DYNAMIC ROUTES THROUGH VIRTUAL PATHS ROUTING FOR AD HOC NETWORKS

Abdulrahman H. Altalhi Golden G. Richard, III
Computer Science Department
University of New Orleans
New Orleans, LA, 70148 USA

Abstract
Routes in ad hoc wireless networks are typically discovered and selected by routing protocols based on a particular single criterion such as shortest path or lowest cost of power. Furthermore, those routing protocols do not capture the mobility of the network nodes nor do they factorize it into the operation of the protocols. In this paper, we feature the Virtual Paths Routing (VPR) Protocol for ad hoc wireless networks. VPR captures the mobility of the nodes and provides correct, highly adaptive, and dynamic route creation and maintenance between nodes with multiple selection criteria for routes.

Keywords: Ad Hoc Networking, Routing, and Mobility.

1. Introduction

Rapid progression in technology for mobile devices, including laptops and handheld computers, and the availability of inexpensive wireless networking hardware has resulted in serious interest in wireless connectivity among mobile users. One approach to provide wireless connectivity is through the formation of an ad hoc wireless network. This approach does not assume the support of any pre-existing infrastructure, but instead, uses other nodes in the ad hoc network as routers to facilitate message delivery. One of the challenging problems in this type of network is the routing process. This paper features a distributed, on-demand, dynamic, and adaptive routing protocol that takes into account the mobility of the nodes and allows for the selection of routes based on various criteria. Beside the typical shortest path and Power-Aware route selection criteria, VPR provides Load-Aware and Mobility-Aware routes selection criteria. VPR observes the mobility and the load of the nodes and uses these observations to adjust its functionality to find routes based on the mobility or load condition of the nodes. In VPR, each node maintains routing information to other nodes, discovers new routes as necessary, and performs maintenance on known routes.

Several Routing protocols for ad hoc wireless networks have been proposed. Some of the extensively studied protocols are DSDV [1], ADV [2], AODV [3], DSR [4] [5] [6], and ISR [7]. However, these protocols do not factorize mobility in their functionalities, and they provide a single selection criterion for routes. The dynamic nature of ad hoc wireless networks strongly suggests a different approach in which the mobility of the nodes is factorized into the operations of the protocol, and paths are discovered and selected based on multiple criteria.

The rest of this paper is organized as follows. In Section 2, we describe, in a high level, the design and operations of the VPR protocol. In Section 3, we detail the mobility monitoring technique of VPR and how it is used to adjust the protocol operations. In Section 4, we illustrate the capability of VPR to find and select routes based on several criteria. In Section 5, we present a simulation study to evaluate the effectiveness of the mobility monitoring technique of VPR by comparing VPR to ISR. Finally, in Section 6, we conclude our work.

2. Virtual Paths Routing (VPR) Protocol

In this section, we briefly describe the Virtual Paths Routing (VPR) protocol. The key purpose of VPR [8] is to provide correct, efficient, and dynamic route creation and maintenance among the network nodes. The VPR utilizes two well known routing techniques, namely source and table routing. It is a distributed, on-demand, and adaptive protocol that is comprised of two phases. The first phase is path creation, which is initiated when a source node needs to communicate with a destination node. All of the nodes within the vicinity potentially participate in this phase, which may yield more than one path. The second phase is path maintenance, in which the protocol monitors all the established paths. In this phase, the protocol observes local connectivity, link breakages, and node mobility and uses these observations to adjust its internal parameters accordingly. During this phase, nodes with active path continually and controllably report their presences by broadcasting periodical HELLO messages.

To deliver a packet between a source and a destination node, a virtual path must be established between the two nodes. A virtual path is simply a route, or sequence of nodes, in which all the nodes are aware of the existence of the path and they monitor its activities. Each node on any given virtual path knows its predecessor and successor nodes on the path. All the packets to be delivered through a particular virtual path are marked with a key that uniquely identifies the virtual path to be used; and each node passes the packets to the next node on the path until they reach their final destination. Since every node operates as a router, a node must be able to handle more than one virtual path. To accomplish that, each participating node maintains a virtual paths routing table. This table contains the currently active paths, which are defined as the paths that are in use and fully operational.

The path creation in VPR is a two-phase process. The first phase is the Path Discovery, in which the source node uses a controlled flooding technique to determine the path through the ad hoc network to the destination. At the end of this phase, the source node may have multiple distinctive paths through which it may reach the destination node. Every path consists of nodes that can relay messages from source to destination. The second phase is the Path Set Up on all the nodes on the list that was obtained previously. At the end of this phase, each node on the path has an entry in its virtual paths routing table for the newly created path. At the end of the dialog between a source and destination node, or when the virtual path is broken, the protocol deactivates the path by removing it from the virtual paths routing tables of all the involved nodes. Every data or control packet routed by VPR contains a VPR header. That header contains an 8-bit flag (called the *operation flag*) that is used to control the operations of the protocol. VPR uses a technique to monitor the mobility of the nodes and then adjusts its operations accordingly. We detail this feature in the next section.

3. Mobility

VPR defines a **mobility indicator** that reflects the current level of mobility. The indicator is a variable with possible values ranging between zero and one. A value of one indicates a high level of mobility in which the network nodes are in constant movement. Whereas, a value of zero indicates a low level of mobility in which the nodes of the network are stationary. To capture the mobility level by and around a node, the protocol monitors the operations it performed on the virtual paths routing table entries. Particularly, it monitors the deletion of entries from the table due to broken paths. These deletions are clear indications of a high level of mobility within a node's environment. The insertion operations are not monitored since they are indications of a high level of activity more than level of mobility.

When a node deletes entries from its routing table because of broken paths, the value of the mobility indicator should be increased to reflect a high level of mobility. In contrast, when the node does not delete any entries for a certain period, the value of the mobility indicator should be decreased to reflect a low level of mobility. The values of the mobility indicator are calculated as follows:

Assume

MLI: *Mobility Level Indicator*

NED: *Number of Entries Deleted* from the routing table since last adjustment of *MLI*.

CNOE: *Current Number of Entries* in the routing table.

MAF: *Mobility Level Adjustment Factor (Constant =0.2)*

NAT: *Next Adjustment Time,* which is the time to be added to the node's current time to schedule next adjustment.

LAT: *Last Adjustment Time,* which is the time that has elapsed since the last *MLI* adjustment. It is the last calculated *NAT*.

UB: *Time Upper Bound,* which is the maximum time allowed between two consecutive adjustments.

LB: *Time Lower Bound,* which is the minimum time allowed between two consecutive adjustments.

Then, the Mobility Level Indicator (*MLI*) is calculated as follows:

$$MLI = (NED / CNOE) \qquad 1$$

The time of the next adjustment (*NAT*) is calculated based on the operations that were performed on the routing table. If the node deleted an entry, it decreases its time for the next adjustment of *MLI* according to the following equation:

$$NAT = LAT - (MLI * (UB-LB) * MAF) \qquad 2$$

If the node did not delete an entry, it increases its time for the next adjustment of MLI according to the following equation:

$$NAT = LAT + (MAF * (UB-LB)) \qquad 3$$

In the first equation, the use of the quantity (*NED* / *CNOE*) is to allow the number of entries deleted and current number of entries to determine the value to be used to set the mobility indicator. In Equation 2, the higher the value of the Mobility Level Indicator, the higher the value to be used to decrease the *NAT* will be. In Equation 3, *NAT* is increased when the node does not delete any entry from its table. The values of *NAT* are kept within *UB* and *LB*. If Equation 2 yields a value that is smaller than *LB*, *NAT* is assigned the value of *LB*; and if Equation 3 results a value that is greater than *UB*, *NAT* takes the value of *UB*. Since some of the entries might be deleted due to **node failures** and not to a high level of mobility, the rate that is used to decrease the *NAT* is lower than the rate to increase it. If the number of the current entry is zero and the number of the deleted entry is greater than zero, the new *MLI* is set to equal one. This adjustment takes place **periodically**. When a node calculates a new *NAT*, it initializes *NED* to zero and schedules the next adjustment to a time equal to the current node's time plus the new *NAT*.

The mobility of the nodes of an ad hoc wireless network greatly impacts the performance of the routing protocols designed for such a network. For instance, the mobility of the nodes alters the validity of cached routes that were collected by the nodes [13]. All the proposed protocols either use no expiry time for their cached routes or use a constant value. Both of these choices certainly do not tune with the dynamic environment of ad hoc wireless networks. In fact, both choices would degrade the performance of the protocols. If no expiry time is used, stale routes may be used, which may result in route errors and a corruption of other cached routes by other nodes. If a constant value is used, it is possible to invalidate a viable route. While the expiry times are static for most protocols, the dynamic nature of ad hoc networks suggests a dynamic approach. In VPR, the expiry time values are varied in response to changes in the mobility level by, and around, a given node. When the level of mobility is high, the expiry time must be short to prevent the node from using invalid or outdated routes. On the other hand, when the mobility level is low, the expiry time must be long to prevent the node from losing valid route or performing unnecessary overhead.

4. Path Selection

VPR is designed to find routes based on several criteria. This feature allows the application layer to request routes that satisfy certain requirements. For example, applications with a high volume of traffic would give preference to routes that balance the load over the network. Some applications may require shortest paths to destination nodes, while other applications might prefer routes with the longest possible life using mobility-aware routes.

There are two reasons behind the ability of VPR to find routes based on any particular requirement. First, VPR is designed to be able to collect data about the network status during the Path Discovery Phase. Second, every participating node is aware of the paths it maintains.

Three bits from the operation flag are used to indicate what type of data the protocol is collecting during the Path Discovery Phase. The data to be collected are added to the VPR header of the Path Discovery packet while it is navigating through the network to the destination. The Path Discovery Reply packet, which is constructed by the destination node, carries this data back to the source node. Upon receiving the Path Discovery Reply, the source node has all the data it needs to make a path selection based on a certain criterion. For this mechanism to work, the destination node must reply to all Path Discovery requests it receives and the initiator of the path discovery must wait for an interval of time known as the PATH_RPLY_WAIT after starting the Path Discovery process. This interval allows more than one reply to arrive at the initiator before it starts the path selection and creation process.

4.1. Shortest Routes

The Path Discovery process, as described in [8], may yield more than one path to the same destination. The replies contain the node list of each path. The number of hops between the initiator and the destination is equal to the number of nodes on this list. The path with the least number of hops is chosen if the shortest path criterion is selected. This is the standard behavior of VPR.

4.2. Load-Aware Routes

To balance the load over the network, VPR may prohibit nodes that maintain a certain number of virtual paths from forwarding any Path Discovery Packet. This prohibition prevents the creation of new Virtual Paths through those nodes. The VPR header of the Path Discovery packet includes a field that is called *Max Paths*. Each node that receives the Path Discovery packet compares the number of virtual paths it maintains to the value of the field. If the number of paths it maintains is greater than or equal to the value of the field, the node simply drops the packet. Otherwise, it processes the packet according the Path Discovery Rules described in [8]. The value of the *Max Paths* field is set by the initiator of the Path Discovery. The value of the field is exponentialy incremented after each unsuccessful attempt to discover a path.

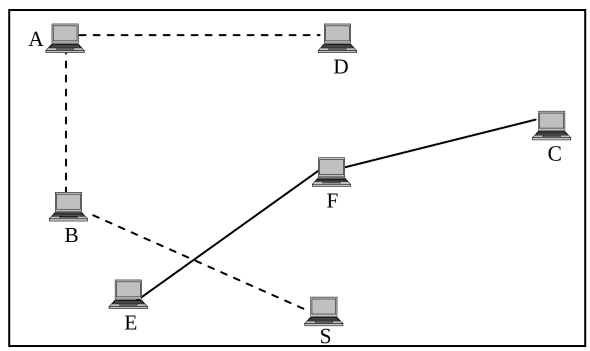

Figure 1: Simple Ad Hoc Network.

The simple ad hoc network depicted in Figure 1 illustrates this technique. The network consists of seven nodes: A, B, C, D, E, F, and S. There is only one active virtual path in the entire network. This path is between Node C and Node E through Node F.

When Node S is in need of a route to Node D, it broadcasts a Path Discovery packet with the *Max Paths* field set to one. The packet would be dropped by Nodes F and E since they maintain one virtual Path. Node C is out of the transmission range of Node S. Node B, on the other hand, will forward the packet to Node A which will deliver the packet to Node D. Then, Node D will send a Path Discovery Reply packet to Node S. To a certain extent, this technique balances the load over the network and avoids creating highly congested areas. Load balancing improves the overall performance of the protocol.

4.3. Mobility-Aware Routes

The mobility of the nodes on any route influences the possible lifetime of that route. The higher the mobility levels of the nodes, the shorter the lifetime of the route will be. If the routes with longest possible lifetime are desirable, VPR chooses the route with nodes having the least mobility level on all possible routes.

In Section 3, we described how VPR captures the Mobility Level of a node. To obtain mobility-aware routes, VPR collects the highest Mobility Level Indicator (*MLI*) among all nodes on each path at the Path Discovery time. An extra field in the Path Discovery packet is used to collect the *MLI*s while the Discovery packet is being propagated. The field is initialized to zero, which is the lowest level of mobility, by the initiator node. Each node that forwards the discovery packet compares the value in the extra field to its current *MLI*. If the node's current *MLI* value is greater than the value in the extra field, the node assigns its *MLI* value to the extra field. The destination node piggybacks these collected *MLI*s on the Path Discovery Replies it sends to the source node. Upon receiving the replies, the source node simply chooses the route with the smallest value in the extra field. The route with the smallest value in the extra field is the route with lowest level of mobility.

5. Simulation Study

To measure the effectiveness of the mobility monitoring technique, we compare the VPR protocol to ISR using *ns-2* network simulator [9], which includes a mobility extension that was ported from CMU's Monarch Group's mobility extension to *ns*. CMU's Monarch mobility extension to *ns-2* allows the simulation of multi-hop ad hoc wireless networks. The extension includes functionalities to simulate node movements, and transmitting and receiving on wireless channels with a realistic radio propagation model. We modeled our network interfaces after the Lucent WaveLan DSSS IEEE 802.11 product with a transmission rate of 2 Mbps. The interfaces use the IEEE 802.11 Distributed Coordination Function (DCF) [10] MAC protocol, which utilizes carrier sensing for collision avoidance.

For the ISR simulation, we used the latest version available from the VINT project that comes with *ns-2*. That version includes an implementation of DSR with a Implicit Source Routing (ISR) technique. We simulated ISR in non-promiscuous mode. The promiscuous mode does not give DSR (the base protocol for ISR) a significant improvement [11]. We chose to simulate ISR and compare it to VPR because ISR is the closest protocol to VPR. ISR creates temporary logical flows to route traffic between the nodes. Moreover, ISR is based on DSR, which is a well-studied and competitive protocol.

Parameter	VPR	ISR
Time between retransmitted requests	500 ms	500 ms
Ring zero search	On	On
Send buffer size	64	64
Time a packet can live in send buffer	30 s	30 s
Promiscuous mode	N/A	Off
Primary cache size	64	30
Reply to requests from the cache	On	On
Salvaging packets using the cache	On	On
Interface queue size	50	50
Hello Interval	Dynamic	N/A

Table 1: Simulation Parameters.

5.1. Traffic and Mobility Models

The traffic type between nodes used in our simulations was Constant Bit Rate (CBR) traffic. The source and destination nodes were randomly selected, and each simulation shows the results of 30 connections. The size of each CBR packet was 148 bytes, which includes 128 bytes of data and 20 bytes for the IP header. The send rate we modeled was four packets per second. The mobility patterns in our simulation followed the *random waypoint* model [12]. In that model, each node starts at a random location, chooses a new location in a rectangular space (1500 x 300 m^2) randomly, and starts its trip to the new location at a randomly chosen speed (uniformly distributed between 0–20 m/sec). After reaching the new location, a node pauses for a period (called the *pause time)*, and then starts a new trip to a new location. We varied the mobility of the nodes by varying the *pause time* values. The results we present in this paper are based on simulation runs of 50 nodes. Each run is 500 seconds. Ten runs of different traffic and mobility scenarios are averaged to generate each data point. However, identical traffic and mobility scenarios were used for both protocols.

5.2. Results

We used three performance metrics to compare VPR to ISR. The first metric is *the packet delivery ratio*, which is defined as the percentage of data packets delivered to their destination nodes of those sent by the source nodes. The second metric is the *routing overhead* of both protocols, which is defined as the number of routing packets "transmitted" per each data packet "delivered." On multi-hop routes, each transmission of the routing packets is counted as one transmission. The third metric is the *average end-to-end delay* of the data packets. We used eleven pause time values (0, 50, 100, 150, 200, 250, 300, 350, 400, 450, and 500 s) to differ the mobility level (where 0 s pause time representing a continually moving nodes and 500 s representing stationary nodes).

5.2.1. Packet Delivery Ratio

VPR, in general, has a better delivery ratio than ISR (see Figure 2). VPR delivered an average of 2.32% of the data packets sent higher than ISR. The difference in the delivery ratio between both protocols is considerable at the high level of mobility (where the pause times are 0 s and 50 s). With pause time 0 s and 50 s, VPR delivered 13.57% and 8.44% higher than ISR, respectively. The mobility monitoring technique is the explanation behind these results. When ISR discovered and cached routes, it did not associate an expiration time with those routes according to the IRS specifications. However, the mobility of the network nodes invalidates those routes, which results in ISR trying to utilize outdated routes. On the other hand, VPR expires the cached routes based on its monitoring of mobility level around the node. The higher the level of mobility is, the shorter the expiration time of the cached routes would be. That technique prevents the usage and propagation of stale routes, which explains the better performance of VPR.

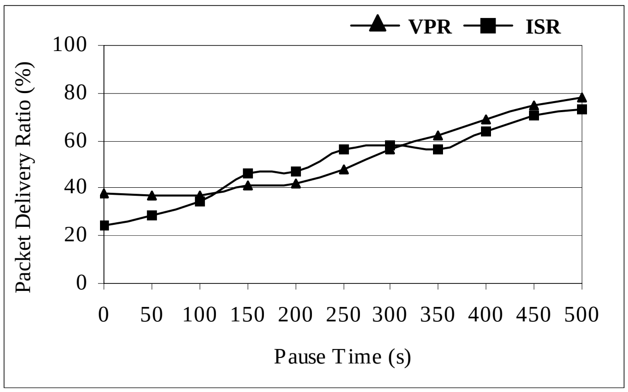

Figure 2: Packet Delivery Ratio.

5.2.2. Overhead

VPR also outperformed ISR in the routing overhead metric (see Figure 3). ISR incurred an average of 5.94 overhead packets per data packet higher than VPR. We found that the higher the level of mobility, the higher the difference in overhead between VPR and ISR. At the highest level of mobility, ISR incurred about 19.87 of overhead packets per data packet while VPR incurred about 1.29 overhead packets per data packet. However, the difference in the overhead incurred by both protocols is insignificant at low mobility levels.

The reason for such a high overhead is the additional Route Discoveries incurred by ISR through its salvaging process. The mobilities of the nodes invalidate the cached routes that were collected by the protocols. While VPR uses its mobility monitoring technique to limit the validity of a cached route, ISR invalidates a route when it receives a route error only. When ISR uses these invalid routes, the node that recognizes the broken link on a route will attempt to salvage the data packets it had received on that route, and usually uses the Path Discovery Process. If a node uses multiple invalid routes for the same destination, that results in multiple nodes initiating the Route Discovery Process for the same destination at the same time in order to salvage the data packets they had received on the broken route. Path Discoveries are proven very expensive in terms of overhead [13].

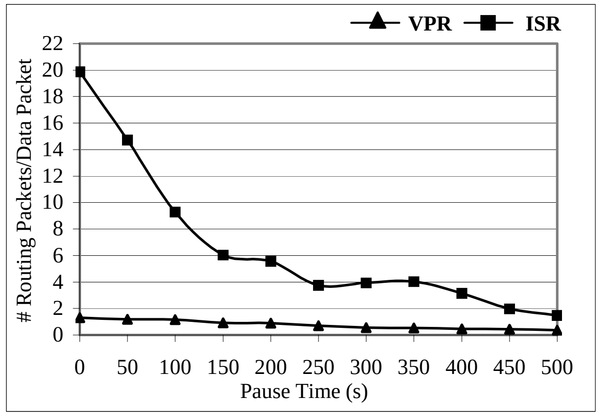

Figure 3: Overhead.

5.2.3. Average End-to-End Delay

VPR surpassed ISR in the average end-to-end delay metric (see Figure 4). The average end-to-end delay of VPR was about three second less than that of ISR. At the highest level of mobility (0 s pause time), in which nodes are moving constantly, the average end-to-end delays for VPR and ISR were 2.44 s and 7.14 s, respectively. Generally, the average of the end-to-end delay of VPR is less than that of ISR. We found the reason for that is the high overhead traffic generated by ISR. High overhead traffic congests the network and causes a long delay for the data packets. The reason for high overhead traffic was discussed in Section 5.2.2.

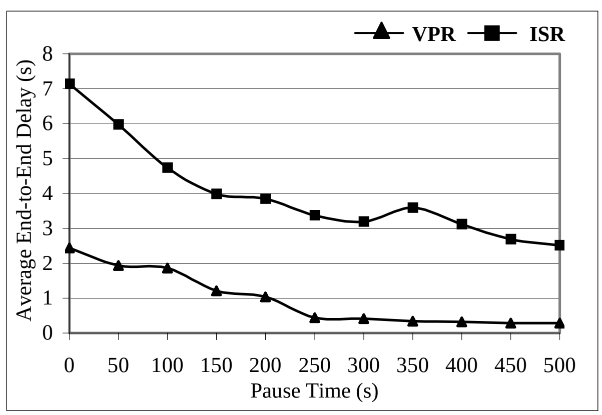

Figure 4: Average End-to-End Delay.

6. Conclusion and Future Work

In this paper, we featured the Virtual Paths Routing (VPR) Protocol for Ad Hoc Wireless Networks. The key purpose of VPR [8] is to provide correct, efficient, and highly adaptive route creation and maintenance among mobile nodes with multiple path selection criteria. We enhanced VPR by adding more route selection criteria to obtain load and mobility aware routes with a minimal change to VPR. More selection criteria will be added in the future. We also found that the mobility monitoring technique of VPR is effective in preventing the protocol from using invalid cached network topology.

Acknowledgment

The authors would like to thank Ann Jones for her assistance in compiling this paper.

References

[1] C. Perkins and P. Bhagwat, Highly Dynamic Destination Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, In *Proceedings of the Conference on Communications Architectures, Protocol and Applications (SIGCOMM '94)*, London, United Kingdom, August 1994, 234-244.

[2] R. V. Boppana and Satyadeva, Adaptive Distance Vector Routing Algorithm for Mobile Ad Hoc Networks, In *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (InFOCOM2001)*, Anchorage, Alaska, USA, April 2001, 1753-1762.

[3] C. E. Perkins, E. M. Royer, and S. Das, Ad hoc On-Demand Distance Vector (AODV) Routing, Internet Draft, draft-ietf-manet-aodv-07.txt, November 2000.

[4] D. B. Johnson, D. A. Maltz, Yih-Chun Hu, and J. G. Jetcheva, The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks, Internet Draft, draft-ietf-manet-dsr-04.txt, November 2000.

[5] D. B. Johnson, Routing in Ad Hoc Networks of Mobile Hosts, In *Proceedings of the Workshop on Mobile Computing Systems and Applications (WMCSA '94)*, Santa Cruz, California, USA, December 1994, 158-163.

[6] D. B. Johnson and D. A. Maltz, Dynamic Source Routing in Ad Hoc Wireless Networks, in T. Imielinski and H. Korth (Ed.) *Mobile Computing*, (Kluwer Academic Publishers, 1996) Ch 5, 153-181.

[7] Yih-Chun Hu and D. B. Johnson, Implicit Source Routes for On-Demand Ad Hoc Network Routing, In *Proceedings of The ACM Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc 2001)*, Long Beach, California, USA, October 2001, 1-10.

[8] A. Altalhi, and G. Richard III, Virtual Paths Routing: A Highly Dynamic and Adaptive Routing Protocol for Ad Hoc Wireless Networks. Submitted for publication. Available from http://www.cs.uno.edu/~aaltalhi.

[9] K. Fall and K. Varadhan, *The ns Manual*, The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, April 14, 2002. Available from http://www.isi.edu/nsnam/ns/.

[10] IEEE Computer Society LAN/MAN Standards Committee, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std. 802.11-1997* (New York: IEEE, 1997).

[11] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks, In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99)*, Seattle, Washington, USA, August 1999, 195–206.

[12] J. Broch, D. A. Maltz, D. B. Johnson, Yih-Chun Hu, and J. G. Jetcheva, A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols, In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98)*, Dallas, Texas, USA, October 1998, 85–97.

[13] S. R. Das, C. E. Perkins, and E. M. Royer, Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks, In *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (InFOCOM 2000)*, Tell Aviv, Israel, 2000, 3–12.