Robust Routing in Wireless Ad Hoc Networks

Seungjoon Lee, Bohyung Han, Minho Shin
{slee, bhhan, mhshin}@cs.umd.edu
Computer Science Department
University of Maryland
College Park, MD 20742 USA

Abstract

A wireless ad hoc network is a collection of mobile nodes with no fixed infrastructure. The absence of central authorization facility in dynamic and distributed environment requires collaboration among nodes. When a source searches for a route to a destination, an intermediate node can reply with its cached entry. To strengthen correctness of such routing discovery process, we propose a method in which the intermediate node requests its next hop to send a confirmation message to the source. After receiving both route reply and confirmation message, the source determines the validity of path according to its policy. As a result, this strategy discourages malicious nodes from intercepting packets. Simulation results show remarkable improvement in throughput (30% higher delivery ratio and 10% less data transmission overhead) with moderate increase of control messages.

1. Introduction

The advance of mobile device technology leads to wide use of wireless network. Recently, as laptops and PDAs are smaller and cheaper, the wireless communication is becoming popular. Even though such mobile devices have improved, they still have some restrictions such as small memory, limited CPU, and exhaustible battery. Therefore, they are inadequate for resource-demanding operations.

A wireless ad hoc network is a collection of such mobile nodes that do not rely on the predefined infrastructure. There is no administrative node to control the network, and every node participating in the network is responsible for the reliable operation of overall network. Since each node is free to move around, network topology frequently changes. Moreover, it uses open transmission medium, and every node within the range can access it. In this infrastructureless environment, each node in ad hoc networks acts as a router to establish end-to-end connections. However, due to volatile network topology and limited resource in mobile nodes, routing in ad hoc networks is a challenging problem. Also, since bandwidth is a scarce resource in wireless environment, routing efficiency is more critical in ad hoc networks.

Because there is no administrative node in wireless ad hoc networks, most network algorithms are based on the collaboration between nodes. In order to cooperate with each other, trust between nodes is essential, but it is hard to achieve in practice. So the wireless ad hoc network is inherently vulnerable. On the other hand, the transmission medium itself necessitates security in wireless ad hoc networks. For example, suppose that a node needs routing information to transmit data. In many ad hoc network routing protocols, the source node broadcasts the routing request and receives the routing reply from the destination. However, while all the information is delivered through many hops, it can be eavesdropped, forged, or dropped during the transmission. Therefore, we need the security consideration in the wireless ad hoc network.

There are many research efforts to overcome the vulnerability in wireless ad hoc networks such as security in the routing protocol, authentication and authorization problem, intrusion detection, and so on [3, 8, 12, 18, 19]. Among many problems in the wireless environment, we concentrate on the routing robustness in this paper. In many on-demand ad hoc routing protocols [6, 15], intermediate nodes can answer route discovery request from the source if they have a route to the destination in their route caches. However, it is possible for a node to forge the route reply message so that it may accomplish its malicious attempt. From the above observation, it is obvious that malicious nodes can easily corrupt routing information, which may cause communication failure in the network.

In this paper, we present a method that detects such a routing misbehavior by making a neighbor of the replying node send a confirmation message to the source. In the simulation, our algorithm shows remarkable improvement of throughput with reasonable increase of control messages. Our algorithm shows as high delivery ratio as 80% even when 16% of nodes in the network are malicious, whereas delivery ratio of DSR is below 50%.

Also, our protocol reduces data transmission overhead by around 10% in the presence of malicious nodes, while additional control messages induce moderate increase in control overhead. Without malicious nodes, our protocol performs as well as existing routing protocols, with only about 5% control overhead increase.

This paper is organized as follows. In section 2, related work is discussed and our algorithm is explained in section 3. After that, the simulation methodology is provided in section 4, followed by simulation results and analysis in section 5. Finally, section 6 concludes the paper.

2. Related work

2.1. Routing in ad hoc networks

Unique characteristics of ad hoc networks raise several requirements for routing protocol design. Many routing schemes have been presented to provide adequate performance of ad hoc networks. These proposals are classified into *proactive* routing and *reactive* routing based on when routes are determined. Proactive routing continuously makes routing decisions so that routes are immediately available when packets need to be transmitted [1, 14]. Reactive routing determines routes on an as-needed basis: when a node has a packet to transmit, it queries the network for a route [6, 13, 15]. Previous study has shown that reactive routing protocols are better suited in ad hoc network environment than proactive ones [2]. In addition to unicast routing protocols, several multicast routing protocols for ad hoc networks have been proposed in recent years [9, 10, 16].

Dynamic Source Routing (DSR) is a reactive ad hoc unicast routing protocol, which uses source routing [6]. In DSR, if a source node has data to send and does not have a route to destination, it performs route discovery by broadcasting *route request* (RREQ). Any intermediate node that receives a non-duplicate RREQ appends its address to source route list in the RREQ message and re-broadcasts the packet. When receiving RREQ, destination node sends *route reply* (RREP) back to the source. In addition, any node in the network can cache routing information obtained from route discovery packets and data packets. And, intermediate nodes can reply to RREQ if a route to destination is stored in their caches.

As observed in DSR, caching route information is common in ad hoc network routing protocols to save packet transmission and reduce route acquisition latency [6, 15]. However, stale route information in cache leads to additional communication delay since data sent via incorrect route should be retransmitted after all. To prevent this, all routing information is maintained as *soft-state* – recently unused route entries are removed after pre-determined time. Note that deleted route entries in route cache can be interpreted as relatively obsolete and unreliable paths.

2.2. Security issues in ad hoc networks

Before two nodes communicate with each other, they need to know the identity of the other party at first. Since all nodes act as routers and routing information can come from any node in most routing protocols, they should be able to tell if originators of routing information are valid and trustworthy.

In wired networks, each node can distribute the public key to other nodes via a trusted entity called *certification authority* (CA). The secure CA is assumed to exist in the network and correct authentication is guaranteed by the proper operation of CA [5, 7]. However, these solutions do not apply to ad hoc networks. Dynamic network topology and lack of fixed infrastructure make the assumption of global CA unreasonable. Due to distributed nature of ad hoc networks, some researchers have proposed that information about the key be shared among several nodes and the *threshold cryptography* scheme be used [4, 17].

Several literatures address the authentication problems and suggest solutions in the wireless ad hoc environment. In [3], authors have demonstrated exploits that are possible against ad hoc routing protocols, and offered a solution with an authenticated routing protocol. They have pointed out the intrinsic problems in *Ad hoc On Demand Vector* (AODV) routing protocol and proposed *Authenticated Routing for Ad hoc Networks* (ARAN).

Some papers [8, 19] have dealt with the routing security problem in the ad hoc network and employed cryptographic schemes based on Public Key Infrastructure (PKI) to protect both routing information and data. Instead of a single CA in the network, the trust has been distributed to a set of nodes that share the key management responsibility by using threshold cryptography. In these papers, *threshold secret sharing* and *secret share update* enable the intrusion tolerance. Papadimitratos and Hass [12] assume that there is *security association* (SA) between the source and the destination. And then, they have presented a route discovery protocol that mitigates the detrimental effects of malicious nodes by using the shared secret key and SA.

To mitigate the misbehavior of malicious nodes dropping data packets, [11] has suggested two tools called *watchdog* and *pathrater*. The watchdog detects misbehaving node by monitoring neighbors and the pathrater locally rates the reliability of each node. Based on these two methods, authors try to achieve the robustness of networks.

3. Algorithm

3.1. Protocol description

We propose a method that strengthens the correctness of route information sent by intermediate nodes. Also, it helps source node filter out possibly stale route information. By intermediate nodes, we mean nodes that are on a path between source and destination. Our scheme requires only two types of additional control messages, and does not entail extraneous overhead, for example, operating in promiscuous mode. Our protocol is simple and interoperable with most on-demand routing protocols. And, many authentication methods [3, 8, 19] can be combined with our protocol.

In this paper, we only describe how our protocol operates with DSR. In addition to RREQ and RREP in DSR, our scheme also uses the following control packets: *Route Confirmation Request* (CREQ) and *Route Confirmation Reply* (CREP). On finding a route to the destination in its cache, an intermediate node sends RREP back to the source. At the same time, our protocol requires the intermediate node to send CREQ to its next hop node toward the destination. Then, after receiving CREQ, the next hop node looks up its cache for a route to the destination. If it has one, it sends CREP to the source with its route information. Then, the source is able to learn whether the path in RREP is valid by comparing the information with CREP. On the other hand, when the destination initiates RREP, CREQ and CREP are not necessary, since the destination should give correct route if it wishes to receive data packets.

Let us take an example. Figure 1 shows an example of ad hoc network. Suppose *S* wants to send data packets to *T* and has no route. Suppose further that intermediate node *C* has a route to *T* in its cache. To find a route, *S* sends RREQ. *B* receives and broadcasts this RREQ. When *C* receives RREQ from *B*, *C* finds it has a route to *T* and sends RREP back to *S* through *B*. In addition, *C* also sends CREQ to its next hop, say *D*, asking for validation of RREP. *D* sends CREP to *S* if it also knows a route to *T*; otherwise, it does nothing. *S* believes the path claimed by *C* only after receiving CREP from *D*.

If *S* does not receive CREP from *D* within a pre-determined amount of time, it believes that the path is less reliable, and uses other route for data transmission. If *S* receives CREP but the information from the two does not match, it can choose whether to use the path according to its policy.

3.2. Discussion

Our protocol discourages malicious nodes that try to advertise falsely good paths in order to hinder path finding procedures or intercept all data packets. For example, even though malicious node *C* tries to advertise a better path than it actually has, this attempt can be precluded since CREP from *D* will have different information. So our protocol can avoid *blackhole* attack, in which a malicious node advertises that it is on the shortest path to any particular destination and drops all packets.

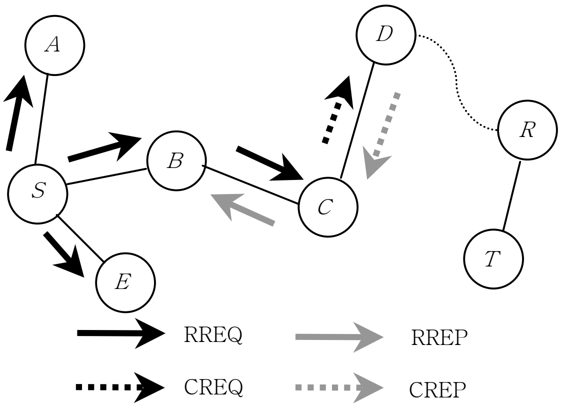

Figure 1. An example of ad hoc network

In addition, our protocol ensures robustness of a path. As mentioned in section 2, intermediate nodes can send RREP after finding a route to destination. Even though cache contents are regularly refreshed, there can be stale information and inconsistency between nodes. In our example, suppose that *C* and *D* both had a route to *T* in their caches and that the route in *D* is removed for some reason (e.g. timeout, route failure), while *C* still has it. Since route caches of *C* and *D* are inconsistent, the path is unlikely to be available. Hence, it is better to use a more reliable path if *S* has another. In our scheme, since *S* does not receive CREP from *D*, it will avoid the possibly stale path.

Although our protocol is not a node trust rating system, it is compatible to such notion and can be used as a basis scheme to identify non-conforming nodes. For example, suppose *C* does not want to forward packets from others (to save battery, etc.) and advertises worse route than it has. This attempt can be detected when *S* receives CREP from *D*. Based on this information, rating scheme can be devised.

On the other hand, two colluding nodes can circumvent our scheme. In our previous example, suppose *C* and *D* are malicious nodes and they are colluding. Even though *C* sends RREP with incorrect information, *D* will send CREP that supports incorrect

RREP from *C*, and *S* will think routing information from *C* is correct. In this paper, we only consider possibility of malicious nodes acting alone.

Another type of misbehavior happens when nodes agree to forward data packets but fail to do so. Watchdog and pathrater address this problem [11], and these strategies can be employed in our protocol.

4. Methodology

4.1. Simulation environment and parameters

Network Simulator 2 (ns-2) is used for simulation. ns-2 is originally developed by the VINT project [20], and later extended for ad hoc network simulation by the MONARCH project [21].

We generate several movement patterns and traffic scenarios, which are used as inputs to simulations. Each movement scenario file determines initial positions and subsequent movements of all nodes. Traffic scenario files specify source-destination pairs, and starting and ending times of each communication session, which are determined in an independent and random manner.

Our simulation consists of 50 mobile nodes, moving around over a flat rectangular region (300m × 1500m) for 200 seconds of simulated time. Transmission range is set to 250 meters. Random waypoint movement model is used and maximum movement speed is 10m/s. Packets among the nodes are transmitted with *constant bit rate* (CBR) of four packets per second, and there are ten pairs of source and destination. Four different traffic scenarios are generated.

In the simulation, we consider different parameters such as the number of malicious nodes, mobility and policy at source. We use four different numbers of malicious nodes (0, 2, 4 and 8), and four different pause times (10, 20, 30, and 40) for performance evaluation.

4.2. Behaviors of malicious nodes and sources

We assume malicious nodes attempt blackhole attack. They pretend to be a direct neighbor of destination and send RREP whenever they receive RREQ. When malicious nodes receive data packets, they drop all the packets.

When RREP is initiated by an intermediate node, source needs to employ a policy to accept RREP. It discards the path unless it receives CREP for RREP. When CREP arrives, source approves RREP depending on one of the following policies:

- EXACT: Source uses the route only if paths in PREP and CREP are identical.

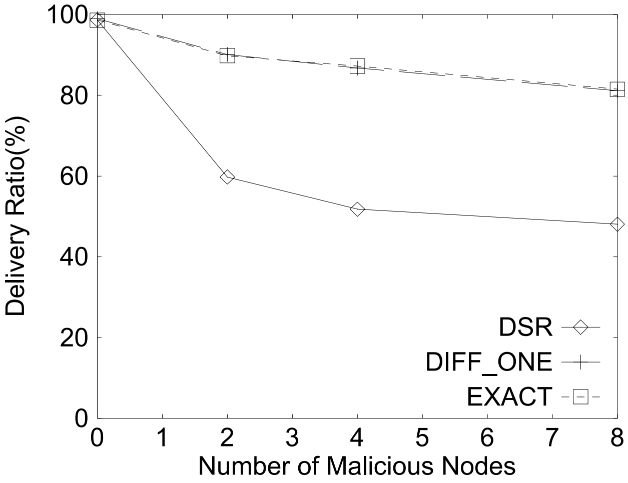

Figure 2. Packet delivery ratio with different number of malicious nodes

- DIFF_ONE: Source uses the route only if the difference between two hop counts is not more than one. The shorter path is preferred.

4.3. Evaluation metrics

The performance of our scheme is evaluated against DSR, using the following metrics:

- *Data packet delivery ratio*: The percentage of data packets delivered to destination with respect to the number of packets sent. This metric shows the reliability of data packet delivery.
- *Data transmission overhead*: The ratio of the number of packets sent or forwarded to the number of received packets at the destination. This metric reflects the efficiency of data packet delivery.
- *Control overhead*: The ratio of routing packets to delivered data packets.

5. Simulation results

Figure 2 shows delivery ratio as a function of the number of malicious nodes. Both DSR and our protocol perform well in case of no malicious nodes. In the presence of malicious nodes, however, the delivery ratio of DSR drops abruptly (about 40%), and it becomes worse as the number of malicious nodes increases. Our protocol exhibits 30% higher delivery ratio than DSR, and DIFF_ONE and EXACT have almost the same delivery ratio. In DSR, a malicious node replies to RREQ pretending a neighbor to destination, and deceives source into choosing the path in RREP as the shortest path to destination. Consequently, data packets sent through the path will not be delivered to destination. However, our protocol does not accept paths replied by malicious

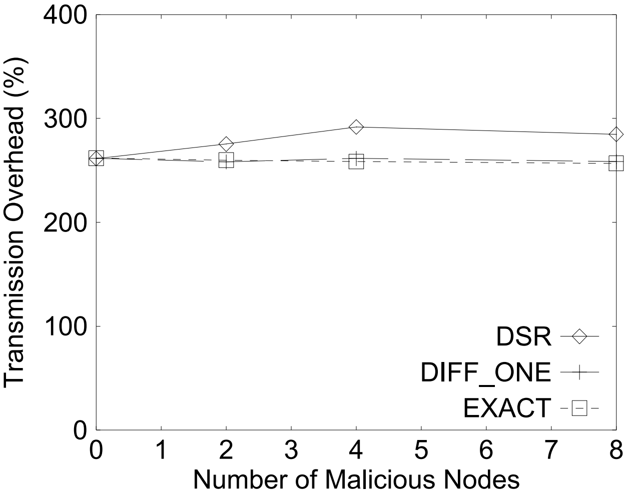

Figure 3. Data transmission overhead with different number of malicious nodes

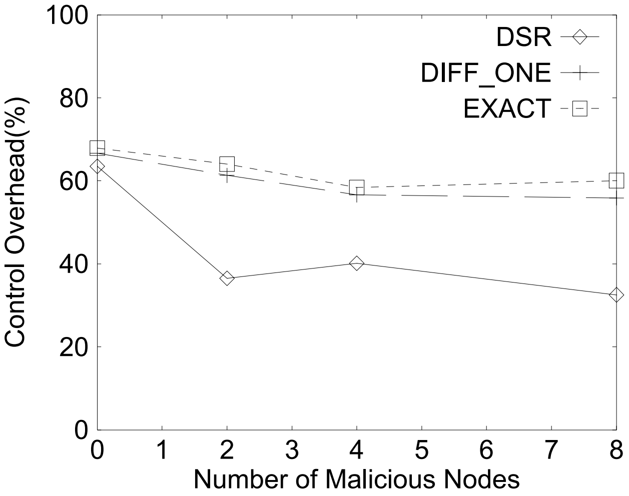

Figure 4. Control overhead with different number of malicious nodes

nodes since they are not confirmed. As a result, it maintains relatively high delivery ratio by avoiding such less reliable routes.

Data transmission overhead is shown in Figure 3. We can observe that our protocol has lower data transmission overhead than DSR by around 10%. As defined in section 4, data transmission overhead means number of sent and forwarded packets per received packet. Packets forwarded through incorrect path increase transmission overhead since data packets cannot be delivered to destination despite packet forwarding. In DSR, since source chooses the path based on RREP, it may use the path including malicious nodes, which cause packet loss and consequent increase in data transmissions overhead. On the other hand, the use of confirmation packets in our protocol precludes paths containing malicious nodes, reducing data transmission overhead as a consequence.

Even though our protocol introduces additional control packets (CREQ and CREP), extra control messages are kept minimal as illustrated in Figure 4. When there is no malicious node, our protocol adds around 5% control overhead over DSR. With low control overhead in the presence of malicious nodes, DSR apparently performs in an efficient manner. However, low control overhead of DSR does not necessarily imply efficient routing for the following reason. When DSR uses a source route including a malicious node, data packets are not forwarded beyond the malicious node. As a result, link error along the path with malicious node is less likely than without malicious nodes. This implies that DSR fails to initiate necessary RREQs even when route maintenance should be performed. Note that reduced number of RREQs results in drastic decrease of subsequent control packets. Therefore, we claim that extra control overhead in our protocol is inevitable to provide adequate performance when there are malicious nodes in the network.

Lastly, we report the impact of mobility on the performance of our protocol. Figure 5 demonstrates delivery ratio as a function of pause time. Longer pause time means lower node mobility. As expected, packet delivery ratio of our protocol becomes higher with lower mobility. Moreover, our protocol outperforms DSR noticeably regardless of node mobility.

6. Conclusion and Future work

We proposed a scheme that strengthens robustness of routing information in ad hoc networks. It introduces additional route confirmation request and response messages, and is interoperable with most existing on-demand routing protocols. Simulation results validate the effectiveness of our protocol against blackhole attack. With malicious nodes, delivery ratio of our protocol stays as high as 80% while DSR delivers less than 50% of data packets sent. Data transmission overhead is also reduced by 10% compared to DSR, and in case of no malicious attempt, our protocol incurs only 5% additional control overhead.

We note that there is only small difference between EXACT and DIFF_ONE. However, in more volatile and dynamic environment, we believe that the policy at

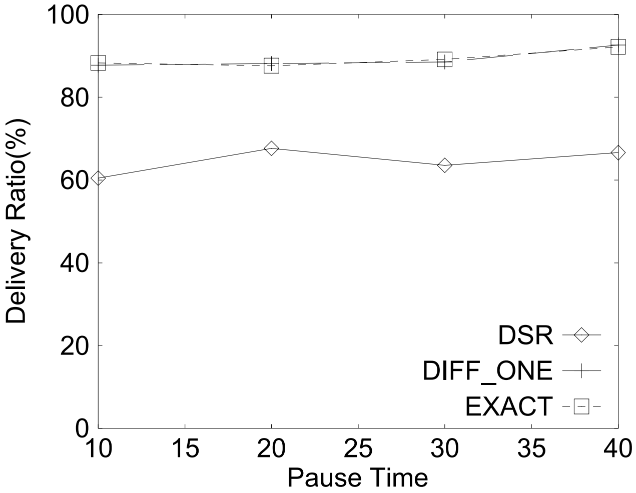

Figure 5. Packet delivery ratio as node mobility changes

source leads to considerable difference in overall performance. We plan to explore how different policies influence on routing performance and robustness. Also, we will examine the behavior of our protocol with other on-demand routing protocols such as AODV.

References

[1] D. Bertsekas, and R. Gallager, "Data Network," second edition, *Prentice-Hall*, pp. 404-410, 1992.

[2] J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 85-97, Oct 1998.

[3] B. Dahill, B. Levine, E. Royer, and C. Shields, "A secure routing protocol for ad hoc networks," *IEEE Network Magazine*, vol. 13, no.6, Nov/Dec 1999.

[4] Y. Desmedt, "Threshold cryptography," *European Transactions on Telecommunications*, 5(4), pp 449-457, July-Aug, 1994.

[5] M. Gasser, A. Goldstein, C. Kaufman, and B. Lampson, "The digital distributed systems security architecture," *Proceedings of the 1989 National Computer Security Conference*, pp. 305-319, 1989.

[6] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks," *Ad Hoc Networking*, edited by Charles E. Perkins, pp. 139-172, Addison-Wesley, 2001.

[7] C. Kaufman, "DASS: Distributed authentication security service," Request for Comments: 1507, Sep 1993.

[8] J. Kong, P. Zerfos, H. Luo, S. Lu and L. Zhang, "Providing robust and ubiquitous security support for mobile ad-hoc networks,", *International Conference on Network Protocols (ICNP)*, pp. 251-260, 2001.

[9] S. Lee, and C. Kim, "A New Wireless Ad hoc Multicast Routing Protocol ," *Computer Networks*, vol. 38, no. 2, pp. 121-135, Elsevier Science, Feb 2002.

[10] S.-J. Lee, W. Su, and M. Gerla, "On-demand multicast routing protocol in multihop wireless mobile networks," *ACM/Baltzer Mobile Networks and Applications*, 2000.

[11] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," *Proceeding of MOBICOM*, Aug 2000

[12] P. Papadimitratos, and Z. Hass, "Secure routing for mobile ad hoc networks," *IEEE Network Magazine*, vol. 13, no.6, Nov/Dec 1999.

[13] V. Park, and S. Corson, "Temporally-ordered routing algorithm (TORA)," ver. 1, Internet draft, IETF, Aug 1998.

[14] C. Perkins, and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," *ACM SIGCOMM 94 Conference on Communications Architectures, Protocols and Applications*, pp. 234-244, Oct 1994.

[15] C. Perkins, E. M. Royer, and S. R. Das, "Ad hoc on demand distance vector (AODV) routing," Internet draft, IETF, Nov 2001.

[16] E. M. Royer, and C. E. Perkins, "Multicast operation of the ad hoc on-demand distance vector routing protocol," *Proceedings of MOBICOM*, pp. 207-218, Aug 1999.

[17] A. Shamir, "How to share a secret," *Communication of the ACM*, 22, pp. 612-613, 1979.

[18] Y. Zhang, and W. Lee, "Intrusion detection in wireless ad hoc networks," *Proceeding of MOBICOM*, Aug 2000.

[19] L. Zhou, and Z. Haas, "Securing ad hoc networks," *IEEE Network Magazine*, vol. 13, no.6, Nov/Dec 1999.

[20] K. Fall, and K. Varadhan (Eds.), ns notes and documentation, The VINT Project, UC Berkeley, LBL, USC/ISI and Xerox PARC, Nov 1997. Available at http://www.isi.edu/nsnam/ns/.

[21] Wireless and mobility extensions to ns-2, The Rice University MONARCH Project, Aug 1999. Available at http://www.monarch.cs.rice.edu/.