A COMPARISON STUDY OF DSDV AND SEAD WIRELESS AD HOC NETWORK ROUTING PROTOCOLS

M.S.R.S Prasad¹, S.S. Panda², MNM Prasad³, S.T.V.S.Kumar⁴

¹Assistant Professor (SG), Dept. of CSE, Regency Institute of Technology, Yanam, prasad_merla@yahoo.com
²Assistant Professor (SG), Dept. of CSE, Regency Institute of Technology, Yanam, shekhar_regency@yahoo.com
³Assistant Professor (SG), Dept. of CSE, Regency Institute of Technology, Yanam, prasadmuhini@gmail.com
⁴Professor, Dept. of CSE, Regency Institute of Technology, Yanam, stvskumar@gmail.com

Abstract
An ad hoc network is a collection of wireless mobile nodes dynamically forming a temporary network without the use of any existing network infrastructure or centralized administration. Due to the limited transmission range of wireless network interfaces, multiple networks “hops” may be needed for one node to exchange data with another across the network. In recent years, a variety of new routing protocols targeted specifically at this environment have been developed, but little performance information on each protocol is available. This paper presents the results of a detailed packet-level simulation comparing two multi-hop wireless ad hoc network routing protocols DSDV and SEAD.

Index Terms: Packet delivery ratio, MAC, ns-2, Routing over head.

1. INTRODUCTION
In a mobile wireless ad hoc network, computers nodes) in the network co operate to forward packets for each other, due to the limited wireless transmission range of each individual node. The network route from some sender node to a destination node may require a number of intermediate nodes to forward packets to create a “multi hop” path from this sender to this destination. The role of the routing protocol in an ad hoc network is to allow nodes to learn such multi hop paths. Since the nodes in the network may move at any time, or may even move continuously, and since sources of wireless interference and wireless transmission propagation conditions may change frequently, the routing protocol must also be able to react to these changes and to learn new routes to maintain connectivity. Josh Broch et al [1] given a very good comparison between DSDV, TORA, DSR and AODV in an efficient way. The recent protocol SEAD designed by Yih-Chun Hu et al[2] has not been compared with any protocol therefore we have taken an attempt to have comparison among them.

The area of ad hoc networking has been receiving increasing attention among researchers in recent years, as the available wireless networking and mobile computing hardware bases are now capable of supporting the promise of this technology. Over the past few years, a variety of new routing protocols targeted specifically at the ad hoc networking environment have been proposed, but little performance information[9,10] on each protocol and no detailed performance comparison between the protocols has previously been available.

This paper provides a realistic, quantitative analysis by comparing the performance of DSDV and SEAD protocols. We present results of detailed simulations showing the relative performance of two ad hoc routing protocols: DSDV and SEAD.

2. DESTINATION-SEQUENCED DISTANCE VECTOR (DSDV)
DSDV [4] is a hop-by-hop distance vector routing protocol requiring each node to periodically broadcast routing updates. The key advantage of DSDV over traditional distance vector protocols is that it guarantees loop-freedom.

2.1 Basic Mechanisms
Each DSDV node maintains a routing table listing the “next hop” for each reachable destination. DSDV tags each route with a sequence number and considers a route R more favorable than Riff R has a greater sequence number, or if the two routes have equal sequence numbers but R has a lower
metric. Each node in the network advertises a monotonically increasing even sequence number for itself. When a node B decides that its route to a destination D has broken, it advertises the route to D with an infinite metric and a sequence number one greater than its sequence number for the route that has broken (making an odd sequence number). This causes any node A routing packets through B to incorporate the infinite-metric route into its routing table until node A hears a route to D with a higher sequence number.

2.2 Implementation Decisions

We did not use link layer breakage detection from the 802.11 MAC protocol in obtaining the DSDV data presented in this paper, because after implementing the protocol both with and without it, we found the performance significantly worse with the link layer breakage detection. The reason is that if a neighbor N of a node A detects that its link to A is broken, it will broadcast a triggered route update containing an infinite metric for A. The sequence number in this triggered update will be one greater than the last sequence number broadcast by A, and therefore does the highest sequence number exist anywhere in the network for A. Each node that hears this update will record an infinite metric for destination A and will propagate the information further. This renders node A unreachable from all nodes in the network until A broadcasts a newer sequence number in a periodic update. A will send this update as soon as it learns of the infinite metric being propagated for it, but large numbers of packets can be dropped in the meantime.

Our implementation uses both full and incremental updates as required by the protocol’s description. However, the published description of DSDV [3,5] is ambiguous about specifying when triggered updates should be sent. One interpretation is that the receipt of a new sequence number for a destination should cause a triggered update. We call this approach DSDV-SQ (sequence number). The advantage of this approach is that broken links will be detected and routed around as new sequence numbers propagate around the broken link and create alternate routes.

3. SECURE EFFICIENT AD HOC DISTANCE VECTOR ROUTING

The secure routing protocol SEAD on the DSDV-SQ version [1,6,7] of the insecure DSDV ad hoc network routing protocol, uses destination sequence numbers, as in DSDV to avoid long-lived routing loops; we also use these destination sequence numbers to provide replay protection of routing update messages in SEAD.

SEAD differ from DSDV in that it does not use an average weighted settling time in sending triggered updates. To reduce the number of redundant triggered updates, each node in DSDV tracks, for each destination, the average time between the node receives the first update for some new sequence number for that destination, and when it receives the best update for that sequence number for it (with the minimum metric among those received with that sequence number); when deciding to send a triggered update, each DSDV node delays any triggered update for a destination for this average weighted settling time, in the hope of only needing to send one triggered update, with the best metric, for that sequence number.

SEAD does not use such a delay, in order to prevent attacks from nodes that might maliciously not use the delay. Since a node selects the first route it receives with highest sequence number and lowest metric, an attacker could otherwise attempt to cause more traffic to be routed through itself, by avoiding the delay in its own triggered updates. Such an attack could otherwise put the attacker in a position to eavesdrop on, modify, or discard other nodes packets.

In addition, unlike DSDV, when a node detect that its next-hop link to some destination is broken, the node does not increment the sequence number for that destination in its routing table when it sets the metric in that entry to infinity. Since higher sequence numbers take priority, this nodes routing update with this new sequence number must be authenticated. Instead, the node flags its routing table entry for this destination to not accept any new updates for this same sequence number, effectively preventing the possible routing loop and traditional distance vector “counting to infinity” problem [7,8] that could otherwise occur in this case. 5.2. Metric and sequence number authenticators.

4. METRICS

In comparing the protocols, we chose to evaluate them according to the following three metrics:

Packet delivery ratio: The ratio between the number of packets originated by the “application layer” CBR sources and the number of packets received by the CBR sink at the final destination.

Routing overhead: The total number of routing packets transmitted during the simulation. For packets sent over multiple hops, each transmission of the packet (each hop) counts as one transmission.
**Path optimality:** The difference between the number of hops a packet took to reach its destination and the length of the shortest path that physically existed through the network when the packet was originated.

5. SIMULATION RESULTS

We conducted simulations using two different node movement speeds: a maximum speed of 20 m/s (average speed 10 m/s) and a maximum speed of 1 m/s. We first compare the two protocols based on the 20 m/s simulations, and then for 1 m/s for comparison. For all simulations, the communication patterns were peer-to-peer, with each run having 10, 20, or 30 sources sending 4 packets per second.

5.1 Comparison Summary

Figures 1 and 2 highlight the relative performance of the two routing protocols on our traffic loads of 20 sources. All of the protocols deliver a greater percentage of the originated data packets when there is little node mobility (i.e., at large pause time), converging to 100% delivery when there is no node motion. In these scenarios, DSDV-SQ fails to converge at pause times less than 300 seconds. The basic character of each protocol is demonstrated in the shape of its overhead curve. SEAD is a on-demand protocol, and their overhead drops as the mobility rate drops. As DSDV-SQ is a largely periodic routing protocol, its overhead is nearly constant with respect to mobility rate.

5.2 Packet Delivery Ratio Details

Figure 3 shows the fraction of the originated application data packets each protocol was able to deliver, as a function of both node mobility rate (pause time) and network load (number of sources). DSDV-SQ fails to converge below pause time 300, where it delivers about 92% of its packets. At higher rates of mobility (lower pause times), DSDV-SQ does poorly, dropping to a 70% packet delivery ratio. Nearly all of the dropped packets are lost because a stale routing table entry directed them to be forwarded over a broken link. DSDV-SQ maintains only one route per destination and consequently, each packet that the MAC layer is unable to deliver is dropped since there are no alternate routes. SEAD does well with 10 or 20 sources, delivering between 90% and 95% of originated data packets even at the highest rate of node mobility (pause time 0). The majority of the packet drops are due to the creation of short-lived routing loops that are a natural part of its link-reversal process. Consider a node A routing packets to C via B. If B’s link to C breaks, B will reverse its link to A, transmit an UPDATE to notify its neighbors it has done this, and begin routing packets to C via A. Until A receives the UPDATE, data packets to C will loop between A and B. Our implementation of SEAD detects when the next-hop of a packet is the same as the previous-hop and drops the data packet, since experiments showed that allowing these packets to loop until their TTL expires or the loop resolves causes more packets to be dropped overall, as the looping data packets interfere with the ability of other near by nodes to successfully exchange the broadcast UPDATE packet that will resolve their routing loop.

With 30 sources, SEAD’s average packet delivery ratio drops to 40% at pause time 0, although upon examination of the data we found that variability was extremely large, with packet delivery ratios ranging from 8% to 91%. In most of these scenarios, SEAD fails to converge because of increased congestion.

5.3 Routing Overhead Details

DSDV-SQ plotted in one scale and SEAD is plotted on other scales to best show the effect of pause time and offered load on overhead. SEAD is on-demand routing protocols, so as the number of sources increases, we expect the number of routing packets sent to increase because there are more destinations to which the network must maintain working routes.
nodes is one update per node per second, yielding an overhead of 45,000 packets for a 900-second, 50-node simulation.

6. CONCLUSIONS
This paper makes use of ns2 simulation environment, we present the results of a detailed packet-level simulation comparing two recent multi-hop wireless ad-hoc network routing protocols. These protocols, DSDV, SEAD, cover a range of design choices, including periodic advertisements vs. on demand route discovery, use of feedback from the MAC layer to indicate a failure to forward a packet to the next hop, and hop-by-hop routing vs. source routing. We simulated each protocol in ad hoc networks of 50 mobile nodes moving about and communicating with each other, and presented the results for a range of node mobility rates and movement speeds.

Each of the protocols studied performs well in some cases yet has certain drawbacks in others. DSDV performs quite predictably, delivering virtually all data packets when node mobility rate and movement speed are low, and failing to converge as node mobility increases. SEAD, although the worst performer in our experiments in terms of routing packet overhead, still delivered over 90% of the packets in scenarios with 10 or 20 sources. At 30 sources, the network was unable to handle all of the traffic generated by the routing protocol and a significant fraction of data packets were dropped.

REFERENCES


BIOGRAPHIES

MSRS Prasad got his M.Tech. Degree in the year 2010 and doing his research on Digital image processing. He is working as Sr. Assistant Professor in the department of Computer Science and Engineering, Regency Institute of Technology, Yanam. (Puducherry), India. He is a life member of ISTE.

S. S. Panda got his M.Tech. Degree in the year 2010 and doing his research on Digital image processing. He is working as Sr. Assistant Professor in the department of Computer Science and Engineering, Regency Institute of Technology, Yanam. (Puducherry), India. He is a life member of ISTE.

MNM Prasad got his M.Tech. Degree in the year 2011. He is working as Sr. Assistant Professor in the department of Computer Science and Engineering, Regency Institute of Technology, Yanam. (Puducherry), India. He is a life member of ISTE.

S.T.V.S.Kumar got his M.Tech. Degree and doing his research on Software Engineering. He is working as Professor in the department of Computer Science and Engineering, Regency Institute of Technology, Yanam. (Puducherry), India. He is a life member of ISTE.