ijournals International Journal of Software & Hardware Research in Engineering
ISSN No: 2347-4890 Volume 2 Issue 3, March 2014

An Evaluation Study of Multipath Routing Protocols in AHWMNs

C. Maddilety*, Vennapusa Prakash Reddy[2], P. Jammulaiah[3]

[1] Assistant Professor, SDITW Nandyal, A.P., India
[2] M.Tech Student, SSSISE Anantapuramu, A.P., India
[3]PG Research Scholar, JNTUACE Pulivendula, A.P., India

ABSTRACT

A Mobile Ad-hoc Network or MANET is a collection of mobile nodes sharing a wireless channel without any centralized control or established communication backbone. We can classify the routing protocol as flat and hierarchal routing and flat routing protocols are classifying into reactive (source initiated) and proactive (table driven). In this paper we mainly focusing on hybrid routing protocols combination of proactive and reactive i.e. Ad-hoc on demand Multipath Distance Vector (AOMDV) for better performance in simulation by comparing AntHocNet Routing Protocol, NS2 simulator is used for comparison and critical analysis of AOMDV and AntHocNet is done to find its merits and demerits.

Keywords: AOMDV, AntHocNet, MANET, NS2, simulation

1. INTRODUCTION

Mobile Ad-hoc networks (MANETs) [1] are networks in which all nodes are mobile and communicate exclusively via wireless connections. Usually, the nodes are equipped with a single, omni directional wireless antenna. There is no fixed infrastructure in the network, and there is no hierarchy: all nodes are in principle equal, and can function both as end points of data communication, and as routers, forwarding data for each other in multi-hop fashion. One can think of a group of users carrying wifi enabled devices such as mobile phones, PANs, laptops, etc., moving in a specific area and forming a dynamic wireless network among them.

As a consequence of the above mentioned properties, MANETs are dynamic, flat, fully decentralized networks without central control or overview. This gives rise to a number of tough challenges for networking algorithms. MANET algorithms should be highly adaptive to the ever changing environment. They should be robust in order to deal with unreliable wireless transmissions. They should work in a fully distributed way. Finally, they should be efficient in their use of the limited network resources, such as bandwidth, battery power in the mobile nodes, etc. when it comes to implementation, the MANET challenges have proven to be very hard to deal with, so that there is now also a growing interest in AHWMNs with less mobility and more hierarchy and organization, such as the wireless mesh networks.

2. AD-HOC WIRELESS MULTI-HOP NETWORKS

Ad-Hoc Wireless Multi-Hop Networks (AHWMNs) [2] are a collection of mobile devices which form a communication network with no pre-existing wiring or infrastructure. Different types of AHWMNs exist. Examples are mobile ad-hoc networks, wireless mesh networks and sensor networks. Mobile ad-hoc networks (MANET) are networks that are made up of homogenous devices. These devices communicate exclusively through wireless connections, normally using a single Omni directional antenna. All nodes in the network are equal, and there are no designated routers, meaning that all nodes can serve both as end points of data communication and as intermediate relay points or routers. Wireless mesh networks (WMNs) differ from MANETs mainly because they are more heterogeneous. They consist of mesh client nodes, which are similar to MANET nodes, and mesh router nodes, which are usually less mobile, have more resources (e.g. more powerful processors, more battery power, etc.), and support a variety of different wireless technologies. The availability of mesh routers allows the creation of a structured organization and can greatly improve the applicability and the capacities of the network. Sensor networks are AHWMNs that consist of wireless sensor nodes. Each sensor node is a small unit containing one or more sensors, a small processing unit and a wireless radio. Problems specific to sensor networks stem from the fact that sensor nodes are small and have very limited capacities that usually vast numbers of nodes are deployed, and that data traffic patterns show certain characteristic regularities.

[1]. TYPES OF AHWMNs:

In network research, work is being done on a number of different, related types of wireless networks, which can all be classified as AHWMNs. First, mobile ad-hoc networks are described [3], since these were the first

ijournals International Journal of Software & Hardware Research in Engineering
ISSN No: 2347-4890 Volume 2 Issue 3, March 2014

AHWMNs that received a lot of attention in the literature. Next, we present wireless mesh networks. These form a more general class of AHWMNs, of which mobile ad-hoc networks could in fact be considered a subclass. After that, sensor networks are described, which is an application specific subclass of AHWMNs.

[2]. ROUTING IN AHWMNs:

Routing in AHWMNs [4] are challenging since there is no central coordinator that manages routing decisions. In this, the focus is on the problem of routing in AHWMNs. Routing is the task of constructing and maintaining the paths that connect remote source and destination nodes of data. This task is particularly hard in AHWMNs due to issues that result from the particular characteristics of these networks. First important issue is the fact that AHWMNs are dynamic networks. This is due to their ad-hoc nature: connections between nodes in the network are set up in an unplanned manner, and are often changed while the network is in use. Especially when mobile nodes are used, such changes can take place continuously. An AHWMN routing algorithm should be adaptive in order to keep up with such dynamics. A second issue is the unreliability of wireless communication. Data and control packets can easily get lost during transmission, especially when mobile nodes are involved, and when multiple transmissions take place simultaneously and interfere with each other. A routing algorithm should be robust with respect to such losses. A third issue is caused by the often limited capabilities of the AHWMN nodes. There are limitations in terms of network bandwidth, node processing power, memory, battery power, etc... It is therefore important for a routing algorithm to work in an efficient way. Finally, last important issue is the network size. With the ever growing numbers of portable wireless devices, many AHWMNs are expected to grow to very large sizes. Routing algorithms should be scalable to keep up with such evolutions.

3. CLASSIFICATION OF AHWMN ROUTING PROTOCOLS

AHWMN routing protocols are classified [5] as topology-based, position-based and bio-inspired routing protocols.

- Topology-based
- Position-based routing algorithms
- Bio-inspired routing protocols

4. MANET PROTOCOLS

A. AntHocNet

AntHocNet [6] is a multipath routing algorithm that combines both proactive and reactive components. AntHocNet reactively finds a route to the destination on demand, and pro actively maintains and improves the existing routes or explore better paths. In AntHocNet, ant maintains a list of nodes it has visited to detect cycles. The source node sends out forward ants and when it receives all the backward ants, one generation is completed. Each node i keeps the identity of the forward ants, the path computation, number of hops, of the ant from the source to node i, and the time the ant visited node i. Note that more than one ant may have reached node i and therefore the identity of the ant is important. When an ant arrives at a node, the node checks the ant’s path computation and the time it reached node i. If the path computation and time are within a certain limit of those produced by another ant of the same generation then the ant is forwarded. Otherwise, the ant is discarded.

In case of a link failure at a node and no alternative paths are available, the node sends a reactive forward ant to repair the route locally and to determine an alternative path. If a backward ant is received for the reactive forward ant, the data packets are sent along the newly found path and all its neighbors are notified about the change in route. Otherwise, the node sends a notification to all its neighbors of the lost destination paths which in turn initiate forward ants from the neighbors.

1) General Overview of AntHocNet Routing Algorithm: AntHocNet is a hybrid algorithm, containing both reactive and proactive elements. The algorithm is reactive in the sense that it only gathers routing information about destinations that are involved in communication sessions. It is proactive in the sense that it tries to maintain and improve information about existing paths while the communication session is going on (unlike purely reactive algorithms, which do not search for routing information until the currently known routes are no longer valid). Routing information is stored in pheromone table. Forwarding of control and data packets is done in a stochastic way, using these tables. Link failures are dealt with using specific reactive mechanisms, such as local route repair and the use of warning messages. Below, we describe the general working of the AntHocNet routing algorithm.

At the start of a communication session, the source node of the session controls its pheromone table, to see whether it has any routing information available for the requested destination. If it does not, it starts a reactive route setup process, in which it sends an ant packet out over the network to find a route to the destination. Such an ant packet is called a reactive forward ant. Each intermediate node receiving a copy of the reactive forward ant forwards it. This is done via unicasting in case the node has routing information about the ant's destination in its pheromone table, and via broadcasting otherwise. Reactive forward ants store the full array of nodes that they have visited on their way to the destination. The first copy of the reactive forward ant to reach the

ijournals International Journal of Software & Hardware Research in Engineering
ISSN No: 2347-4890 Volume 2 Issue 3, March 2014

destination is converted into a reactive backward ant, while subsequent copies are destroyed. The reactive backward ant retraces the exact path that was followed by the forward ant back to the source. On its way, it collects quality information about each of the links of the path. At each intermediate node and at the source, it updates the routing tables based on this quality information. This way, a first route between source and destination is established at completion of the reactive route setup process. The full process is repeated later if the source node falls without valid routing information for the destination of the session while data still need to be sent.

Once the first route is constructed via the reactive route setup process, the algorithm starts the execution of the proactive route maintenance process, in which it tries to update, extend and improve the available routing information. This process runs for as long as the communication session is going on. It consists of two different sub processes: pheromone diffusion and proactive ant sampling. The aim of the pheromone diffusion sub process is to spread out pheromone information that was placed by the ants. Nodes periodically broadcast messages containing the best pheromone information they have available. Using information bootstrapping, neighboring nodes can then derive new pheromone for themselves and further forward it in their own periodic broadcasts. Pheromone diffusion process is similar to the dynamic programming approach used in distance vector routing. Such approaches are efficient to gather routing information, but can be slow to adapt to dynamic situations, possibly temporarily providing erroneous information. Therefore, the pheromone diffusion process can be considered as a cheap but potentially unreliable way of spreading pheromone information. Because of this potential unreliability, the pheromone that is obtained via pheromone diffusion is kept separate from the normal pheromone placed by the ants, and is called virtual pheromone; the pheromone placed by the ants will in what follows be called regular pheromone. The virtual pheromone is used to support the second sub process of proactive route maintenance, which is proactive ant sampling. In this sub process, all nodes that are the source of a communication session periodically send out proactive forward ants towards the destination of the session. These ants construct a path in a stochastic way, choosing a new next hop probabilistically at each intermediate node. Different from reactive forward ants, they are never broadcast. When calculating the probability of taking a next hop, proactive forward ants consider both regular and virtual pheromone. This way, they can leave the routes that were followed by previous ants, and follow the (potentially unreliable) routes that have emerged from pheromone diffusion. Once a proactive forward ant reaches the destination, it is converted into a proactive backward ant that travels back to the source and leaves pheromone along the way (regular, not virtual pheromone), just like reactive backward ants. This way, proactive ants can follow virtual pheromone and then, once they have experienced that it leads to the destination, convert it into regular pheromone. One could say that pheromone diffusion suggests new paths and that proactive ants check them out. The ant based full path sampling provides the reliability that is lacking in the efficient information bootstrapping process.

Data packet forwarding in AntHocNet is done similarly to other ACO routing algorithms: routing decisions are taken hop-by-hop, based on the locally available pheromone. Only regular pheromone is considered, as virtual pheromone is not considered reliable enough. Each forwarding decision is taken using a stochastic formula that gives preference to next hops that are associated with higher pheromone values. The formula is different from that used by the forward ants, so that data packets can follow a less exploratory strategy. Via parameter tuning, it is possible to vary between spreading the data packets over all possible available paths and deterministically sending them over the best path. While the former can in principle provide higher throughput through the use of multiple paths , the latter allows greedy exploitation of the learned information.

Link failures can be detected in AntHocNet via failed transmissions of data or control packets, or through the use of hello messages. Hello messages are short messages that are periodically sent out by all nodes in the network. The reception of a hello message is indicative of the presence of a wireless link, while the failure to receive such messages point to the absence of a link. In practice in AntHocNet, the function of hello messages is fulfilled by the same periodic messages that are used for pheromone diffusion. When a node detects a link failure, it controls its pheromone table, to see which routes become invalid due to the failure, and whether alternative routes are available for the affected destinations. Then, it broadcasts a link failure notification message to warn neighboring nodes about all relevant changes in its pheromone table. In case the link failure was associated with a failed data packet transmission, the node can also start a local route repair to restore the route to the destination of this data packet. To this end, it sends out a repair forward ant. Repair forward ants are similar to reactive forward ants, in the sense that they follow available pheromone information where possible, and are broadcast otherwise, but they have a limited maximum number of broadcasts, so that they cannot travel far from the old failed route. Upon arrival at the destination, the repair forward ant is converted into a repair backward ant that travels back to the node that started the repair process and sets up the pheromone for the repaired route. A last tool in dealing with link failures is the use of unicast warning messages. These are needed when data packets for a lost destination still arrive at the node after a link failure notification has already been broadcast. This can be due to bad reception of the broadcast notification message. In this case, the node unicasts a warning to the node it received the data from, in order to inform it that it can no longer forward data for this destination.

ijournals International Journal of Software & Hardware Research in Engineering
ISSN No: 2347-4890 Volume 2 Issue 3, March 2014

2) Detailed descriptions: In the following section we give a detailed description of the different components of the AntHocNet routing algorithm.

3) Data structures in AntHocNet: The different data structures that are maintained by each of the network nodes under AntHocNet. In particular, pheromone tables and neighbor tables.

B. AOMDV

Ad-hoc on demand Multipath Distance Vector (AOMDV), is an extension to the AODV routing protocol. AOMDV is designed to provide efficient recovery from route failures and efficient fault tolerance. To achieve these goals, AOMDV computes multiple loop-free and link-disjoint paths. A notion of advertised hop count is used to guarantee loop freedom, and a particular property of flooding is used to achieve Link-disjointness of multiple paths. The advertised hop count of a node for a destination represents the maximum hop count of multiple paths for the destination at the node.

When the AODV single path routing protocol is used, new route discovery is needed in response to every route break. This inefficiency can be avoided by having multiple paths for each destination. In this case, new route discovery is only needed when all paths are broken. The AOMDV protocol has two components: a rule to create and maintain multiple loop free paths, and a distributed protocol to find link-disjoint paths. The basic idea for finding link-disjoint paths is as follows. To consider the paths between a pair of nodes as disjoint paths, it is necessary that all but the first and last hops of those paths are distinct. AOMDV augments the AODV route discovery procedure in two ways:. By exploiting the routing information obtained via duplicate route request copies, alternate loop-free reverse paths are formed at the intermediate and the destination nodes. 2. The destination node generates multiple route replies that travel along multiple loop-free reverse paths to the source established during the route request phase to get multiple loop-free forward paths to the destination. As in AODV, AOMDV uses destination sequence numbers to ensure loop-freedom. Every node maintains one or more paths to a destination corresponding to the highest sequence number for that destination. Route maintenance in AOMDV is similar to that in AODV. The difference is that, in AOMDV, a node only generates or forwards a RERR packet for a destination when all paths to the destination break.

5. SIMULATION

The simulator used here is Network Simulator [8] [9], NS, version 2.34. NS is a discrete event simulator targeted at networking research. NS provides substantial support for simulation of UDP, routing, and multicast protocols over wired and wireless (local and satellite) networks. NS is free software, publicly available under the GNU GPLv2 license for research, development, and use.

A. Performance metrics

1) End-to-End delay: It is time delay for data packet from source to destination.

2) Packet delivery ratio: The number of data packets successfully received by the destination.

3) Routing overhead: The number data packets transmitted by the source node.

Table 1. Simulation Parameters

PARAMETER TYPE	VALUE
Simulator	NS 2.34
Number of nodes	10
Simulation time	100 sec
Node speed	10,20,30,40,50,60,70,80,90,100 m/sec
Data rate	10, 20, 30, 40 Mbps
Simulation area	1000m * 1000m
Data type	CBR
Pause time	0-100 sec

B. Simulation Results

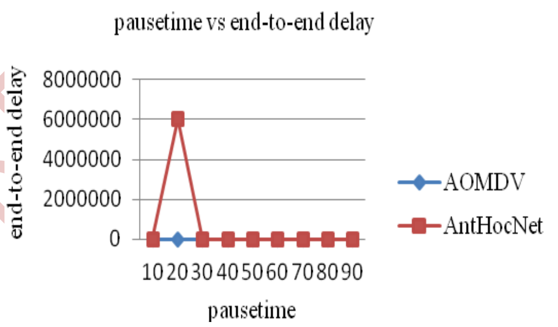

Fig 1: End to end delay with varying pause times

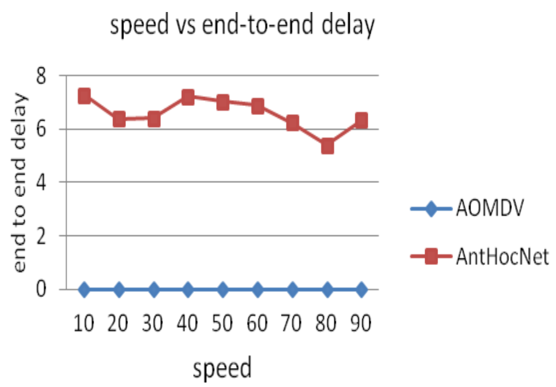

Fig 2: End to end delay with varying speed

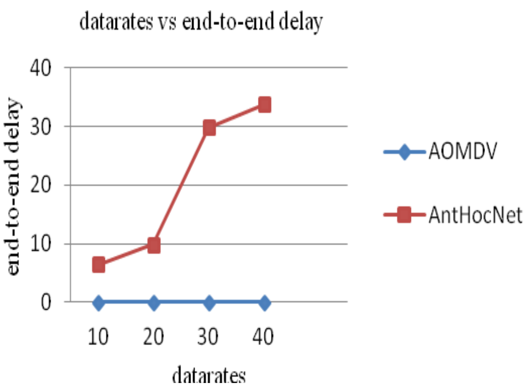

Fig 3: End to end delay with varying datarates

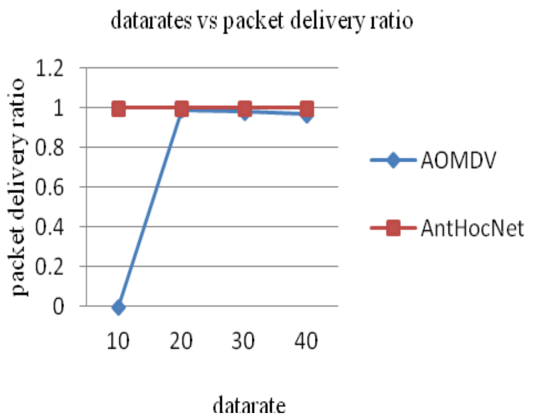

Fig 6: Packet delivery ratio with varying datarates

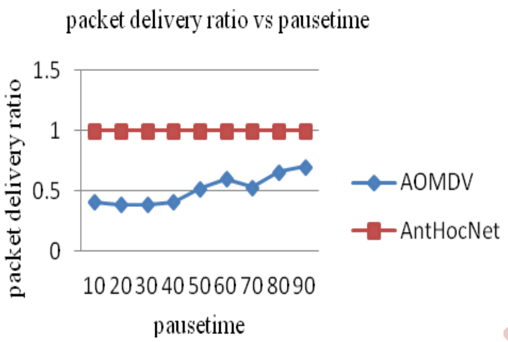

Fig 4: Packet delivery ratio with varying pause times

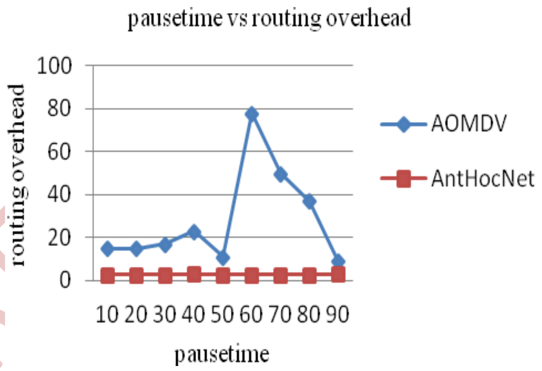

Fig 7: Routing overhead with varying pause time

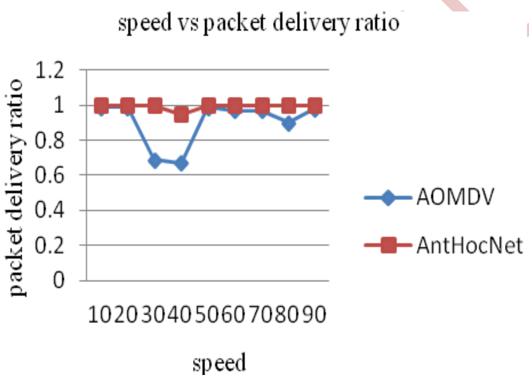

Fig 5: Packet delivery ratio with varying speed

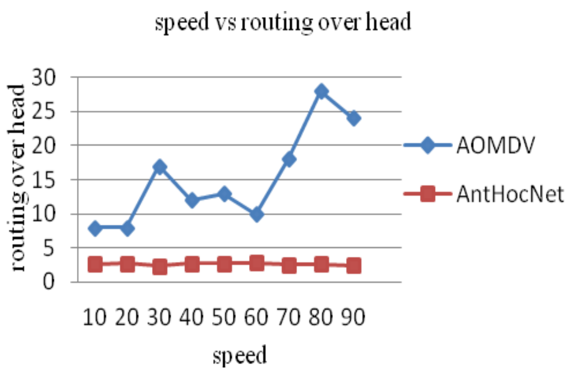

Fig 8: Routing overhead with varying speed

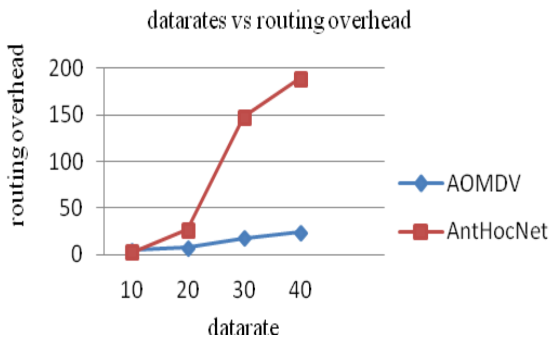

Fig 9: Routing overhead with varying datarates

6. CONCLUSION

The performance of two popular hybrid routing protocols, AOMDV, and AntHocNet is evaluated by parameters like end-to-end delay, packet delivery ratio, routing overhead. From the results it can be concluded that AOMDV performs very well with respect to routing overhead by varying pause time and speed and it is due to this factor that end-to-end delays are low in AOMDV. In end to end delay, packet delivery ratio is more in AntHocNet by varying pause time, speed and datarate. In routing overhead is also more with varying datarates.

REFERENCES

[1] Mobile Ad-hoc network (MANET), www.ietf.org/html.charter/manetcharter.

[2] N. Boulicault, G. Chelius, and E. Fleury. "Experiments of ana4: An im-plementation of a 2.5 framework for deploying real multi-hop ad-hoc and mesh networks." In Proceedings of the IEEE ICPS Workshop on Multi-hop Ad-hoc Networks: from theory to reality, Santorini, Greece, July 2005.

[3] SujataAgrawal, Dr. M.B. Daigavane, Dr.K.D.Kulat, "Performance Evaluation Of Routing Protocols For Wireless Ad-hoc Network", International Journal of Advanced Engineering Technology, IJAET/Vol.III/ Issue I/January-March, 2012/13-17.

[4] TahirSaleem, KifayatUllah, "Performance Issues of Routing Protocols in Mobile Ad-hoc Networks", Master of Science in Electrical Engineering, Thesis no: MEE 10:95 November, 2010.

[5] Carlos de Morais Cordeiro and Dharma P. Agrawal, "Mobile Ad-hoc networking".

[6] Gianni Di Caro, Frederick Ducatelle and Luca Maria Gambardella, "AntHocNet: An Adaptive Nature-Inspired Algorithm for Routing in Mobile Ad-hoc Networks".

[7] Mahesh K. Marina, and Samir R. Das, "Ad-hoc on-demand multipath distance vector routing (AOMDV)", Published online in Wiley InterScience, Wirel. Commun. Mob. Comput.2006; 6:969–988, DOI:10.1002/wcm.432.

[8] NS-2 Network simulator http://www.isi.edu/nsnam/ ns.

[9] The VINT Project, "The ns Manual (formerly ns notes and documentation)",http://www.isi.edu/nsnam/ns /ns- documentation.html, Nov, 2011.