TCP Performance over Proactive and Reactive Routing Protocols for Mobile Ad Hoc Networks

M. Mahbubur Rahman[1], Md. Asif Nashiry[1], T K Godder[1] and A B M Shawkat Ali[2]

[1]Department of Information & Communication Engineering, Islamic University, Kushtia 7003, Bangladesh.
[2]School of Computing Sciences, Central Queensland University, QLD, Australia.

Abstract- *In this paper, we investigate the performance of TCP over a proactive and a reactive routing protocol for mobile ad hoc networks. For this investigation we choose DSDV for the proactive side and AODV for the reactive one. We use ns-2 to evaluate the TCP window size, throughput, packet delay and routing overhead over a single TCP connection. We also use a different metric, expected throughput, for the comparison of throughput of the routing protocols. Our observation shows that the chosen metrics are closely related and the TCP performance is heavily dependent on the routing protocol. It also shows that the TCP performs better over the reactive one when mobility is high.*

Keywords: MANET, TCP, Proactive, Reactive, Mobility.

1 Introduction

TCP/IP is the standard network protocol stack on the Internet. For this reason its use over mobile ad hoc network is a certainty. The Transmission Control Protocol (TCP) was designed to provide reliable end-to-end delivery of data over wired networks. It performs very well in a network of fixed topology. But in MANET [1], network topology may changes rapidly due to node mobility and variability of link quality. Therefore, TCP does not perform well in a typical ad hoc network.

All TCP versions assume that packet losses are occurred due to congestion. Consequently, when a packet is detected to be lost either by timeout or by multiple duplicated acknowledgements (ACKs), TCP slows down the sending rate by adjusting its congestion window. Unfortunately, wireless networks suffer from several types of losses that are not related to congestion, making the performance of TCP poor to this environment [2]. Thus improving TCP performance is still an active and interesting area of research.

Routing is an activity or a function that is related to the exchange of information from origin to destination in telecommunication networks. The concept of routing protocol is basically two folds: determination of optimal routing paths and transformation of packets through an internetwork. Since a node does not have previous knowledge of the network topology, it has to discover the path towards the destination. For achieving this, a new node announces its presence and listens to broadcast announcements from its neighbors. As time goes on, each node becomes aware of all other nodes and one or more ways to reach them by maintaining and updating a routing table. To perform efficiently and effectively a routing protocol must choose the optimal route for given destination and also has to converge within an exchange of a small amount of messages. It also should keep the routing table up-to-date and reasonably small.

TCP performance is not as stable as in wired networks because various wireless ad hoc network characteristics not found in wired networks. Among them, frequent link breakage due to mobility is one of the major factors degrading TCP performance. So improving TCP performance over ad hoc network is a challenging task. A number of protocols have been developed to accomplish this task. Depending on flat routing network structure some of them are proactive (DSDV, OLSR, FSR) protocols, where each and every node maintains routing information even before it is needed and others are reactive (AODV, DSR, TORA) routing protocols, which find a route to a destination on demand. Although there are many proposed protocols, only a few attempts have been made to compare their performance in a realistic manner. This paper provides a realistic and quantitative comparison of the TCP performance over DSDV (proactive) and AODV (reactive) routing protocols.

The reminder of this paper is organized as follows: Section 2 depicts the related works; mechanisms of routing protocols are shown in Section 3. We define the simulation model and present our analytical results in Section 4. The conclusion and future work follow in Section 5.

2 Related works

Since TCP is a reliable protocol used in the Internet, its performance in MANET has become an interesting and active area of research. Several performance evaluations of MANET routing protocols for TCP traffic have been presented in the literature [3-6]. However, earlier research suggested that TCP performance is poor in MANET [7]. This is because the packet loss due to node mobility and wireless link are erroneously treated as

congestion induced, which triggers an inappropriate response by TCP.

Various types of network layer feedback mechanism have been proposed, such as *TCP-F, ELFN, ATCP, TCP-BUS*, that rely on the intermediate nodes, where the route failures are detected and send some control messages to the TCP sender. Holland et al. advocates the use of explicit link failure notification (*ELFN*) to significantly improve TCP performance in MANETs [7]. Chandran et al. proposed a feedback-based scheme called TCPFeedback or *TCP-F* [8]. In this scheme, when an intermediate node detects the disruption of a route due to the mobility of the next host along that route, it explicitly sends a Route Failure Notification (RFN) to the TCP sender. Upon receiving the RFN, the source suspends all packet transmissions and freezes its state, including the retransmission time out interval and the congestion window. When an intermediate node learns of a new route to the destination, it sends a Route Reestablishment Notification (RRN) to the source, which then restores its previous state and resumes transmission. The effect of this scheme was studied by simulating a single TCP connection. The main conclusion of the study was that average route repair time has a major impact on TCP throughput.

The difference between *TCP-F* and *ELFN* is the response to route failures: *TCP-F* relies on intermediate nodes to send a route re-establishment notification to notify the sender that the path is restored; In *ELFN*, the TCP sender needs to send probing packets periodically to detect the route recovery.

Transport layer feedbacks are also developed to detect route disruptions by looking at the timing and sequence information of TCP packets. Two well known approaches to employ this are: the consecutive timeouts heuristic and *TCP-DOOR* [9]. *Fixed-RTO [10]* can achieve TCP throughput comparable to that of the *ELFN* mechanism.

As stated above, most related earlier works focus on devising mechanisms for TCP to detect route failures or link breakage and react to them accordingly. In this paper, we evaluate TCP performance over DSDV and AODV routing protocols in terms of metrics such as TCP window size, throughput, end-to-end packet delay and routing overhead. We also use our expected throughput metric to compare their performance over different scenarios. We show that these metrics are tightly related. Performance of throughput varies proportionally to the window size, whereas delay performance is inversely proportional to the throughput. For analyzing the TCP window size, we considered combined slow-start with congestion avoidance algorithm [11]. Thus, our work complements previous work and can be combined to help TCP achieve better performance in mobile ad hoc networks.

3 Routing protocols in MANET

Ad hoc routing protocols can be classified in many different ways. In terms of scheduling, that is, obtaining a route to forward packets to given destinations, routing protocols can be classified as proactive or reactive.

Proactive: Most of the conventional routing protocols are proactive or table driven, such as Distance Vector, Link State. In proactive routing, each and every node maintains routing information to every other node in the network. Route information is generally kept in the routing tables and is periodically updated as the network topology changes. This is the advantage of minimizing the delay in obtaining a route when initiating traffic to a destination and quickly determining whether a destination is reachable. But this process requires a lot of network resources. Moreover, these routing protocols maintain different number of tables. The proactive protocols are not suitable for larger networks, as they need to maintain node entries for each and every node in the routing table of every node. This causes more overheads in the routing table leading to consumption of more bandwidth.

Reactive: These protocols are also called On Demand protocols since they do not maintain routing information or routing activity at the network nodes if there is no communication. Each node tries to reduce routing overhead by sending routing packets when a communication is needed. Reactive algorithms typically have a route discovery phase. Query/response packets are flooded by the source into the networks for searching of a path. This phase completes when a route is found or all the possible outgoing paths from the source are search. These protocols are proposed to overcome some of the problem of proactive routing protocols.

TCP responds to packet loss by reducing window size, which unnecessarily reduces throughput. When link failures occur in MANETs, it is the responsibility for the routing protocol to detect and restore routing paths. In this paper, we consider the DSDV and AODV routing protocols as the representative of proactive and reactive protocols respectively. The key features of these two are briefly described next.

DSDV: The DSDV [12] is a variation of the distance vector routing protocol modified for ad hoc networks. The key advantage of DSDV over traditional distance vector protocols is that it guarantees loop-freedom. DSDV is a hop by hop routing and pro-active protocol that provides each node a routing table that lists the next-hop information for each reachable destination. Thus, it requires periodic broadcasting of routing updates and triggered beacon messages, which leads to an increase in routing overhead. A sequence number is used to tag each route, where a higher sequence number indicates a more updated route. Between two routes with the same

sequence number, the one with fewer hops is more favorable. If a node detects that route failure to a destination, its hop number is set to infinity and its sequence number is assigned an odd number: even numbers are assigned to connected paths.

AODV: The AODV [13] is a reactive protocol, which combines both DSR and DSDV characteristics. DSR [14] is another reactive protocol, composed of two on-demand mechanisms: Route Discovery and Route Maintenance. AODV borrows the basic route discovery and route-maintenance of DSR as well as hop-by-hop routing, sequence numbers and beacons of DSDV. When a source node desires to establish a communication session, it initiates a route discovery process to locate the destination node, by generating a "route request" message, which might be replied by the intermediate nodes in the path to destination or the destination node itself. At the time of arrival, the "route reply" message contains the whole path to destination. To handle the case in which a route does not exist or the query or reply packets are lost, the source node rebroadcasts the query packet if no reply is received by the source after a time-out. Failure of a link can be detected via hello messages or link layer detection. Failure to receive three consecutive HELLO messages from a neighbor is taken as an indication that the link to the neighbor in question is down. When a link fails, the upstream nodes are notified of the failure and the destination is marked as unreachable in the routing tables of these nodes.

4 Simulation model and performance results

4.1 Simulation model

For analyzing the performance of TCP over proposed protocols, we used ns-2 (Network Simulator 2) [15] with CMU wireless extensions. We consider three network scenarios. At first, we consider a chain topology, denoted as **A**, for our comparison requirement. Next, we consider another network of seven nodes, where the source and destination are stationary but all intermediate nodes move at 5 m/s. We denote this topology as **B**. Finally, we consider topology **C**, similar to **B** but all nodes including source and destination are mobile. The initial arrangement of nodes for topology **B** and **C** is shown in Fig. 1.

The nodes ladled with S, D and R denote as source, destination and router in the following figure. The arrows indicate the direction of movements of intermediate nodes. The effective communication range of each node is 250 meters. We use TCP NewReno and FTP begins transferring packets of size 1000 bytes after 10 second. Each wireless node has a buffer size of 50 packets and its raw radio link capacity is 2 Mbps. Each simulation runs 150 seconds.

4.2 Performance result

4.2.1 Topology A

Our observations start with a chain topology of four nodes, as shown in Fig. 1. The distance between two neighboring nodes is 150 meters in the 500x500 topology and all nodes are stationary.

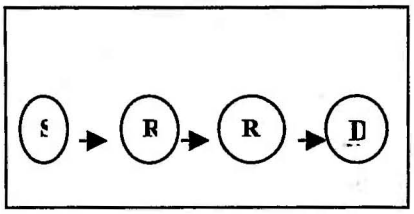

Fig: 1 Chain Topology

Radio channels are bi-directional; the arrows indicate the direction of data packet transmissions. ACKs travel in the opposite direction.

4.2.1.1 Performance in Window Size

According to the combination of slow-start and congestion avoidance TCP algorithm, if W refers to the window size, then for each successfully received packet at the destination $W = W + 1/W$. This process is repeated until the connection breaks up. For each packet dropped due to the change in routing path, the window size is decreased to half the current window value. The evolution of TCP window size with time, obtained with each routing protocol is shown in Fig. 2.

It is seen from figure 2, that AODV established the TCP connection very quickly. But, DSDV requires a longer time to establish the connection. Because it is proactive in nature, thus require all the valid routing information to reach packet to destination. Hence, it is slow and which is not desirable in a typical ad hoc network

According to Fig. 2, we observe that after connection is established with both algorithms, the window size increases monotonously since the same routing path is used. In terms of connection establishment time and size of window, AODV performs better than DSDV.

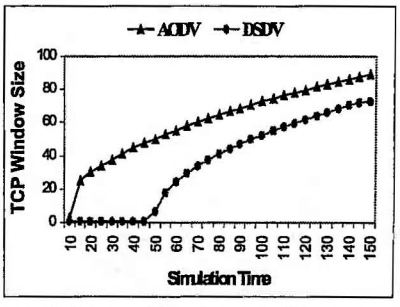

Fig: 2 TCP Window Size for AODV and DSDV for topology A

4.2.1.2 Throughput performance

We observe the throughput obtained with each routing protocol. Throughput is evaluated in terms of the total number of packets received at the destination node per unit time over intervals of 5 seconds. The results are depicted in Fig. 3.

According to Fig. 2 and Fig. 3, we observe that throughput is very tightly coupled with window size. Throughput increases as the size of the window increases. Due to a larger connection establish time, DSDV provides zero throughput from approximately 10 to 45 sec, whereas AODV maintains a constant throughput throughout the connection. At, some later moment both protocols provide same throughput, since the same number of hops (3 hops) is used.

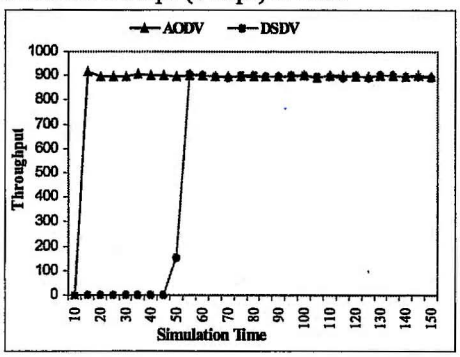

Fig: 3 TCP throughputs for both routing protocols for topology A

Expected throughput: TCP performance in ad hoc network is poor for taking inappropriate action when packets are lost due to link breakage. In MANET, link breakage may occur due to mobility. For being mobile, a node may goes far away from the transmission range of its neighbors, which may result in throughput degradation. Topology **A** guarantees link breakage freedom due to mobility, since it is static in nature and neighboring nodes always within their communication range. But in our other two topologies some or all nodes are mobile. Hence, we take the throughput of topology **A** as our expected throughput for topology **B** and **C** for comparison requirement. We obviously take the AODV throughput of topology **A** as expected throughput, because it performs better than DSDV as we seen from Fig. 3.

4.2.2 Topology B

The network scenario for topology **B** is shown in Fig. 4. In this topology, we considered a network of seven nodes. This network consists of stationary source and destination nodes while all intermediate nodes move at 5 m/s. Moving directions of intermediate nodes are selected randomly, and when nodes reach their defined destination, they bounce back and continue to move.

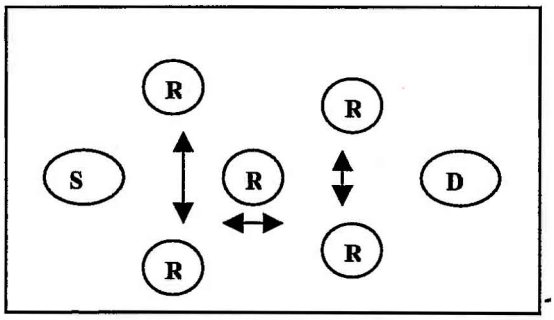

Fig: 4 500x500m Network scenario for topology B

4.2.2.1 Performance in Window Size

The evolution of TCP window size with respect to time, obtained with each routing protocol for topology **B** is shown in Fig.5. We observe that after establishing connection, AODV maintains the monotonically increasing window for a long time, since it maintains the same route during that time.

Proactive, DSDV does not adapt well in this environment. Window size is reduced for several times during connection period. Moreover, it requires longer time to re-establish the connection. The maximum window sizes for AODV and DSDV are 82 and 50, respectively. Comparing with DSDV, AODV retains the communication path for longer period.

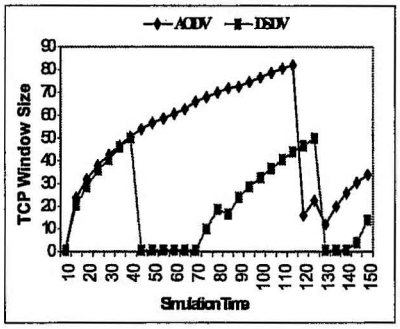

Fig: 5 TCP Window Size for AODV and DSDV for topology B

4.2.2.2 Throughput performance

For topology **B**, we evaluate the throughput of AODV and DSDV over time and compare the results with the expected throughput. The result is given in Fig. 6. After establishing the connection both protocols achieve throughput higher than the expected throughput. This is because the packets are forwarded to the destination through 2 hops at that time, but expected throughput was measured under 3 hops. TCP throughput decreases as the number of hop increases. After then the route changes and AODV finds another route quickly to forward packets to destination. But DSDV fails to do so quickly and provides no throughput for a while.

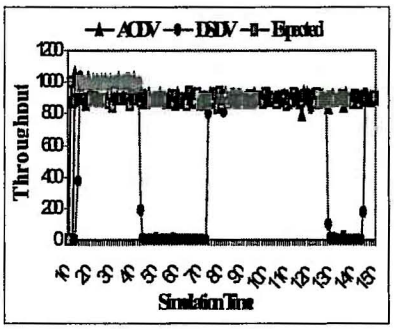

Fig: 6 TCP throughput for AODV and DSDV for topology B with expected throughput

According to the Fig. 6 AODV performs well and very much similar to our expected throughput, which is not the case for DSDV.

4.2.3 Topology C

In this case, we take the same network scenario as in topology **B**, but here, all nodes including source and destination move at various speeds (0,5,10,15,20,25,30,35 m/s).

4.2.3.1 Throughput performance

We evaluate TCP throughput when all nodes move at 5 m/s and obtain the similar result as previous.

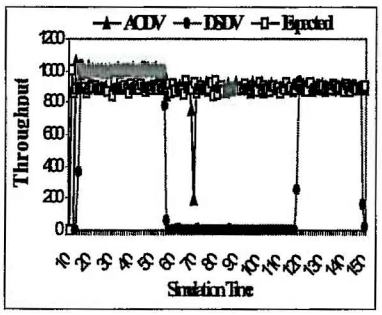

Fig: 7 TCP throughput for AODV and DSDV for topology C with expected throughput

According to Fig. 7, the connection time for DSDV is short. At time 58 to 118 sec, it provides no throughput, because of the lack of adaptability of proactive protocols. The performance of AODV almost satisfies our expectation.

Thus, from Fig. 6 and Fig. 7, we observe that nodes mobility play a significance role in the throughput performance. Now we will observe how various speeds of nodes affect the TCP performance.

4.2.3.2 Average throughput performance

For topology **C**, we evaluate the average throughput over simulation duration with respect to node mobility and depict the result in Fig. 8.

Both protocols achieve almost the same throughput when nodes are stationary. But as node mobility increases the performance of DSDV if degraded. It performs reasonably well at low mobility. But AODV maintains a constant throughput under high mobility (approximately 225 Kbps)

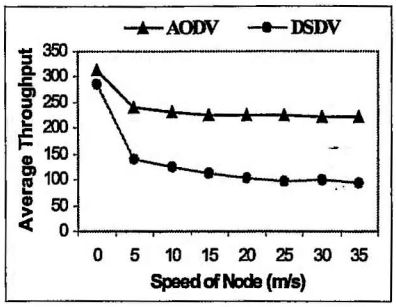

Fig: 8 Average throughputs with respect to node mobility for topology C

4.2.3.3 Average Delay Performance

For topology **C**, we evaluate the average end-to-end delay over connection establishment duration with respect to node mobility and depict the result in Fig. 9. Proactive DSDV shows the low delay activities, though its throughput is lower, than reactive AODV. It is because DSDV requires a long time to establish connection. Moreover, in DSDV, every node maintains the routing information, which minimizes delay.

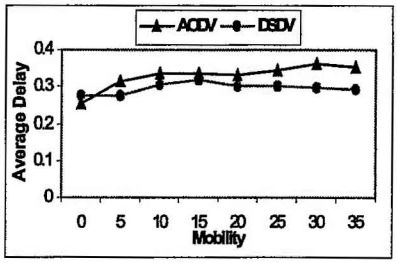

Fig: 9 Average delay (sec) versus Mobility (m/s) for topology C

4.2.3.4 Routing overhead performance

For topology **C**, we calculate the number of routing packets that the protocols maintain to manage the connection and data exchange with respect to node mobility and the result is shown in Fig. 10. We did not include IEEE 802.11 MAC packets or ARP packets in routing overhead. Protocols that send large numbers of routing packets can also increase the probability of packet collisions and may delay data packets in network interface transmission queues. According to figure 10, routing overhead of DSDV is much lager than AODV. Due to be a proactive algorithm, DSDV periodically broadcasts routing information for updates. Bandwidth is wasted when nodes are stationary.

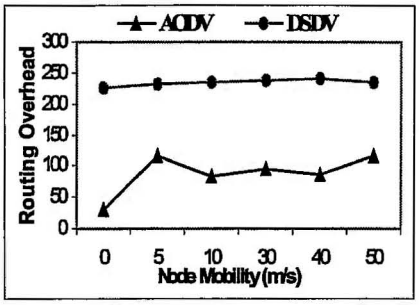

Fig: 10 Routing packet overhead for AODV and DSDV for topology C

AODV on the other hand is on demand in nature. Thus it transmits routing information as "when needed" basis. In AODV, nodes only transmit hello messages to maintain connectivity between neighbors.

4.3 Effect of node mobility

In this Section, we considered the impact of node mobility on TCP performance. We studied the effect of mobility on performance by varying the speed of network hosts. Routing protocols that adapt sufficiently quickly (e.g. AODV) are able to switch to a routing path with fewer hops, giving rise to an increase (decrease) in throughput (delay) as the node speed increases. We have seen from the above performance result that as mobility of nodes increases throughput, delay, routing overhead performance decrease. This is because node mobility during TCP transmission causes link failures, giving rise to degradation in window size, throughput and delay performance until a new route is formed. When mobility speed increases, the bi-directional link assumption in ad hoc networks becomes weak (i.e., a node can reach a neighboring node, but not necessarily vice versa). By using location and mobility information route expiration time can be estimated and receivers can select the path that will remain valid for the longest time. With the mobility prediction method, sources can reconstruct routes in anticipation of route breaks. This way, the protocol becomes more resilient to mobility. Moreover, multi-path protocols can also be used in order to improve TCP performance.

5 Conclusion and Future work

The ability of communicating devices to form a network in the absence of communication infrastructure is an active area of research. TCP throughput drops significantly when node movement causes link failures, due to TCP's inability to recognize the difference between link failure and congestion. But TCP performance can be improved by selecting a suitable routing protocol. In this paper, we investigate the performance of TCP over DSDV (proactive) and AODV (reactive) protocols using simulations in *ns*-2 for a range of node mobility with a single traffic source. The performance metrics that we considered includes TCP window size, throughput, packet delay and routing overhead. We also use a different metric, expected throughput, for providing a more accurate performance comparison. Our study shows that proactive one consumes more bandwidth, because of transmitting routing updates frequently. It reacts slowly for dynamically changing topologies. Its performance decreases drastically as mobility increases. But the reactive one consumes less bandwidth and lower overhead of routing information. It also provides almost the same throughput as our expectation. To resist against the performance degradation of TCP under high mobility, it is however necessary to have some sort of feedback from link layer protocol. More research is needed to better understand the complex interactions between TCP and MANET. We plan to investigate TCP performance of routing protocols with multiple traffic sources in the future.

References

[1] M. Frodigh, P. Johansson, and P. Larsson. "Wireless ad hoc networking: the art of networking without a network" *Ericsson Review*, No.4, 2000

[2] Fernando Tapia. "TCP over mobile ad hoc network", May 2004

[3] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proc. ACM/IEEE MobiCom*, pages 85–97, October 1998.

[4] Samir R. Das, Robert Casta˜neda, and Jiangtao Yan. Simulation based performance evaluation of mobile, ad hoc network routing protocols. In ACM/Baltzer Mobile Networks and Applications (MONET) Journal [44], pages 179–189. July 2000

[5] Krishna Gorantala. "Routing protocols in mobile ad-hoc networks" June 2006

[6] Hong X, Xu K, Gerla M. *Scalable Routing Protocols for Mobile Ad Hoc Networks*,IEEE Network, July/August 2002.

[7] G. Holland and N. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," *Proceedings of ACM Mobicom*, pp. 219–230, August 1999.

[8] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "A feedback-based scheme for improving TCP performance in ad hoc wireless networks," *Proceedings of IEEE ICDCS*, pp. 472–479, February 1998.

[9] F. Wang and Y. Zhang, "Improving TCP performance over mobile ad hoc networks with out-of-order detection and response," *Proceedings of ACM Mobihoc*, pp. 217–225, June 2002.

[10] T. D. Dyer and R. V. Boppana, "A comparison of TCP performance over three routing protocols for mobile ad hoc networks," *Proceedings of ACM Mobihoc*, pp. 56–66, October 2001.

[11] V. Jacobson, "Congestion avoidance and control," *Proceedings of ACM SIGCOMM*, pp. 314–329, August 1988.

[12] Charles E. Perkins and Pravin Bhagwat. Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. In *Proc. ACM SIGCOMM Conference (SIGCOMM '94)*, pages 234–244, August 1993.

[13] Charles E. Perkins and Elizabeth M. Royer. Ad hoc On-Demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, February 1999, pp. 90-100.

[14] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, chapter 5, pages 153–181. Kluwer Academic Publishers, 1996.

[15] K. Fall and K. Varadhan, "NS notes and documentation," *The VINT Project*, February, 2005.