IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 10, Issue 6 (May. - Jun. 2013), PP 42-49
www.iosrjournals.org

Routing protocols in Ad-hoc Networks- A Simulation Study

Chanchal*, Manisha*, Pawan Bhadana**,Ritu Khurana**
** Computer Science & Engineering B.S.A. Institute of Technology & Management Faridabad, India*
***Department of Computer Engineering B.S.A. Institute of Technology & Management Faridabad, India*

Abstract: *An ad-hoc network is a temporary network without any form of centralized administration. Multiple hops might be necessary to reach other nodes in the network. For this reason, each node acts both as a router and a host, meaning that every node must be willing to forward packets for other nodes. For this reason a routing protocol is needed.*
Keywords: *Ad-hoc, Routing, Wireless.*

I. Introduction

Ad-hoc networks are a new paradigm of wireless communication for mobile hosts. No fixed infrastructure such as base stations as mobile switching .Nodes within each other radio range communicate directly via wireless links while these which are far apart rely on other nodes to relay messages. Node mobility causes frequent changes in topology.

1.1 Related Work

Many routing protocols have been proposed, but few comparisons between the different protocols have been made. There exist some other simulation results that have been done on individual protocols. These simulations have however not used the same metrics and are therefore not comparable with each other.

II. Ad-Hoc Routing Protocols

2.1 Desirable properties

1. Distributed operation: The protocol should not be dependent on a centralized controlling node.
2. Loop free: To improve the overall performance, we want the routing protocol to guarantee that the routes supplied are loop-free. This avoids any waste of bandwidth or CPU consumption.
3. Demand based operation: It means that the protocol should only react when needed and that the protocol should not periodically broadcast control information.
4. Unidirectional link support: The radio environment can cause the formation of unidirectional links.
5.Security: i.e. authentication and encryption.
6. Power conservation
7 . Multiple routes: To reduce the number of reactions to topological changes and congestion multiple routes could be used.
8. Quality of service support.

None of the proposed protocols from MANET have all these properties, but it is necessary to remember that the protocols are still under development and are probably extended with more functionality. The primary function is to find a route to the destination, not to find the best/optimal/shortest-path route.

2.2 MANET

IETF has a working group named MANET(Mobile Ad-hoc Networks) that is working in the field of ad-hoc networks. They are currently developing routing specifications for ad-hoc IP networks that support scaling to a couple of hundred nodes.
Currently they have 8 routing protocol draft:

1. AODV- Ad-hoc On Demand Distance Vector
2. ZRP- Zone Routing Protocol
3. TORA/IMEP- Temporary Ordered Routing Algorithm/ Internet MANET Encapsulation Protocol
4. DSR- Dynamic Source Routing
5. CBRP-Cluster Based Routing Protocol
6. CEDAR-Core Extraction Distributed Ad hoc Routing
7. AMRoute- Ad-hoc Multicast Routing Protocol
8. OLSR-Optimized Link State Routing Protocol

Of these proposed protocols we have chosen to analyze AODV, DSR, ZRP, CBRP and TORA theoretically. We have also analyzed DSDV, which is a proactive approach, as opposes to other reactive protocols. We have not

realized AMRoute because it is a multicast routing protocol, neither CEDAR because it is primary a QoS routing protocol, nor OLSR, because it was submitted as an internet draft so late. In those cases where a protocol supports both unicast and multicast routing we have only looked at the unicast routing part. Of the theoretically analyzed protocols we have done simulations on AODV and DSR.

2.3. Destination Sequenced Distance Vector- DSDV

DSDV is a hop-by-hop distance vector routing protocol that in each node has a routing table that for all reachable destinations stores the next-hop and number of hops for that destination. Like distance-vector, DSDV requires that each node periodically broadcast routing updates. The advantage with DSDV over traditional distance vector protocols is that DSDV guarantees loop-freedom.

Properties: Because DSDV is dependent on periodic broadcasts it needs some time to converge before a route can be used. The periodic updates also add a large amount of overhead into the network.

2.4. Ad-hoc On Demand Distance Vector- AODV

AODV routing protocol enables multi-hop routing between participating mobile nodes wishing to establish and maintain an ad-hoc network. AODV is based upon the distance vector algorithm. The difference is that AODV is reactive, as opposed to proactive protocols like DV, i.e. AODV only requests a route when needed and does not require nodes to maintain routes to destinations that are not actively used in communications. As long as the endpoints of a communication connection have valid routes to each other, AODV does not play any role.

Features of this protocol include loop freedom and that link breakages cause immediate notifications to be sent to the affected set of nodes, but only that set. Additionally, AODV has support for multicast routing and avoids the Bellman Ford counting to infinity problem. The use of destination sequence numbers guarantees that a route is "Fresh".

The algorithm uses different messages to discover and maintain links.

Properties: The advantage with AODV compared to classical routing protocols like distance vector and link state is that AODV has greatly reduced the number of routing messages in the network.

AODV is also routing in the more traditional sense compared to for instance source routing based proposals like DSR. The advantage of it is that connections from the ad-hoc network to a wired network like the Internet is most likely easier.

The sequence numbers in AODV represents the freshness of a route and is increased when something happens in the surrounding area.

AODV only support one route for each destination.

AODV uses hello messages at IP-Level. This means that AODV does not need support from the link layer to work properly.

AODV does not support unidirectional links.

2.5. Dynamic Source Routing- DSR

DSR also belongs to the class of reactive protocols and allows nodes to dynamically discover a route across multiple network hops to any destination. Source routing means that each packet in its header carries the complete ordered list of nodes through which the packets must pass. DSR uses no periodic routing messages. The 2 basic modes of operation in DSR are route discovery and route maintainance.

Properties: DSR uses key advantage of source routing.

This protocol has the advantage of learning routes by scanning for information in packets that are received.

DSR also has support for unidirectional links by the use of piggybacking the source route a new request.

2.6. Zone Routing Protocols-ZRP

ZRP is a hybrid of a reactive and a proactive protocol. It divides the network into several routing zones and specifies two totally detached protocols that operate inside and between the routing zones.

The Intrazone routing Protocol (IARP) operates inside the routing zone and learns the minimum distance and routes to all the nodes within the zone.

The second protocol, the Interzone Routing Protocol (IERP) is reactive and is used for finding routes between different routing zones.

Properties: ZRP is very interesting protocol and can be adjusted of its operation to the current operational conditions e.g. change the routing zone diameter.

This protocol uses advantage of both proactive and reactive schemes.

It also limits the propagation of information about topological changes to the neighborhood of the change only(as opposed to a fully proactive scheme, which would basically flood the entire network when a change in topology occurred.)

2.7. Temporally-Ordered Routing Algorithm- TORA

The Temporally-Ordered Routing Algorithm (TORA) is an algorithm for routing data across Wireless Mesh Networks or Mobile ad-hoc networks. It is a distributed routing protocol.

TORA can be separated into 3 basic functions: creating routes, maintaining routes and erasing routes. The creation of routes basically assigns directions to links in an undirected network or portion of the network, building a directed acyclic graph(DAG).

Maintaining routes refers to reacting to topological changes in the network in a manner such that routes to the destination are re-established within a finite time, meaning that its directed portions return to a destination-oriented graph within a finite time.

Properties: The protocols underlying link reversal algorithm will react to link changes through a simple localized single pass of the distributed algorithm.

2.8. Internet MANET Encapsulation Protocol – IMEP

IMEP is a protocol designed to support the operation of many routing protocols in Ad-hoc networks. The idea is to have a common general protocol that all routing protocols can make use. It incorporates many common mechanism that the upper layer protocol may need.

It also provides an architecture for MANET router identification, interface identification and addressing. IMEPs purpose is to improve overall performance by reducing the number of control messages and to put common functionality into one unified, generic protocol useful to all upper layer routing protocols.

Of the currently proposed protocols, only TORA, and OLSR use IMEP. It must however be noted that TORA and IMEP were designed by the same author.

Properties: It adds another layer to the protocol stack.

IMEP generates lot of overhead, mainly because of IMEPs neighbor discovery mechanism that generates atleast one hello message per secong, but also because of the reliable in-order delivery of the packets that IMEP provides.

2.9. Cluster Based Routing Protocol- CBRP

The idea behind CBRP is to divide the nodes of an ad-hoc network into a number of overlapping or disjoint clusters. One node is selected as cluster head for each cluster. This cluster head maintains the membership information for the cluster. Inter cluster routes are discovered dynamically using the membership information.

CBRP is based on source routing, similar to DSR. CBRP is like the other protocols fully distributed.

Properties: It has a route discovery and route removal operation that has a lot in common with DSR and AODV.

The clustering is probably a very good approach when dealing with large ad-hoc networks.

2.10. Comparison between ad-hoc routing protocols

	DSDV	**AODV**	**DSR**	**ZRP**	**TORA/ IMEP**	**CBRP**
Loop free	**Yes**	Yes	**Yes**	**Yes**	**No, short Lived loops**	**Yes**
Multiple routes	**No**	**No**	**Yes**	**No**	**Yes**	**Yes**
Distributed	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
Reactive	**No**	**Yes**	**Yes**	**Partially**	**Yes**	**Yes**
Unidirectional link support	**No**	**No**	**Yes**	**No**	**No**	**Yes**
QoS support	**No**	**No**	**No**	**No**	**No**	**No**
Multicast	**No**	**Yes**	**No**	**No**	**No**	**No**
Security	**No**	**No**	**No**	**No**	**No**	**No**
Power conservation	**No**	**No**	**No**	**No**	**No**	**No**
Periodic broadcast	**Yes**	**Yes**	**No**	**Yes**	**Yes (IMEP)**	**Yes**
Requires reliable or sequenced data	**No**	**No**	**No**	**No**	**Yes**	**No**

III. Simulation Environment

The simulator we have used to simulate the ad-hoc routing protocols in is the Network Simulator 2 (ns). To simulate the mobile wireless radio environment we have used a mobility extension to ns.

3.1. Network Simulator

Network Simulator 2 is the result of an on-going effort of research and development. It is a discrete event simulator targeted as network research. It provides substantial support for simulation of TCP, routing and multicast protocols.

ns (from **network simulator**) is a name for series of discrete event network simulators, specifically **ns-1**, **ns-2** and **ns-3**. All of them are discrete-event network simulator, primarily used in research[4] and teaching. ns-3 is free software, publicly available under the GNU GPLv2 license for research, development, and use.

Ns2 is a package of tools that simulates behavior of networks that

1.Create Network Topologies

2.Log events that happen under any load analyze events to understand the network behavior

Ns-2 is written in C++ and an Object oriented version of Tcl called OTcl.

OTcl: (short for MIT Object Tcl,)

It is an extension to Tcl/Tk for object-oriented programming.

- Used to build the network structure and topology which is just the surface of your simulation;
- Easily to configure your network parameters;
- Not enough for research schemes and protocol architecture adaption.

C++: Most important and kernel part of the NS2

- To implement the kernel of the architecture of the protocol designs;
- From the packet flow view, the processes run on a single node;
- To change or "comment out" the existing protocols running in NS2;
- Details of your research scheme.

2 requirements of the simulator are:

– Detailed simulation of Protocol: Run-time speed;

– Varying parameters or configuration: easy to use.

The NS-2 architecture is composed of five parts:

- Event scheduler
- Network components
- Tclcl
- OTcl library
- Tcl 8.0 scipt language

NS models all network elements through a class hierarchy. In this class hierarchy, the TclObject class is the superclass of all OTcl library objects (network components, event scheduler, timers and others). A subclass of TclObject, NsObject again is the superclass of all basic network component objects that handle packets. Network objects, such as nodes and linkes can then be composed of this basic network components. Moreover, NsObject has two subclasses, Connector and Classifier. Connector is the superclass of all basic network objects that have only one output data path and Classifier is the superclass of all switching objects that have possible multiple output data paths.

3.2 Mobility extension

Mobility extensions to ns are:

1. Wireless mobility extension
2. Mobility support, mobile IP and wireless channel support

The version of the extension that we have worked with adds the following features to the network simulator.

NODE MOBILITY

Each mobile node is an independent entity that is responsible for computing its own position and velocity as a function of time. Nodes move around according to a movement pattern specifid at the beginning of the simulator.

REALISTIC PHYSICAL LAYERS

Propagation models are used to decide how far packets can travel in air. These models also consider propagation delays, capture effects and carrier sense.

MAC 802.11

It handles collision detection, fragmentation and acknowledgements. It also used to detect transmission errors. It is a CSMA/CA protocol

ADDRESS RESOLUTION PROTOCOL

ARP is implemented. It translates IP-address to hardware MAC address.
AD-HOCKEY
It is an application that makes it possible to visualize the mobile nodes as they move around and send/receive packets. It can also be used as a scenario generator tool to create the input files necessary for the simulations.
RADIO NETWORK INTERFACES
It is a model of the hardware that actually transmits the packets onto the channel with a certain power and modulation scheme.
TRANSMISSION POWER
The radius of the transmitter with an omni-directional antenna is about 250m in this extension.
ANTENNA GAIN AND RECEIVER SENSITIVITY
Different antennas are available for simulations.
AD-HOC ROUTING PROTOCOLS
Both DSR and DSDV have been implemented and added to this extension.

3.2.1. Shared media

The extension is based on a shared media model (Ethernet in the air). This means that all nodes have one or
more network interfaces that are connected to a channel.

3.2.2. Mobile node

Each mobile node makes use of a routing agent for the purpose of calculating routes to other nodes in the ad-hoc network.

3.3. Simulation overview

Basically it consists of generating the following input files to ns:

1. A scenario file that describes the movement pattern of nodes.
2. A communication file that describe the traffic in the network.

These files are generated by drawing them by hand using the visualization tool Ad-hockey or by generating completely randomized movement and communication patterns with a script.
These files are then used for simulation and as a result from this, a trace file is generated as output. Prior to the simulation, the parameters that are going to be traced during the simulation must be selected. The trace file can then be scanned and analyzed for the various parameters that we want to measure. This can be used as data for plots with for instance Gnupplot. The trace file can also be used to visualize the simulation run with either Ad-hockey or Network animator.

3.4. Modifications

To be able to use ns for the simulation, we had to do some modifications. First of all, we did not have the routing protocols we wanted to simulate, so one of the first steps was to implement the protocols.

3.4.1. AODV

The changes that affect the unicast routing part is primarily:

1. Reduced or complete elimination of hello messages.
2. Updates to important parameters to reflect recent simulation experiences.

The DSR implementation that was included in the mobility extension used a sendbuffer that buffered all packets that the application sent while the routing protocol searched for a route.

3.4.2. DSR

The DSR implementation that came with the extension uses promiscuous mode(i.e. eavesdropping), which means that the protocol learns information from packets that it overhears.

3.4.3. DSDV

The extension also included an implementation of the DSDV protocol. This is an actually 2 implementations that handle the triggred update a little different. In first version only a new metric for a destination causes a triggered update to be sent. In the 2^{nd} version, a new sequence number for a destination causes a triggered update to be sent.

3.4.4. Flooding

To have some sort of cleverness and avoiding data to bounce back and forth we use a sequence number in each packet, which is incremented for each new packet.

3.4.5. The simulator

To the actual simulator, we have added some new features:
Obstacles: The Ad-hockey allows the user to place obstacles(lines and boxes) into the scenario.
Version management: To allow us to test different versions of one protocol simultaneous.

1. AODV1 =AODV with only hello messages.
2. AODV2 =AODV with only MAC layer feedback.
3. AODV3 =AODV with both hello messages and MAC layer feedback.
4. DSR1 = DSR with eavesdropping.
5. DSR2 = DSR without eavesdropping.

IV. Simulation Study

The simulations were conducted on an Intel PC with a Pentium-2 processor at 400MHz, 128 Mbytes of RAM running FreeBSD.

4.1. Measurements

There are 2 main performance measures that are substantially affected by the routing algorithm

1. The average end-to-end throughput(quantity of service)
2. The average end-to-end delay (quality of service)

4.2. Simulation setup

We have done 4 types of simulations:

1. Mobility simulations: we vary the mobility to see how it affects the different metrics that we are measuring.
2. Offered load simulations: we vary the load that we offer the network to see how the protocols behave when for instance the load is high
3. Network size simulations: we vary the number of nodes in the network.
4. Realistic simulations: to test certain characteristics of the protocols.

In all simulations except realistic simulations, we have used a randomized scenario. The randomized scenario have different parameters that affect the movement patterns. The parameters that can be changed are:

1. Maximum speed
2. Number of nodes
3. Environment size
4. Simulation time
5. Pause time

First of all every node stands still for pause time seconds. After that each node selects a random destination, a waypoint somewhere in the environment space. Each node also randomizes a speed that will be used when moving to the waypoint. This speed is randomized uniformly in the interval 0 to maximum speed. Every time a node reaches a waypoint, this procedure will be repeated.

4.3. Mobility simulations

4.3.1 Setup

The simulations where we varied the mobility done by randomizing scenario files. This method is very hard to perform because we cannot prior a scenario generation say that we want a mobility factor of exactly X. instead we used the maximum speed parameter to control the scenario.
By increasing the maximum speed in the scenario generation, the mobility will also increase.

Parameters used are:

Transmission range	250m
Bandwidth	2Mbit
Simulation time	250s
Number of nodes	50
Pause time	1s
Environment size	1000*1000m
Traffic type	constant bit rate
Packet rate	5packets/s

Packet size	64 byte
Number of flows	15

Packets received:

We see that the fraction of received packets for DSR versions is very large even for high mobility. A reason for the higher fraction received packets for DSR compared to AODV is that DSR allows packets to stay in the send buffer for as long as 30s, AODV only 8s. it must however be noted that AODV draft does not specify how long a packet is allowed to stay in the sendbuffer.

When comparing these results with DSDV it can clearly be seen that a proactive approach is not acceptable at all when the mobility increases.

Delays:
Also it can be shown that of the different versions AODV with only hello messages has lowest delay on the data packets that are received. The reason is not that it finds routes faster or that the routes are shorter or more optimal, instead AODV with only hello messages is the AODV version that gets significantly fewest packets through the network.
AODV with both hello messages an MAC layer support has a slightly lower delay than AODV with only MAC layer support.
Both DSR versions show a tendency to get higher delay when mobility is increased.

Throughput:
It can be seen that both DSR and AODV versions with link layer support have almost identical throughput.

Overhead:
DSR does not include the data packets in the number of control packet calculations, only the extra byte overhead from these packets is included.

Optimal path:
It can be shown that DSDV has the highest degree of optimality.

4.3.2. Summary mobility simulations

The protocols that have link layer support for link breakage detection will be more stable. The fraction of packets received for these protocols is almost constant at 95% even when mobility increases. These protocols include both DSR versions and the two AODV versions that have link layer support. Protocols that are highly dependent on periodic broadcasts show a rather poor result, only little more than 50% of the packet are received when mobility is increased.

4.4. Offered load simulations

The offered load simulations where done by varying the load that we offer the network. We had mainly 3 parameters to adjust the offered load:

1. Packet size
2. Number of CBR flows
3. Rate at which the flows are sending

The performance of the protocols differs slightly during different network loads. The most apparent difference is the byte overhead. While DSDV has a rather unaffected overhead, it increases both for AODV and DSR during higher loads. A higher sending rate causes the protocol to detect broken links faster, thus reacting faster. This leads to a slight increase in control packets, which also affects the byte overhead. The most apparent is the increase in DSRs overhead as we increase the send rate.

4.5. Network size simulations

We decrease number of nodes, which meant that connectivity also decreased; each node had a fewer neighbors. The results from these simulations did not give any new information regarding the performance of the protocols. The relative difference between the protocols was the same.

4.6. Realistic scenarios

The randomized simulations has some problems:

1. It is hard to identify situations in which the protocols fail or have problems.

2. It has no connection to a real life situation.
3. It may favor complex protocols, while in real life scenarios simpler protocols can find the routes almost effectively.

It is therefore also very interesting to see how these protocols behave in more realistic scenarios. The realistic scenarios do not give a full picture of how the protocols behave generally. Instead they give some sense of weak points in the protocols. The 3 basic types of scenarios that we have done simulations on are:

1. Conference type, with low movement factor.
2. Event coverage type, with fairly large movement factor. Could for instance be reporters trying to interview politicians.
3. Disaster area, with some relatively slow nodes and some very fast nodes.(mounted on a car or a helicopter).

It can be shown that in realistic scenarios DSR show the best performance results overall. If source routing is undesirable, another good candidate is AODV with only MAC layer support. It has a slightly higher packet overload, but an overall good delivery ratio.

4.7. Improvements

Our proposal is to implement a good protocol that is a combination of source routing and distance vector. Source routing should be used in route discovery and route maintenance phases. These phases would also include that the routing tables were set up accordingly during the propagation of requests and replies. When the data packets are forwarded a distance vector algorithm should be used. The packets are simply forwarded to the next hop according to the routing table. This in combination with that the protocol stores several routes for each destination would probably mean a protocol with a performance that is even better than the protocols that have been simulated in this.

V. Conclusions

The simulations have shown that there certainly is a need for a special ad-hoc routing protocol when the mobility increase. It is however necessary to have some sort of feedback from the link layer protocol like IEEE MAC 802.11 when links go up and down or for neighbor discovery. The simulations have shown that more conventional types of protocols like DSDV have a drastic decrease in performance when mobility increases and are therefore not suitable for mobile ad-hoc networks.

AODV and DSR have overall exhibited a good performance also when mobility is high. DSR is however based on source routing, which is not desirable in ordinary forwarding of data packets because of large byte overhead. In these situations a hop by hop routing protocol like AODV is more desirable. A combination of AODV and DSR could therefore be a solution with even better performance than AODV and DSR.

Also DSR has the best performance in realistic scenarios, but the large byte overhead caused by the source route in each packet makes AODV a good alternate candidate. It has almost a good performance.

VI. Further studies

There are many issues that could be subject to further studies.
First of all, the simulator environment could be improved.
Secondly, There are many issues related to ad-hoc networks that could be subject to further studies.

References

[1] Dimitri Bertsekas and Robert Gallager, "Data Networks-2nd ed". Prentice Hall, New Jersey, ISBN 013-200916-1
[2] Bommaiah, McAulley and Talpade. AMRoute, "Adhoc Multicast Routing Protocols", Internat draft, drafttalpade-manet-amroute-00.txt, august 1998.
[3] Josh Broch, David B. Johnson, David A. Maltz," The Dynmic Source Routing Protocol for mobile adhoc networks". Internet draft, draft-ietf-manet-dsr-00.txt.
[4] Kevin Fall and Kannan Varadhan, "ns ntes and documentation". The VINT project, UC Berkley, LBL, USC/ISI, and Xerox PARC.
[5] IEEE Computer society LAN MAN Standards Committee ," Wireless LAN Medium Access Control(MAC) and Physical Layer (PHY) Specifications", IEEE std 802.11-1997. The Institute of Electrical and Engineers, New York.
[6] Mingling Jiang, Jingang Li and Yong Chiang Tay," Cluster Based Routing Protocol(CBRP) Functional SPECIFICATION". Internet draft, draft-ietf-manet-cbrp-spec-00.txt.
[7] David B Johnson and David A. Maltz," Dynaamic source routing in ad hoc wireless networks". In Mobile computing, edited by Tomasz Imielinski and Hank Korth, chapter 5, pages 153-181. Kluner Academic Publication.
[8] David B Johnson and David A. Maltz,"Security architecture for the internet protocol", Internet draft,draft-ietf-ipsec-arch-sec-07.txt.