Performance Analysis of Various Protocols for Manets through Simulation

Ajit Kumar
Lecturer, Deptt. of Computer Engg., Govt. Polytechnic, Hisar

Dharmender Kumar
Associate Professor, Deptt. of Computer Sc. & Engg., DCRUST, Murthal

ABSTRACT
Mobile Ad Hoc Network (MANET) is a collection of wireless mobile hosts that communicate with each other without fixed infrastructure and central administration. Routes between two hosts in a MANET may consist of hops through other hosts in the network. Host mobility can cause frequent unpredictable topology changes. Many protocols have been proposed for MANETs for efficient routing. The primary goal of routing protocols is to establish route with a minimum of overhead and bandwidth consumption. This paper analyses the performance of various protocols for efficiency (in terms of packet loss) and throughput (in terms of bps) with simulation results.

1. INTRODUCTION
Wireless networks can be classified in two types: i) Infrastructure based networks with fixed and wired gateways. In this network, nodes are restricted to move within the transmission range of networking devices. When it goes out of range of one base station, it connects with new base station and starts communication. ii) Mobile Ad Hoc Networks (MANETs) without fixed and wired gateways. In this network, each of wireless nodes performs routing functions and forwards packets to others. Thus each node in a MANET acts as a router. It requires proper designing of routing protocol for efficient and correct delivery of information between communicating nodes. The routing protocols are classified according to their characteristics in many ways. However, the most common form of classification is Proactive versus On-Demand routing protocols. A routing protocol can maintain routing information either on-demand or proactively (at all times). Further, proactive protocols can be divided into protocols that update routing information in regular intervals and protocols that update on certain events.

2. REVIEW OF ROUTING ALGORITHMS
In Mobile Ad Hoc Networks, the communication does not rely on any existing infrastructure such as dedicated routers, transceiver base stations or even cables of any kind. Mobile devices (e.g. notebook computers, PDAs, cell phones, etc.) with wireless radio equipment are supposed to communicate with each other, without the help of any other fixed devices. In order to make that work, typically each node needs to act as a router to relay packets to nodes out of direct communication range. Under these circumstances, routing is much more complex than in conventional (static) networks. Many of the possible solutions are determined by the characteristics of the media, the behavior of nodes and the data flow.

Since research in Ad Hoc Networking has resulted in such a large amount of routing algorithms and protocols, it has become more and more difficult to decide, which algorithms are superior to others under what conditions. For a successful deployment, this is an important problem, since a wrong choice may have a severe impact on the performance, and consequently on the acceptance of the new technology. Also providing just any protocol is not feasible, due to the different requirements on hardware and lower network layers. Further, it would not make sense, since all devices in an area would need to agree on one method if they want to communicate. These routing protocols are classified according to their characteristics in many ways such as follows:

- Single channel Vs Multi channel
- Uniform vs. Non-Uniform Protocols
- Hierarchical Topology vs. Clustered Routing
- Position based Protocols
- Proactive vs. Reactive Routing Protocols

2.1 Proactive vs. Reactive Routing Protocols
A routing protocol can maintain routing information either on-demand or proactively (at all times). Further proactive protocols can be divided into protocols that update routing information on regular intervals and protocols that update on certain events. Finally, there are routing protocols that are hybrid and make use of both schemes. Considering the dynamic nature of the wireless networks, the reactive protocols are regarded as more efficient routing solutions since they build the routes on demand.

2.1.1 Reactive (On-Demand) Protocols
A network using an on-demand protocol will not maintain correct routing information on all nodes for all times. Instead, such routing information is obtained on demand. If a node wants to transmit a message, and does not have enough routing information to send the message to the destination, the required information has to be obtained. Usually the node needs at least to know the next hop (among its neighbors) for the packet. Although the node could just broadcast the packet to all neighbors, this leads to serious congestion in many cases. However, such broadcasts are used in a route discovery process, since there is no other next-hop information available, yet. Usually this consists of a broadcast message from the originating node, indicating the desired route. Nodes which have the required information will respond to the originating node, which will eventually choose a route from the replies it received. The broadcast may be limited to travel only a few hops first, before net wide broadcast will be issued (which would flood the whole network). Of course, the route request and selection process must be finished, before the message can be sent. This leads to an initial setup delay for messages, if their route is not known to the node. To limit the impact of this delay, most protocols will use a route cache for once established routes. However, the information in this cache will time out, since in a mobile environment, the routes will be invalid after some time. Clearly, applications that are used over an on-demand routing protocol need to be tolerant for such an initials setup delay. The advantage of on-demand routing protocols lies in the fact that the wireless channel (a scarce resource) does not need to carry a lot of routing overhead data for routes that are not even used. This advantage may diminish in certain scenarios where there is a lot of traffic to a large variety of nodes. Thus the scenario will have a very significant impact on the performance. In such a scenario with lots of traffic to many nodes, the route-setup traffic can grow larger than constant background traffic to maintain correct routing information on each node. Still, if

enough capacities would be available, the reduced efficiency (increased overhead) might not affect other performance measures, like throughput or latency.

There also exists some location based protocols as on-demand protocols, since they determine the direction in which to send the packet on demand and some protocols may even initiate a location query of the destination nodes for their packets on demand. Few examples for on-demand protocols are the following:
ABR, AODV, CEDAR, DREAMS, DSR, FORP, GEDIR, LAR, SSR, WAR.

AODV – Ad-Hoc on Demand Distance Vector Routing Protocol
This is one of the most discussed and most advanced routing protocols. This is probably the most mature suggestion for an ad hoc routing protocol. Its main developers are Charles E. Perkins and Elizabeth Belding-Royer. AODV is discussed in lots of studies and is often used as a reference to compare other routing protocols. AODV was derived from C. Perkins earlier work, DSDV. Compared to DSDV, AODV no longer needs to exchange periodic messages proactively, but works in an on-demand fashion, instead. If a route to a destination is unknown, a route discovery process is initiated. This consists of broadcasting a Route Request (RREQ) packet throughout the network. To limit the impact of a net wide broadcast, these requests should be sent with an expanding ring search technique: the TTL of the packets starts with a small value; if no route has been found, the TTL will be increased and the request will be resent. Each node that rebroadcasts this request, adds its address into a list in the packet. If the destination sees the request, it will reply with a unicast Route Reply (RREP) to the source. Each intermediate node may cache the learned routes. The routing table entries consist of a destination, the next hop toward this destination and a sequence number. Routes are only updated if the sequence number of the updating message is larger than the existing one. Thus routing loops and updates with stale information are prevented. The sequence number technique was already used in DSDV and was adopted by a variety of other routing protocol developers. The amount of information, which needs to be present at each node, is rather limited:

- The node is aware of its neighbors (via link-layer-notification, or explicit HELLO messages).
- The node knows route destinations and the next hop.
- The node has a "precursor list" for each destination. This list consists of all nodes, which use the current node as a relay for the destination.
- In case of a route failure to this destination, the node knows exactly which other nodes to notify.
- Each routing entry also has a lifetime.

DSR - Dynamic Source Routing
DSR is an on-demand protocol that .uses source routing. In this case, this means, that each packet carries the complete route to its destination in its header (which introduces some overhead). Since DSR works on demand, a route must be discovered through a Route Discovery Mechanism before use. Discovered routes may be cached and routes may be overheard by a node (by parsing the source route information of packets that are relayed). If broken links are detected, a corresponding Route Error message is transmitted through the network and a route maintenance mechanism takes over to fix the broken routes, if possible. To further reduce unnecessary traffic, a node may reply to a route request with a locally cached route, even if it is not the destination node. Delays in these replays with promiscuous observation (overhearing) of other routing traffic prevent multiple nodes replying with a cached entry all at once. The dynamic source routing protocol is also a very mature protocol.

2.1.2 Proactive Protocols
Proactive routing protocols will try to maintain correct routing information on all nodes in the network at all times. This can be achieved in different ways, and thus divides the protocols into two subclasses: event driven and regular updated protocols. Event driven protocols will not send any routing update packets, if no change in topology occurs. Only if a node detects a change of the topology (usually a node moves out of reach of this node, or a new node comes close enough), this is reported to other nodes, according to the strategy of the routing protocol. Protocols that are updated in regular intervals will always send their topology information to other nodes at regular intervals. Many link state protocols work in such a manner (but varying the maximum distance of an update message with the length of the interval). Nodes farther away get updates less frequently than close nodes, thus balancing the load imposed on the network.
Proactive protocols of either subclass impose a fixed overhead to maintain the routing tables. Even if many of the entries are not used at all. Their advantage is that the routes can be used at once and there is no setup delay.
Event driven proactive routing protocols are:
CBRP, CGSR, DSDV, GSR, LMR, TORA and WRP
Regular updated protocols are:
DDR, FSLS, FSR, GPSR, LAN MAR, OLSR, STAR and TBRPF

WRP - Wireless Routing Protocol
This is a proactive protocol that maintains routing information through the exchange of triggered and periodic updates. A node that successfully receives an update message, transmits an acknowledgement back to the sender, indicating the link is still viable. In the event that a node has not transmitted anything within a specified period of time, it must transmit a Hello message to ensure connectivity. Otherwise, the lack of messages from a node indicates the failure of that link. When a node receives a Hello message from a new node, it sends that neighbor a copy of its routing table information. This is based on a path-finding algorithm that reduces the probability or routing loops. In WRP, each node shares its routing tables with its neighbors, by communicating the distance and second to- last hop to each destination. Nodes send an acknowledgement upon reception of update routes. Each node maintains a distance table, a routing table, a link-cost table, and a message retransmission list. However, the relatively large no. of tables that this protocol requires imposes tight memory constraints on the mobile devices, especially when the network begins to grow.

3. GLOMOSIM SIMULATOR
Global Mobile Information System Simulator (GloMoSim) is a scalable simulation environment for large wireless and wireline communication networks. GloMoSim uses a parallel discrete-event simulation capability provided by Parsec. GloMoSim simulates networks with up to thousand nodes linked by a heterogeneous communications capability that includes multicast, asymmetric communications using direct satellite broadcasts, multi-hop wireless communications using ad-hoc networking, and traditional Internet protocols.
The node aggregation technique is introduced into GloMoSim to give significant benefits to the simulation performance. Initializing each node as a separate entity inherently limits the scalability because the memory requirements increase dramatically for a model with large number of nodes. With node aggregation, a single entity can simulate several network nodes in the system. Node aggregation technique implies that the number of nodes in the system can be increased while maintaining the same number of entities in the simulation. In GloMoSim, each entity represents a geographical area of the simulation. Hence the network nodes which a particular entity represents are determined

by the physical position of the nodes.Most network systems are currently built using a layered approach that is similar to the OSI seven layer network architecture. Standard APIs will be used between the different simulation layers. This will allow the rapid integration of models developed at different layers by different people.

Glomosim can be modified to add new protocols and applications to the library. Therefore Glomosim is a good choice for implementing the new proposed algorithm. GIoMoSim is aimed at simulating models that may contain as many as 100,000 mobile nodes with a reasonable execution time; this is done by using node aggregation. As each entity needs to examine packets received only from the nodes located in the region it is simulating, many partitions are used to reduce the total search space for packet delivery. If a packet sent by a node located in Partition (0, 0) cannot reach the border of the partition, no message needs to be sent to the other partitions.

4. SIMULATION ENVIRONMENT

For simulation the basic input file "config.in" needs to be configured. This file contains the general simulation parameters for glomosim. It is structured according to each layer and allows a protocol to be chosen for each layer. The results were calculated using the following parameters:

[nodes.input file]

0 0 (100, 1000, 0.0)
1 0 (500, 1000, 0.0)
2 0 (900, 1000, 0.0)

the simulation model is node0--- 400m --- node1 ---- 400m ---- node2

[config.in file]

NUMBER OF NODES: 3
NODE-PLACEMENT: FILE
NODE-PLACEMENT-FILE: /nodes.input
MAC-PROTOCOL:
802.11/CSMA/MACA/TSMA
NETWORK-PROTOCOL: IP
ROUTING-PROTOCOL: AODV/WRP/DSR
APP-CONFIG-FILE: /app.conf
RADIO-TYPE: RADIO-ACCNOISE

[app.conf]

CBR 0 1 0 512 4MS 0 0
CBR 2 1 0 512 4MS 6MS 0

Constant Bit Rate from node 0 to 1 and node 2 to 1 where node0 continuously sends node1 items of 512B each at the start of the simulation up to the end of the simulation. The inter-departure time for each item is 4MS. Node2 continuously sends node1 items of 512B each 6MS after start of simulation up to the end of the simulation.

5. RESULTS & PERFORMANCE ANALYSIS

To run the input file, from the /bin directory, type "glomosim config.in". The simulation will run and produce an output file "glomo.stat". An example of "glomo.stat" is given below. Several cases were studied by varying different parameters in nodes.input file, config.in file and app.conf file.

In app.conf:

CBR 2 1 0 512 4MS 0 0

Output:

Node:1,Layer:AppCbrServer,(0)Client address: 2
Node:1,Layer:AppCbrServer,(0)First packet received at [s]: 0.003316999
Node:1,Layer:AppCbrServer,(0)Last packet received at [s]: 4.998408333
Node:1,Layer:AppCbrServer,(0) Average end-to-end delay [s]: 0.002409059
Node:1, Layer:AppCbrServer,(0) Total number of bytes received: 640000
Node:1, Layer:AppCbrServer,(0) Total number of packets received: 1250
Node:1, Layer:AppCbrServer,(0)Throughput (bits per second): 1024679
Node:2, Layer:AppCbrClient, (0) Server address: 1
Node:2, Layer:AppCbrClient, (0) First packet sent at [s]: 0.000000000
Node:2, Layer:AppCbrClient, (0) Last packet sent at [s]: 4.996000000
Node:2, Layer:AppCbrClient, (0) Total number of bytes sent: 640000
Node:2, Layer:AppCbrClient, (0) Total number of packets sent: 1250
Node:2, Layer:AppCbrClient, (0) Throughput (bits per second): 1024000
packets sent: 1250 packets received: 1250

All packets that are sent by node 2 are clearly received by node 1. The same occurs if we let node 0 to transmit alone, but if we let both nodes 0 and 2 to transmit at the same time to node 1, the output is different.

The comparison is made on following parameters:

a) Throughput (bits per second)
b) Efficiency (no. of packets received to the packets transmitted)

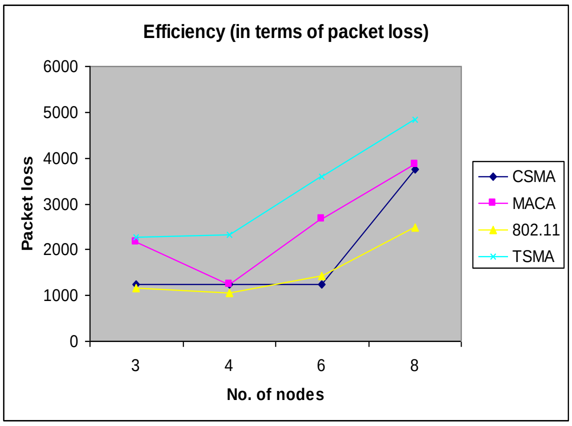

Figure 1: Comparison of Efficiency

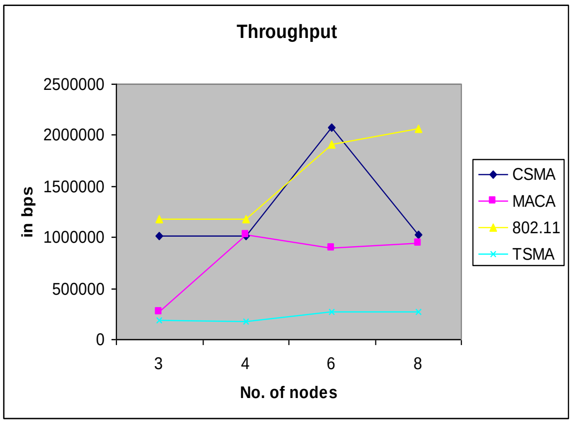

Figure 2: Comparison of throughput

Similarly the output was obtained by comparing three algorithms, two with reactive approach and one with proactive approach. The comparison is made on following parameters:

a) Throughput (bits per second)
b) Efficiency (no. of packets received to the packets transmitted)

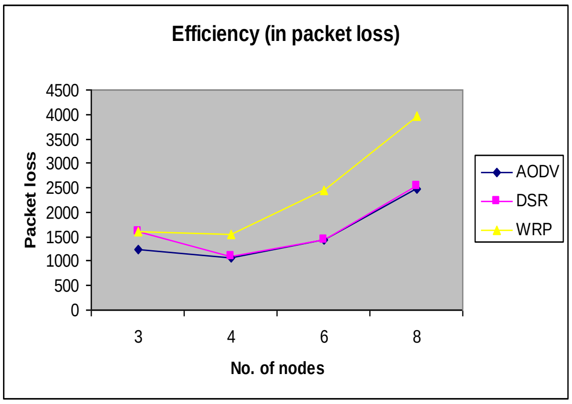

Figure 3: Comparison of Efficiency

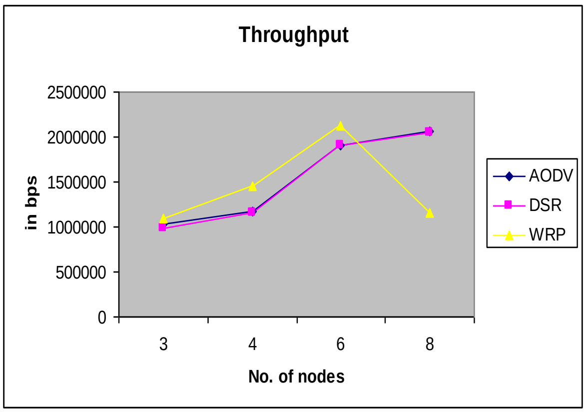

Figure 4: Comparison of Throughput

Efficiency:

When we compare efficiency, total number of packets sent is compared with total number of packets received. It gives us the idea about packets loss which has taken place. The packet loss could be because of terrain constraints, collisions etc. From the figure 3 it is observed that packet loss in WRP is much more than AODV and DSR. It is happening first because of terrain constraints of WRP and also because of proactive approach of WRP. The collision increases as the number of nodes grow hence the number of packets loss also increases. In AODV when ever source node wants to establish a connection, typically a kind of broadcasting takes place which further becomes worse with flooding of these RERP packets during route discovery process. It causes collision, droppings of RERP packet due to TTL time out and hence number of packets loss.

Throughput:

It is measured in bits per second. As in WRP, it is proactive approach, so nodes exchange information with neighbors even if no data transmission takes place. Hence all the nodes have enough information about the topology, nearest gateways, hop count, next hop etc. Hence when ever the transmission takes place from application layer, set up time would be very less. This could be more prominent as the number of nodes grow. With AODV, source node first initiates node discovery for the destination before any transmission can take place. This causes sufficient delay at the time of setting up the connection.

6. CONCLUSIONS

The various algorithms have been compared including reactive as well as proactive i.e. AODV, WRP, DSR etc. The AODV has found to be the best in terms of no. of packets received and packets sent (i.e. packets loss) whereas WRP has shown a good throughput. Further the algorithms may be compared on other parameters also. As there are a large number of routing algorithms and protocols, it is more difficult to decide, which algorithm is superior to others in different situations. For a successful deployment, this is an important problem, since a wrong choice may have a severe impact on the performance, and consequently on the acceptance of the new technology. Also providing just any protocol is not feasible, due to the different requirements on hardware and lower network layers. Hence appropriate algorithm must be chosen after detail study of hardware and software requirements and environment.

7. REFERENCES

[1] Glomosim: Global Mobile Information Systems Simulation Library at http:/ /pcl.cs.ucla.edu/ projects/ glomosim/

[2] Jorge Nuevo, "A Comprehensible Glomosim Tutorial", March 4, 2004.

[3] Deniel Lang, "A comprehensive overview about selected AD Hoc Routing protocols", March 14, 2003.

[4] C.K. Toh, Elizabeth M. Royer, "Review of current routing protocols for ad-hoc mobile wireless networks", Tech. Rep., UCSB, 1999.

[5] C. Perkings, E. Belding Royer and S. Das, "RFC 3561" Ad hoc On Demand Distant Vector Routing", July, 2003.

[6] C. Hedrick, "Routing information protocol" RFC 1058, 1988.

[7] Elizabeth M. Royer, Santa Barbara and Chai-Keong Toh, "A Review of Current Routing Protocols for Ad-hoc Mobile Wireless Networks", IEEE Personal Communications, April 1999.

[8] C.Ahlund, A. Zaslavskyn, "Software solution to Internet connectivity in Mobile Ad Hoc Networks", Melbourn, Astralia, 2002.

[9] P. Jacquet and A. Laouiti, "Analysis of mobile ad-hoc network routing protocols in random graph models", Tech. Rep. 3835, INRIA, France, 1999.

[10] M.Gerla, L.Bajaj, M.Takai, R.Ahuja, R.Bargodia, "Glomosim:A Scalable Network Simulation Environment", Technical Report 990027, University of California, 13, 1999.

[11] Hong Jiang, JJ Garcia-Luna-Aceves, "Performance comparison of Three Routing Protocols for Ad Hoc Networks", Computer Engg. Deptt., Baskin School of Engg., University of California, U.S. 2001.