Mobile Ad Hoc Routing Protocols using ns2 simulations

M.Rajmohan
Asst Prof,ECE,
Hindustan University,
Chennai.

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. A number of routing protocols like Dynamic Source Routing (DSR), Ad Hoc On-Demand Distance Vector Routing (AODV), Destination-Sequenced Distance-Vector (DSDV) and Temporally Ordered Routing Algorithm (TORA) have been implemented. In this project an attempt has been made to compare the performance of two prominent on-demand reactive routing protocols for mobile ad hoc networks: DSR and AODV, along with the traditional proactive DSDV protocol. A simulation model with MAC and physical layer models is used to study interlayer interactions and their performance implications. The On-demand protocols, AODV and DSR perform better than the table-driven DSDV protocol. Although DSR and AODV share similar on-demand behavior, the differences in the protocol mechanics can lead to significant performance differentials. The performance differentials are analyzed using varying network load, mobility, and network size. These simulations are carried out based on the Rice Monarch Project that has made substantial extensions to the ns-2 network simulator to run ad hoc simulations.

1 Introduction

A mobile ad hoc network (MANET) consists of a set of wireless mobile nodes. Arbitrary motion of mobile nodes can change the network topology dramatically, and thus finding an efficient and reliable route between source and destination is a challenging task in MANETs. Significant research efforts have been carried out to design routing protocols for ad hoc networks. Most of them take the shortest-path with minimum hop count as the route selection

Issn.No:0975-0932

criterion. But some important link capacity properties have been ignored. First, each node has different traffic load. Therefore, the average number of packets in the queue and the associated queuing delay at each node is different. Second, the numbers of a node's neighbor nodes as well as their traffic patterns are different, and thus nodes that have more active neighbors may encounter more collisions. If the shortest route includes some of these heavy nodes, it may actually cause end-to-end delay longer even though the number of hops is minimal. Furthermore, if one of the heavy nodes is congested, it may lead to massive packet drops, higher packet dropping rate, retransmission and faster battery power depletion on certain mobile nodes. Load aware routings for ad hoc networks have been proposed in some papers. But most of these protocols use queue size as the main traffic load metric. This metric works well in wired networks, but it does not reflect the impact of another important factor: channel contention from neighbor nodes. In a wireless network, nodes contend for the shared channel, which causes access delay and collision at MAC layer. In this paper, we propose a Contention and Queue-aware Routing (CQR) that utilizes the contention information (contention window, CW) collected from the 802.11 Distributed Coordination Function (DCF) [2] and queue size. With this information, the channel's contention situation and the neighbor's traffic load can be estimated and considered for making routing decisions. The rest of this paper is organized as follows.

2. Description of the Ad-hoc Routing Protocols

Destination-Sequenced Distance-Vector (DSDV)

The Destination-Sequenced Distance-Vector (DSDV) Routing Algorithm is based on the idea of the classical Bellman-Ford Routing Algorithm with certain improvements. Every mobile station maintains a routing table that lists all available destinations, the number of hops to reach the destination and the sequence number assigned by the destination node. The sequence number is used to distinguish stale routes from new ones and thus avoid the formation of loops. The stations periodically transmit their routing tables to their immediate neighbors. A station also transmits its routing table if a significant change has occurred in its table from the last update sent. So, the update is both time-driven and

Issn.No:0975-0932

event-driven. The routing table updates can be sent in two ways: - a "full dump" or an incremental update. A full dump sends the full routing table to the neighbors and could span many packets whereas in an incremental update only those entries from the routing table are sent that has a metric change since the last update and it must fit in a packet. If there is space in the incremental update packet then those entries may be included whose sequence number has changed. When the network is relatively stable, incremental updates are sent to avoid extra traffic and full dump are relatively infrequent. In a fast-changing network, incremental packets can grow big so full dumps will be more frequent.

Temporally Ordered Routing Algorithm (TORA)

TORA is a distributed routing protocol based on a “link reversal” algorithm. It is designed to discover routes on demand, provide multiple routes to a destination, establish routes quickly, and minimize communication overhead by localizing algorithmic reaction to topological changes when possible. Route optimality (shortest-path routing) is considered of secondary importance, and longer routes are often used to avoid the overhead of discovering newer routes.The actions taken by TORA can be described in terms of water flowing downhill towards a destination node through a network of tubes that models the routing state of the real network. The tubes represent links between nodes in the network, the junctions of tubes represent the nodes, and the water in the tubes represents the packets flowing towards the destination. Each node has a height with respect to the destination that is computed by the routing protocol. If a tube between nodes **A** and **B** becomes blocked such that water can no longer flow through it, the height of **A** is set to a height greater than that of any of its remaining neighbors, such that water will now flow back out of **A** (and towards the other nodes that had been routing packets to the destination via A).When a node discovers that a route to a destination is no longer valid, it adjusts its height so that it is a local maximum with respect to its neighbors and transmits an UPDATE packet. If the node has no neighbors of finite height with respect to this destination, then the node instead attempts to discover a new route as described

Issn.No:0975-0932

above. When a node detects a network partition, it generates a CLEAR packet that resets routing state and removes invalid routes from the network.

Dynamic Source Routing (DSR)

The key distinguishing feature of DSR is the use of source routing. That is, the sender knows the complete hop-by-hop route to the destination. These routes are stored in a route cache. The data packets carry the source route in the packet header. When a node in the ad hoc network attempts to send a data packet to a destination for which it does not already know the route, it uses a route discovery process to dynamically determine such a route. Route discovery works by flooding the network with route request (RREQ) packets. Each node receiving an RREQ rebroadcasts it, unless it is the destination or it has a route to the destination in its route cache. Such a node replies to the RREQ with a route reply (RREP) packet that is routed back to the original source. RREQ and RREP packets are also source routed. The RREQ builds up the path traversed across the network. The RREP routes itself back to the source by traversing this path backward. The route carried back by the RREP packet is cached at the source for future use. If any link on a source route is broken, the source node is notified using a route error (RERR) packet. The source removes any route using this link from its cache. A new route discovery process must be initiated by the source if this route is still needed. DSR makes very aggressive use of source routing and route caching. No special mechanism to detect routing loops is needed. Also, any forwarding node caches the source route in a packet it forwards for possible future use.

Ad Hoc On-Demand Distance Vector Routing (AODV)

AODV shares DSR's on-demand characteristics in that it also discovers routes on an as needed basis via a similar route discovery process. However, AODV adopts a very different mechanism to maintain routing information. It uses traditional routing tables, one entry per destination. This is in contrast to DSR, which can maintain multiple route

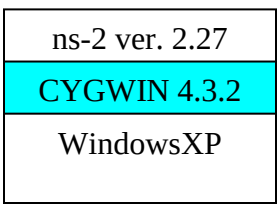
Issn.No:0975-0932

cache entries for each destination. Without source routing, AODV relies on routing table entries to propagate an RREP back to the source and, subsequently, to route data packets to the destination. AODV uses sequence numbers maintained at each destination to determine freshness of routing information and to prevent routing loops. All routing packets carry these sequence numbers.An important feature of AODV is the maintenance of timer-based states in each node, regarding utilization of individual routing table entries. A routing table entry is expired if not used recently. A set of predecessor nodes is maintained for each routing table entry, indicating the set of neighboring nodes which use that entry to route data packets. These nodes are notified with RERR packets when the next-hop link breaks. Each predecessor node, in turn, forwards the RERR to its own set of predecessors, thus effectively erasing all routes using the broken link. In contrast to DSR, RERR packets in AODV are intended to inform all sources using a link when a failure occurs. Route error propagation in AODV can be visualized conceptually as a tree whose root is the node at the point of failure and all sources using the failed link as the leaves.

3. Simulation Model

Ns-2 is an open source discrete event simulator used by the research community for research in networking . It has support for both wired and wireless networks and can simulate several network protocols such as TCP, UDP, multicast routing, etc. More recently, support has been added for simulation of large satellite and ad hoc wireless networks. The ns-2 simulation software was developed at the University of Berkeley. It is constantly under development by an active community of researchers. The latest version at the time of writing this tutorial is ns-2 2.27.

The standard ns-2 distribution runs on Linux. However, a package for running ns-2 on Cygwin (Linux Emulation for Windows) is available. In this mode, ns-2 runs in the Windows environment on top of Cygwin as shown in the figure 1.



Issn.No:0975-0932

Fig.1 ns-2 over Cygwin

In this tutorial, we initially discuss the general installation and configuration of ns-2. Later on, we will discuss how to simulate and analyze the performance of routing protocols for Mobile Ad hoc networks using scenario based experiments. Finally, a list of useful resources is provided for the novice user.

4 Getting your hands wet with ns-2

The ns-2.27 is available as an all-in-one package that includes many modules. Two modules that we will discuss in this tutorial are

i. ns-2 simulator.
ii. TCL/OTcl interpreter.

3.1 The Traffic and Mobility Models

Continuous bit rate (CBR) traffic sources are used. The source-destination pairs are spread randomly over the network. Only 512-byte data packets are used. The number of source-destination pairs and the packet sending rate in each pair is varied to change the offered load in the network. The mobility model uses the random waypoint model in a rectangular field. The field configurations used is: 500 m x 500 m field with 50 nodes. Here, each packet starts its journey from a random location to a random destination with a randomly chosen speed (uniformly distributed between 0–20 m/s). Once the destination is reached, another random destination is targeted after a pause. The pause time, which affects the relative speeds of the mobiles, is varied. Simulations are run for 100 simulated seconds. Identical mobility and traffic scenarios are used across protocols to gather fair results.

5. Performance Metrics

Three important performance metrics are evaluated:

4.1 Packet delivery fraction — The ratio of the data packets delivered to the destinations to those generated by the CBR sources.

4.2 Average end-to-end delay of data packets — This includes all possible delays caused
by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, and propagation and transfer times.

4.3 Normalized routing load — The number of routing packets transmitted per data packet delivered at the destination. Each hop-wise transmission of a routing packet is counted as one transmission.

Issn.No:0975-0932

The first two metrics are the most important for best-effort traffic. The routing load metric evaluates the efficiency of the routing protocol. Note, however, that these metrics are not completely independent. For example, lower packet delivery fraction means that the delay metric is evaluated with fewer samples. In the conventional wisdom, the longer the path lengths, the higher the probability of a packet drops. Thus, with a lower delivery fraction, samples are usually biased in favor of smaller path lengths and thus have less delay.

6. Results and Discussions

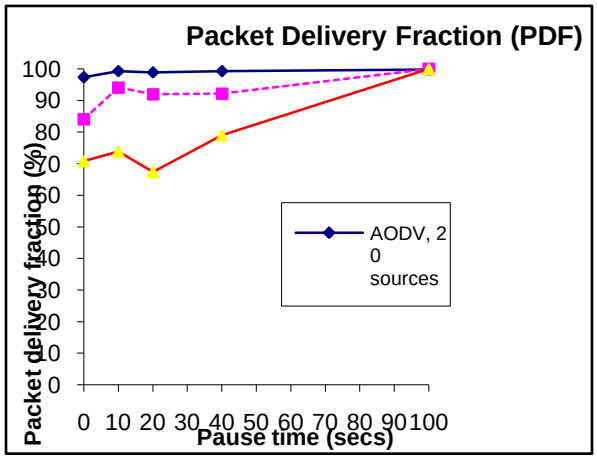


Fig 2 Simulation Result

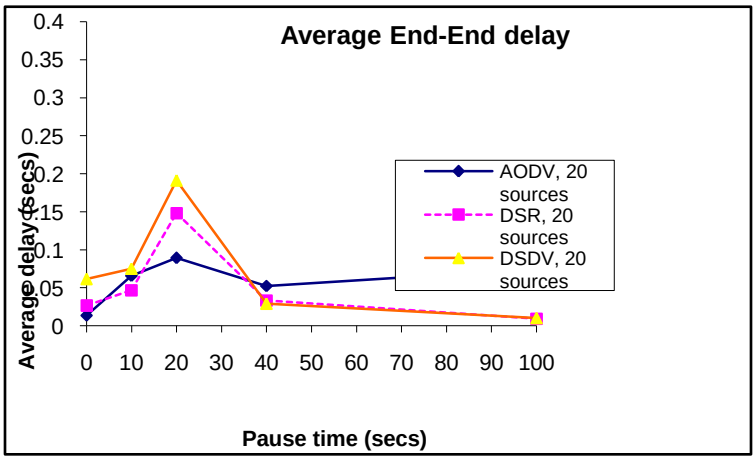


Figure 3 Simulation Result

6.1 Performance comparison of the protocols:

Issn.No:0975-0932

First, an attempt was made to compare all the 4 protocols under the same simulation environment. However, simulations couldn't be successfully carried out for the TORA routing protocol, as ns-2 repeatedly gave a bus error while running the TORA simulations. For all the simulations, the same movement models were used, the number of traffic sources was fixed at 20, the maximum speed of the nodes was set to 20m/s and the pause time was varied as 0s, 10s, 20s, 40s and 100s.

Figures 2 and 3 highlight the relative performance of the three routing protocols. 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.

6.1.1 Packet delivery Comparison:

The On-demand protocols, DSR and AODV performed particularly well, delivering over 85% of the data packets regardless of mobility rate.

6.1.2 Average End-End Packet delivery:

The average end-to-end delay of packet delivery was higher in DSDV as compared to both DSR and AODV.

In summary, both the On-demand routing protocols, AODV and DSR outperformed the Table-driven routing protocol; DSDV and the reasons are discussed later.

Next, since both AODV and DSR did better, an attempt was made to evaluate the performance difference between the two by varying the Mobility pattern and Number of traffic sources.

In summary, when the number of sources is low, the performance of DSR and AODV is similar regardless of mobility. With large numbers of sources, AODV starts outperforming DSR for high-mobility scenarios. As the data from the varying sources demonstrate, AODV starts outperforming DSR at a lower load with a larger number of nodes. DSR always demonstrates a lower routing load than AODV. The major contribution to AODV's routing over-head is from route requests, while route replies constitute a large fraction of DSR's routing overhead. Furthermore, AODV has more route requests than DSR, and the converse is true for route replies.

DSDV fails to converge below lower pause times. At higher rates of mobility (lowerpause times), DSDV does poorly, dropping to a 70% packet de-livery ratio. Nearly all of the dropped packets are lost because a stale routing table entry directed them to be forwarded over a broken link. As described in the earlier section, DSDV 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. Since DSDV uses the table-driven

Issn.No:0975-0932

approach of maintaining routing information, it is not as adaptive to the route changes that occur during high mobility. In contrast, the lazy approach used by the on-demand protocols, AODV and DSR to build the routing information as and when they are created make them more adaptive and result in better performance (high packet delivery fraction and lower average end-to-end packet delays).

7. Conclusions

The performance of DSDV, AODV and DSR routing protocols for ad hoc networks using ns-2 simulations. Unfortunately, TORA simulations couldn't be successfully carried out. DSDV uses the proactive table-driven routing strategy while both AODV and DSR use the reactive On-demand routing strategy. Both AODV and DSR perform better under high mobility simulations than DSDV. High mobility results in frequent link failures and the overhead involved in updating all the nodes with the new routing information as in DSDV is much more than that involved AODV and DSR, where the routes are created as and when required.DSR and AODV both use on-demand route discovery, but with different routing mechanics. In particular, DSR uses source routing and route caches, and does not depend on any periodic or timer-based activities. DSR exploits caching aggressively and maintains multiple routes per destination. AODV, on the other hand, uses routing tables, one route per destination, and destination sequence numbers, a mechanism to prevent loops and to determine freshness of routes.

A. References

[1] J. Broch, D. A. Maltz, D. B. Johnson DB, Y-C. Hu , J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols" , Proc. of ACM MOBICOM'98, October 1998.

[2] ANSIIIEEE SU 802.11, "Wireless lan medium access control (MAC) and physical layer (PHY) specifications", 1999.

Issn.No:0975-0932

[3] S-J. Lee, M. Gerla, "Dynamic load-aware routing in ad hoc networks", Proc. of IEEE ICC, October 2001: 3206-3210.

[4] H. Hasanein, A. Zhou, "Routing with load balancing in wireless ad hoc networks," Proc. of the 4th ACM international workshop on modeling, analysis and simulation of wireless and mobile systems,July 2001:89-96.

[5] K. Wu, I. Harms, "Load-sensitive routing for mobile ad hoc networks," Proc. of Tenth Intl. Conf. on Computer Communications and Networks, October 2001:540-546.

[6] J.H. Song, V. Wong, V. Leung, "Load-aware on-demand routing protocol for mobile ad hoc networks", Proc. of the 57th IEEE Semiannual Vehicular Technology Conference, March 2003:1753-1757.

[7] S.-T. Sheu, J. Chen, "A novel delay-oriented shortest path routing protocol for mobile ad hoc networks", Proc. of IEEE ICC, 2001.

[8] X. Q. Zheng , L. J. Ge, W. Guo,"A load-balanced MAC protocol for multi-channel ad-hoc networks", Proc. of 6th International Conference ITS Telecommunications, June 2006.

[9] Y. Li, H. Man, "Three load metrics for routing in ad hoc networks", Proc. of Vehicular Technology Conference, September 2004:2764 - 2768.

[10] F.F. Zou, X.M. Zhang, X.M. Gao, D. Shi, E.B. Wang, " Load balance routing using packet success rate for mobile ad hoc networks", Proc. Of the 3rd IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WICOM'07), Sep. 2007.

[11] F. Xie, X.M. Zhang, J.F. Guo, G. L. Chen, "A delay oriented adaptive routing protocol for mobile ad hoc networks", Journal of Software,16(9): 1661−1667, Sep. 2005

[12] J.F. Guo, X.M. Zhang, F. Xie, G.L. Chen, "A leisure degree adaptive routing protocol for mobile ad hoc network", Journal of Software, 16(5):960−969, May 2005.

Issn.No:0975-0932