ietf-manet-dsr-05.txt*, March 2001. Work in Progress.

[7] Vincent D. Park and M. Scott Corson. "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks". *Proceedings of IEEE INFOCOM*, Kobe, Japan, April 1997.

[8] Z. J. Haas and M. R. Pearlman. "The Zone Routing Protocol(ZRP) for Ad hoc Networks". Internet draft - Mobile Ad hoc NETworking (MANET) Working Group of the Internet Engineering Task Force (IETF), March 2001.

[9] R. Prakash. "Unidirectional Links Prove Costly in Wireless Ad-Hoc Networks". *Proceedings of ACM DIAL M'99 Workshop*, Seattle, WA, August 1999, pp. 15-22.

[10] Sanket Nesargi and Ravi Prakash. "A Tunneling Approach to Routing with Unidirectional Links in Mobile Ad-Hoc Networks". *In Proceedings of IEEE ICCCN 2000*, Las Vegas, Nevada, October 16 - 18, 2000.

[11] Prasun Sinha, Srikanth Krishnamurthy and Son Dao. "Scalable Unidirectional Routing with Zone Routing Protocol (ZRP) Extensions for Mobile Ad-Hoc Networks". *In Proceedings of IEEE WCNC 2000*, Chicago, September, 2000.

[12] J. Gomez, A. T. Campbell, M. Naghshineh, and C. Bisdikian. "Power-aware routing in wireless packet networks". *In Proceesings of the 6th IEEE International Workshop on Mobile Multimedia Communucations*, November, 1999.

[13] S. Corson, S. Papademetriou, P. Papadopoulos, V. Park, A. Qayyum, "An Internet MANET Encapsulation Protocol (IMEP) Specification". *IETF Internet Draft, draft-ietf-manet-imep-spec-01.txt*, August 1999, Work in Progress.