International Journal of Research in
Engineering & Technology (IJRET)
ISSN 2321-8843
Vol. 1, Issue 2, July 2013, 143-150

PERFORMANCE ANALYSIS OF AODV AND DSDV IN MANETs

K. THAMARAISELVI

Assistant Professor, Department of Computer Science Engineering, KSR College of Engineering, Tiruchengode, Tamil Nadu, India

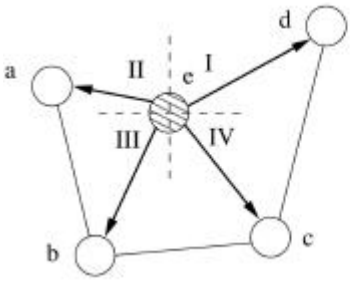
ABSTRACT

An ad-hoc mobile network is a collection of mobile nodes that are dynamically and arbitrarily located in such a manner that the interconnections between nodes are capable of changing on a continual basis. Mobile Ad-Hoc networks are not new to computer science, but the concept of a well organized routing simulator that can demonstrate routing protocols used in Ad-Hoc networks a reality. This simulator will be capable of demonstrating two different routing protocols initially, but will also have the room to expand its capabilities. The simulator will be able to play out many real life scenarios, allowing users to seek out a routing protocol that can optimize the Mobile Ad-Hoc Network experience. While selecting a route, nodes with battery power greater than the threshold will only be considered. It would then go on to compute the minimum power-cost route. We propose an alternative ERS scheme to support reactive and proactive protocols such as DSDV and AODV, and it is called Blocking Expanding Ring Search (Blocking-ERS for short) and also Bellman-Ford algorithm is used as another alternative scheme. The Bellman-Ford algorithm is used to find the best shortest path. The Blocking-ERS integrates, instead of TTL sequences, a newly adopted control packet, stop instruction and a hop number (H) to reduce the energy consumption during route discovery stage.

KEYWORDS: Broadcasting, AODV, DSDV, Ad-Hoc Mobile Network, Simulation

INTRODUCTION

Broadcasting is the most frequently used operation in mobile ad hoc networks (MANETs) for the dissemination of data and control messages in many applications. Usually, a network backbone is constructed for efficient broadcasting to avoid the broadcast storm problem caused by simple blind flooding, where only selected nodes that form the virtual backbone, called forwarding nodes, forward data to the entire network.



(a)

	A	B	C	D
a	1	1	0	0
b	1	1	1	0
c	0	1	1	1
d	0	0	1	1
e	1	1	1	1

(b)

	P_1	P_2
I	1	1
II	1	1
III	0	1
IV	1	0

(c)

Figure 1: (a) A Sample Network, (b) Neighbour Reception Table of Node E, and (c) Transmission Table of Node E Using Coding and Directional Antennas

Network coding is defined as allowing intermediate nodes to process the incoming information flows. When a forwarding node, chosen by a certain approach, needs to forward several messages to all of its neighbours, while some neighbours already have some of the messages, this node can combine some of the messages to reduce the number of

forwarding, and each neighbour can still get every message via decoding. Using directional antennas, the Omni-directional transmission range of each node can be divided into several sectors and the transmission can be performed only in selected sectors. Therefore, by reducing the total number of transmission sectors of the forwarding nodes in the network, the interference can be alleviated, as well as the energy consumption.

RELATED WORKS AND PRELIMINARIES

Broadcast in MANET's

Both probabilistic and deterministic approaches have been proposed for efficient broadcasting. Probabilistic approaches use limited neighbourhood information (local information) and require relatively high broadcast redundancy to maintain an acceptable delivery ratio. Deterministic approaches select a few forwarding nodes to achieve full delivery. Most of these approaches are localized, where each node determines its status (forwarding or no forwarding) based on its h-hop neighbourhood information (for small values of h, such as 2 or 3). The decision of forwarding nodes can be made under both static and dynamic local views. In the static approaches, only topology information is considered, whereas in dynamic ones, broadcast state information of the neighbourhood is also piggybacked. More efforts have been made on developing efficient broadcast approaches. In an integer programming approach and improved heuristic algorithms, were proposed. In a broadcast scheme that combines the advantages of both probability and counter based approaches, was developed.

Network Coding

Network coding can be used to allow the intermediate nodes to combine packets before forwarding. Therefore, network coding can be used for efficient broadcasting by reducing the total number of transmissions. Fragile et al. quantified the energy savings that network coding has the potential to offer in broadcasting. They also proposed an implementable method for performing the network coding and addressed some practical issues such as setting the forwarding factor and managing generations. Liu et al. [14] derived bounds for the throughput benefit ratio, the ratio of the throughput of the optimal network coding scheme to the throughput of the optimal non-coding flow scheme. They used the general physical communication model found in [9]. In [3], it was shown that designing appropriate MAC scheduling algorithms is critical for achieving the throughput gains expected from network coding and frame framework

Directional Antennas

Two techniques are used in smart antenna systems that form directional transmission/reception beams: switched beam and steerable beam. The most popular directional antenna model is ideally sector zed, as where the effective transmission range of each node is equally divided into K non overlapping sectors, where one or more such sectors can be switched on for transmission or reception. The channel capacity when using directional antennas can be improved and the interference can be reduced. Steerable beam systems can adjust the bearing and width of a beam to transmit to, or receive from, certain neighbours. The corresponding antenna mode is an adjustable cone. In practical systems, antenna beams have irregular shapes due to the existence of side lobes, which may cause inaccurate estimations.

BROADCASTING WITH BLOCKING EXPANDING RING SEARCH AND BELLMAN-FORD ALGORITHM

The flooding protocol described above has a scalability problem, because whenever a node requests a route, it sends a message that passes through potentially every node in the network. When the network is small, this is not a major concern. However, when the network is large, this can be extremely wasteful, especially if the destination node is relatively

close to the RREQ originator. Preferably, we would like to set the TTL value on the RREQ message to be just large enough so that the message reaches the destination, but no larger. However, it is difficult for a node to determine this optimal TTL without prior global knowledge of the network. To solve this problem, I have implemented an expanding ring search algorithm, which works as follows. When a node initiates a route request, it first broadcasts the RREQ message with a small TTL value (say, 1). If the originating node does not receive a RREP message within a certain period of time, it rebroadcasts the RREQ message with a larger TTL value (and also a new RREQ identifier to distinguish the new request from the old ones). The node continues to broadcast messages with increasing TTL and RREQ ID values until it receives a route reply. If the TTL values in the route request have reached a certain threshold, and still no RREP messages have been received, then the destination is assumed to be unreachable, and the messages queued for this destination are thrown out.

BELLMAN-FORD ALGORITHM

Bellman-Ford algorithm solves the single-source shortest-path problem in the general case in which edges of a given digraph can have negative weight as long as G contains no negative cycles. This algorithm, like Dijkstra's algorithm uses the notion of edge relaxation but does not use with greedy method. Again, it uses d[u] as an upper bound on the distance d[u, v] from u to v. The algorithm progressively decreases an estimate d[v] on the weight of the shortest path from the source vertex s to each vertex v in V until it achieve the actual shortest-path. The algorithm returns Boolean TRUE if the given digraph contains no negative cycles that are reachable from source vertex s otherwise it returns Boolean FALSE

BELLMAN-FORD (G, w, s)

- INITIALIZE-SINGLE-SOURCE (G, s)
- for each vertex i = 1 to V[G] - 1 do
- for each edge (u, v) in E[G] do
- RELAX (u, v, w)
- For each edge (u, v) in E[G] do
- if d[u] + w(u, v) < d[v] then
- return FALSE
- return TRUE

Analysis

- The initialization in line 1 takes Θ(v) time
- For loop of lines 2-4 takes O (E) time and For-loop of line 5-7 takes O (E) time.

Thus, the Bellman-Ford algorithm runs in O (E) time.

Protocol Design

My implementation of protocols is based on a recent draft of the specification. I have implemented all the essential functionality of AODV and DSDV, including:

- RREQ and RREP messages (for route discovery)
- RERR messages, HELLO messages, and precursor lists (for route maintenance)

- Sequence numbers
- Hop counts
- Blocking Expanding ring search and Bellman-Ford Algorithm.

Some functionality described in the specification has been omitted, such as Gratuitous RREP messages, RREP acknowledgements, and multicast support, because they are either not essential to the algorithm, or inapplicable given our network model.

AODV and DSDV Route Discovery

When a node needs to determine a route to a destination node, it floods the network with a Route Request (RREQ) message. The originating node broadcasts a RREQ message to its neighbouring nodes, which broadcast the message to their neighbours, and so on. To prevent cycles, each node remembers recently forwarded route requests in a route request buffer. As these requests spread through the network, intermediate nodes store reverse routes back to the originating node. Since an intermediate node could have many reverse routes, it always picks the route with the smallest hop count. When a node receiving the request either knows of a "fresh enough" route to the destination (see section on sequence numbers), or is itself the destination, the node generates a Route Reply (RREP) message, and sends this message along the reverse path back towards the originating node.

As the RREP message passes through intermediate nodes, these nodes update their routing tables, so that in the future, messages can be routed though these nodes to the destination. Notice that it is possible for the RREQ originator to receive a RREP message from more than one node. In this case, the RREQ originator will update its routing table with the most "recent" routing information; that is, it uses the route with the greatest destination sequence number. (See section on sequence numbers).

The Route Request Buffer

In the flooding protocol described above, when a node originates or forwards a route request message to its neighbours, the node will likely receive the same route request message back from its neighbours. To prevent nodes from resending the same RREQs (causing infinite cycles), each node maintains a route request buffer, which contains a list of recently broadcasted route requests. Before forwarding a RREQ message, a node always checks the buffer to make sure it has not already forwarded the request. RREQ messages are also stored in the buffer by a node that originates a RREP message. The purpose for this is so a node does not send multiple RREPs for duplicate RREQs that may have arrived from different paths.

The exception is if the node receives a RREQ with a better route (i.e. smaller hop count), in which case a new RREP will be sent. Each entry in the route request buffer consists of a pair of values: the address of the node that originated the request, and a route request identification number (RREQ id). The RREQ id uniquely identifies a request originated by a given node. Therefore, the pair uniquely identifies a request across all nodes in the network. To prevent the route request buffers from growing indefinitely, each entry expires after a certain period of time, and then is removed. Furthermore, each node's buffer has a maximum size. If nodes are to be added beyond this maximum, then the oldest entries will be removed to make room.

Sequence Numbers

Each destination (node) maintains a monotonically increasing sequence number, which serves as a logical time at

that node. Also, every route entry includes a destination sequence number, which indicates the "time" at the destination node when the route was created. The protocol uses sequence numbers to ensure that nodes only update routes with "newer" ones. Doing so, we also ensure loop- freedom for all routes to a destination. All RREQ messages include the originator's sequence number, and its (latest known) destination sequence number. Nodes receiving the RREQ add/update routes to the originator with the originator sequence number, assuming this new number is greater than that of any existing entry.

If the node receives an identical RREQ message via another path, the originator sequence numbers would be the same, so in this case, the node would pick the route with the smaller hop count. If a node receiving the RREQ message has a route to the desired destination, then we use sequence numbers to determine whether this route is "fresh enough" to use as a reply to the route request.

To do this, we check if this node's destination sequence number is at least as great as the maximum destination sequence number of all nodes through which the RREQ message has passed. If this is the case, then we can roughly guess that this route is not terribly out-of- date, and we send a RREP back to the originator. As with RREQ messages, RREP messages also include destination sequence numbers. This is so nodes along the route path can update their routing table entries with the latest destination sequence number.

LINK MONITORING & ROUTE MAINTENANCE

Each node keeps track of a precursor list, and an outgoing list.

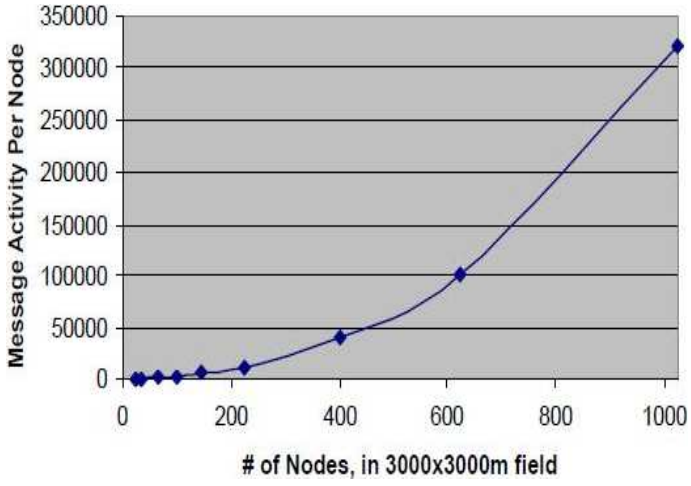


Figure 2: The Following Flow Chart Summarizes the Action of an AODV and DSDV Node when Processing an Incoming Message. HELLO Messages are Excluded from the Diagram for Brevity

A precursor list is a set of nodes that route through the given node .Each node periodically sends HELLO messages to its precursors. A node decides to send a HELLO message to a given precursor only if no message has been sent to that precursor recently. Correspondingly, each node expects to periodically receive messages (not limited to HELLO messages) from each of its outgoing nodes. If a node has received no messages from some outgoing node for an extended period of time, then that node is presumed to be no longer reachable.

Whenever a node determines one of its next- hops to be unreachable, it removes all affected route entries, and generates a Route Error (RERR) message. This RERR message contains a list of all destinations that have become unreachable as a result of the broken link. The node sends the RERR to each of its precursors. These precursors update their routing tables, and in turn forward the RERR to their precursors, and so on the action of an AODV and DSDV node when processing an incoming message. HELLO messages are excluded from the diagram for brevity.

PERFORMANCE

One of the goals in simulating AODV and DSDV is to determine how well it scales. Both the protocol is used to find the shortest path. How does the protocol performance vary with respect to the number of nodes in the network? Attempting to answer this question, I conducted experiments measuring message activity, varying the number of nodes. I compute total message activity as the total number of AODV messages sent and received at each node.

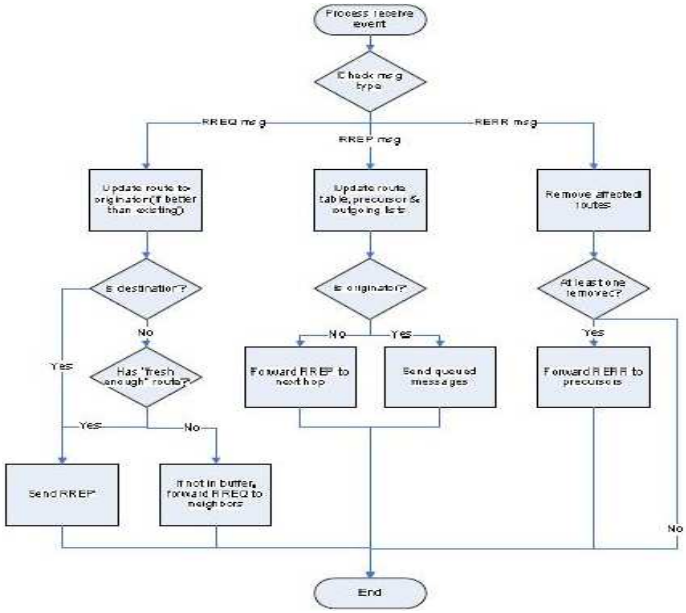


Figure 3: Node Density Increases

In this first experiment, I attempted to determine the effect of increasing the density of mobile nodes within a fixed area. I varied the number of nodes, from 4 to 1024 nodes, within a fixed field (3000x3000 meters). Varying the number of nodes can be accomplished in two basic ways.

One is by varying field size, keeping node density constant. Another is by keeping the field size constant and increasing the density. I performed experiments using both these approaches. It is important to count both sent and received messages, as they will generally differ, for not all sent messages are received, while some messages are received many times (broadcasts). Additionally, I measured memory usage and elapsed time.

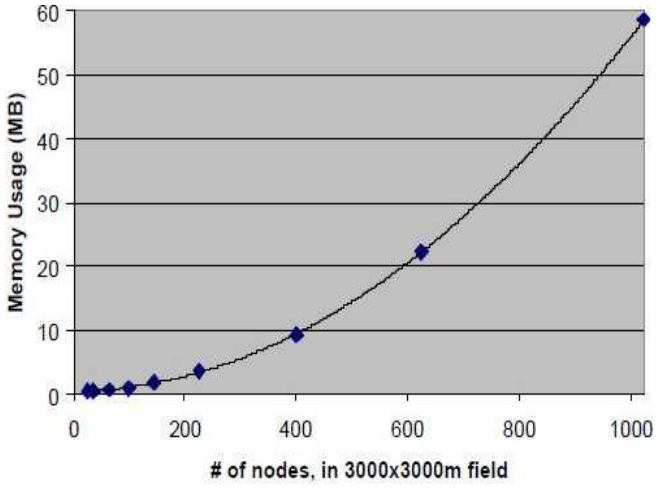


Figure 4: Memory Usage Grows Quadratically. Best-Fit Curve: M=.0561n2+.593n+543

In all my simulated experiments, each node sent messages to random destinations at an average rate of one

message per minute. The nodes sent messages for ten minutes, and then statistics were recorded one minute afterwards

CONCLUSIONS

Mobile Ad-Hoc networks are not new to computer science, but the concept of a well organized routing simulator that can demonstrate routing protocols used in Ad-Hoc networks a reality. This simulator will be capable of demonstrating two different routing protocols initially, but will also have the room to expand its capabilities. The simulator will be able to play out many real life scenarios, allowing users to seek out a routing protocol that can optimize the Mobile Ad-Hoc Network experience. While selecting a route, nodes with battery power greater than the threshold will only be considered. It would then go on to compute the minimum power-cost route. We propose an alternative ERS scheme to support reactive protocols such as DSDV and AODV, and it is called Blocking Expanding Ring Search (Blocking-ERS for short). The Blocking-ERS integrates, instead of TTL sequences, a newly adopted control packet, stop instruction and a hop number (H) to reduce the energy consumption during route discovery stage.

FUTURE WORK OF THIS PROJECT

We present a family of energy-conserving flooding protocols capable of supporting both reactive and proactive routing approaches, as well as network applications that rely on flooding. Based on realistic simulation models, these protocols show significant energy-conserving potential. Future work will focus on methods for balancing the protocols' overhead and relay optimality to further enhance their efficiency.

ACKNOWLEDGEMENTS

I also express my thanks to my institution, parents, and friends, well wishers for their encouragement and best wishes in the successful completion of this dissertation

REFERENCES

1. R. Ahlswede, N. Cai, S.R. Li, and R.W. Yeung, "Network Information Flow," IEEE Trans. Information Theory, vol. 46, no. 4, pp. 1204-1216, July 2000.
2. D. Bein and S.Q. Zheng, "An Effective Algorithm for Computing Energy-Efficient Broadcasting Trees in All-Wireless Networks," Proc. IEEE Int'l Conf. Distributed Computing Systems (ICDCS), 2008.
3. P. Chaporkar and A. Proutiere, "Adaptive Network Coding and Scheduling for Maximizing Throughput in Wireless Networks," Proc. ACM MobiCom, 2007.
4. F. Dai and J. Wu, "Efficient Broadcasting in Ad Hoc Wireless Networks Using Directional Antennas," IEEE Trans. Parallel and Distributed Systems, vol. 17, no. 4, pp. 335-347, Apr. 2006.
5. F. Delgosha, E. Ayday, K. Chan, and F.Fekri,"SecuritServices in Wireless Sensor Networks Using SparseRandomCoding,"Proc. IEEE INFOCOM, 2007.
6. A. Dimakis, B. Godfrey, M. Wainwright, and K. Ramchandran, "Network Coding for Distributed Storage in Peer-to-Peer Networks," Proc. IEEE INFOCOM, 2007.
7. C. Fragouli, J. Widmer, and J.-Y.L. Boudec, "A Network Coding Approach to Energy Efficient Broadcasting: From Theory to Practice," Proc. IEEE INFOCOM, 2006.
8. S. Gollakota and D. Katabi, "ZigZag Decoding: Combining Hidden Terminals in Wireless Networks," Proc. ACM SIGCOMM, 2008.

9. P. Gupta and P.R. Kumar, “The Capacity of Wireless Networks,” IEEE Trans. Information Theory, vol. 46, no. 2, pp. 388-404, Mar. 2000.

10. C. Hu, Y. Hong, and J. Hou, “On Mitigating the Broadcast Storm Problem with Directional Antennas,” Proc. IEEE Int’l Conf. Comm. (ICC), 2003.

11. S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Me´dard, “Resilient Network Coding in the Presence of Byzantine Adversaries,” Proc. IEEE INFOCOM, 2007.

12. S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, “XORs in the Air: Practical Wireless Network Coding,” Proc. ACM SIGCOMM, 2006.

13. L. Li, R. Ramjee, M. Buddhikot, and S. Miller, “Network Coding-Based Broadcast in Mobile Ad HocNetworks,” Proc. IEEE INFOCOM, 2007.

14. J. Liu, D. Goeckel, and D. Towsley, “Bounds on the Gain of Network Coding and Broadcasting in Wireless Networks,” Proc. IEEE INFOCOM, 2007.

15. X.Y. Li, Y. Wu, S.J. Tang, and X.H. Xu, “BroadcastCapacity for Wireless Ad Hoc Networks,” Proc. IEEE Int’lConf. Mobile Ad-Hoc and Sensor Systems (MASS), 2008.