Performance Comparisons of AOMDV and OLSR Routing Protocols for Mobile Ad Hoc Network

May Zin Oo
Faculty of Computer Science & Information Technology
University of Malaya
50603 Kuala Lumpur, Malaysia
mayzinoo.mz@gmail.com

Mazliza Othman
Faculty of Computer Science & Information Technology
University of Malaya
50603 Kuala Lumpur, Malaysia
mazliza@um.edu.my

***Abstract*— A Mobile Ad Hoc Network (MANET) eliminates the complexity of an infrastructured configuration and allows wireless devices to communicate with each other on the fly for any application. It does not rely on a base station to coordinate the flow of messages to nodes in the network. A primary challenge is for each device to maintain the information to properly route traffic and data packets. In this paper, we analyze the performance differences of Ad hoc On-demand Multi-path Distance Vector (AOMDV) and Optimized Link State Routing (OLSR) routing protocols. We evaluate their performance through simulation using network simulator (ns-2). We analyse the strengths and weaknesses of these two protocols by measuring packet loss rates (%), average end to end delay (sec), and normalized routing load.**

Keywords: AOMDV, OLSR, Packet loss rate, End-to-end delay, Normalized routing load

I. Introduction

In mobile ad hoc networks, communication between two end nodes is carried out through a number of intermediate nodes whose function is to relay information from one point to another. Routing packets between any pair of nodes becomes a challenging task because the nodes can move randomly within the network. A route that is believed to be optimal at a given point in time might not work at all a few moments later.

Traditional routing protocols [2, 3, 6, 9] are proactive and they maintain routes to all nodes. They react to any change in the topology even if no traffic is affected by the change, and they require periodic control messages to maintain routes to every node in the network. As mobility increases, more of scarce resources, such as bandwidth and power will be used. Alternative reactive routing protocols [4, 5, 7, 10] determine the route when they explicitly need to route packets, thus avoiding nodes from updating every possible route in the network. The behavior of routing protocols depends on the network size, link capacity, and node mobility.

The most popular on-demand routing protocol, Ad-hoc On-demand Multipath Distance Vector (AOMDV) routing protocol [7] is an enhancement of Ad-hoc On-demand Routing Protocol (AODV) [4]. AOMDV allows multiple paths between a source and a destination to provide efficient fault tolerance by providing faster and more efficient recovery from route failures in a dynamic network. By computing multiple paths in a single route discovery attempt, new route discovery is needed only when all paths fail. This reduces not only the route discovery latency but also the routing overheads.

Among proactive routing protocols, Optimized Link State Routing (OLSR) [9] is an optimization of pure link state routing protocol and inherits the stability of a link state algorithm and takes over the advantage of proactive routing nature to provide the routes immediately when needed. OLSR minimizes the flooding of the control messages by allowing only selected nodes to relay.

The remainder of this paper is organized as follows. Section II discusses AOMDV and OLSR routing protocols. Section III describes the simulation parameters to analyze the performance differences. Section IV discusses the simulation results of the two routing protocols with different backgrounds i.e. the variations of node mobility and congestion window size. Finally, we summarize and conclude our paper in section V.

II. Overview of AOMDV and OLSR Routing Protocols

A. AOMDV Routing Protocol

AOMDV was designed primarily for highly dynamic ad hoc networks where link failures and route breaks occur frequently. It maintains routes for destinations in active communication and uses sequence numbers to determine the freshness of routing information to prevent routing loops. It is a timer-based protocol and provides a way for mobile nodes to respond to link breaks and topology changes.

1. Types of control messages

It uses three types of control message: route request (RREQ), route reply (RREP) and route error (RRER) to discover routes, that are sent to port 654 using User Datagram Protocol (UDP).

If a node needs a route to transfer data packets, it broadcasts RREQ throughout the network. When a node receives a RREQ packet, it checks the destination address field of RREQ. If it has information about a destination or it is destination itself, it uses RREP packet to unicast to a

978-0-7695-3982-9/10 $26.00 © 2010 IEEE
DOI 10.1109/ICCEA.2010.33

source node. If a node does not have a route to destination and its multiple alternative paths are not available, it sends RRER message back to the upstream node.

2. Format of routing table

AOMDV uses the *advertised hop count* field for multiple route entries. To define multiple next hops with relevant hop counts, it uses *next hop* lists in the routing table. A node updates its advertised hop count for a destination whenever it sends a route advertisement for the destination [7].

Destination
SN
Advertised hop count
Next hop {(next hop$_1$, hop count$_1$), (next hop$_2$, hop count$_2$), ...}
Expiration time out

Figure 1: Routing table of AOMDV

3. Route discovery procedure

When a source node *A* has data packets for a destination, it first checks its routing table to ascertain whether it already has a route to the destination node *B*. If a route is available, it sends the data packets by utilizing its existing route. If it is not available, it initiates a route discovery procedure by broadcasting RREQ to obtain a route to the intended destination *B*. AOMDV computes multiple paths and observes each route advertisement to define an alternate path to the source or the destination during a route discovery procedure. RREQ packets arriving at the nodes are copied and sent back to the source nodes. This approach may push the formation of loops due to accepting all copied routes. In order to eliminate any possibility of loops, it uses *advertised hop count* field in the route tables. The advertised hop count of a node S for a destination D is set the maximum hop count of the multiple paths for D at S. The *advertised hop count* is initialized each time the sequence number is updated. By doing so, AOMDV only accepts alternative routes with lower hop counts. Each RREQ conveys an additional *first hop* field to indicate the first neighbor of the source node. The intermediate nodes do not discard duplicate copies of RREQ immediately as long as each RREQ provides a new node-disjoint path to the source. If an intermediate offers a new path, a reverse path is set up. It sends back a RREP to the source. At the destination, reverse routes are established like in the same situation of intermediate nodes.

4. The Sequence Number Procedure of AODV and AOMDV

To prevent the formation of routing loops and to select the most recent routing path to the destination, a sequence numbering technique is exploited. AOMDV sets its destination SN as follow.

if $(seqnum^d_i < seqnum^d_j)$ *or* $(((seqnum^d_i = seqnum^d_j)$ *and* $(hopcount^d_i > hopcount^d_j))$
then
$seqnum^d_i = seqnum^d_j$;
$hopcount^d_i = hopcount^d_j + 1$
$nexthop = j$;
end if

Each RREQ conveys an additional *first hop* field to indicate the first neighbor of the source node. The intermediate nodes do not discard duplicate copies of RREQ immediately as long as each RREQ provides a new node-disjoint path to the source. If an intermediate offers a new path, a reverse path is set up. It sends back a Route REPly (RREP) to the source. At the destination, reverse routes are established like in the situation of intermediate nodes. If a link break occurs between a source and destination, a node that knows this link break needs to send Route ERRor (RERR) message back to the source node. For periodic route updates, HELLO messages are broadcast in a timely manner.

B. OLSR Routing Protocol

OLSR operates as a proactive routing protocol to exchange routing information with other nodes in the network. The pure link state protocols are optimized to reduce the number of control packets and their sizes that are required for data transmission.

1. Types of control messages

There are three types of control messages: HELLO, Topology Control (TC), and Multiple Interface Declaration (MID).

HELLO is used to find the information about the link status and neighbors' information.

A node receives the topology information by flooding a TC message periodically using the multipoint relying mechanism.

To announce that a node is running more than one interface, a MID message is flooded throughout the network.

2. Format of routing table

Destination address
Next address
Number of hops
Local interface address

Figure 2: Routing table of OLSR

Due to the proactive routing protocol, the routing table must have routes for all available hosts in the network.

3. Route discovery procedure

OLSR is a carefully designed protocol to work in a distributed manner and does not depend on any central entity. Each node chooses its neighbor nodes as multipoint relays (MPR), which are responsible for forwarding control traffics by flooding. MPRs provide a shortest path to a destination by declaring and exchanging the link information periodically for their MPR's selectors. By doing so, the nodes maintain the network topology information.

The Hello messages are broadcast periodically for link sensing, neighbor's detection and MPR selection process. The information contained in the HELLO message:

- how often the host sends Hello messages,
- willingness of a host to act as a Multipoint Relay, and
- information about its neighbor (i.e. interface address, link type and neighbor type)

The link type indicates that the link is symmetric, asymmetric or simply lost. The neighbor type is just symmetric, MPR or not a neighbor. If the link to the neighbor is symmetric, this node is chosen as MPR.

After receiving a HELLO message information, a node builds a routing table. When a node receives a duplicate packet with the same sequence number, it discards the duplicate. A node updates its routing table either when a change in the neighbor is detected or a route to any destination has expired and a shorter route is detected for a destination.

4. Multipoint Relays

MPR is used to reduce the number of nodes that broadcast the routing information throughout the network. To forward data traffic, a node selects its one hop symmetric neighbors, referred to as MPRset that covers all nodes that are two hops away. The MPRset is calculated from information about the node's symmetric one hop and two hop neighbors. This information in turn is extracted from HELLO messages. Similar to the MPRset, a MPR Selectors set is maintained at each node. A MPR Selector set is the set of neighbors that have chosen the node as a MPR. Upon receiving a packet, a node checks its MPR Selector set to see if the sender has chosen the node as a MPR. If so, the packet is forwarded, otherwise the packet is processed and discarded.

Although the proactive nature of the routing protocols provides all routing information to all nodes in the network, they need to send periodic control message throughout the entire network, leading to the power, bandwidth and memory usages.

III. SIMULATION PARAMTERS

We use network simulator ns2 [12] to analyze AOMDV and OLSR routing protocols and measure the packet loss rate (%), average end-to-end (EtE) delay and normalized routing load (NRL) as performance metrics under two scenarios: node mobility variation and Transmission Control Protocol (TCP) congestion window size variation.

We use a movement pattern of the random waypoint mobility model [10], which is generated by BonnMotion version 1.4 [1]. The pause time of nodes is set to zero for continuous movement. The 100 nodes are randomly moving in a 2000 x 750 simulation area. The file transfer application of TCP is used as a traffic source and 60 TCP connections are built during 500 second simulation time. We use an enhanced version of TCP, called TCP-Reno [8], that enables fast retransmit and fast recovery functions.

IV. SIMULATION RESULTS

A. Performance metrics

We evaluate three essential performance metrics to analyze the performance differences of AOMDV and OLSR:

1. Packet loss rates (%) — the number of packets loss at an application layer while transferring data packets i.e.
$$Packet\ loss\ rate = \frac{dropped\ packets}{Highest\ packet\ ID + 1} * 100$$
2. Average end to end delay — an end-to-end transmission delay of data packets that are delivered to the intended destination successfully.
3. Normalized routing load — the number of routing packets transmitted per data packet delivered at the destination. Also each forwarded packet is counted as one transmission. This metric is also highly correlated with the number of route changes that occurs in the simulation.

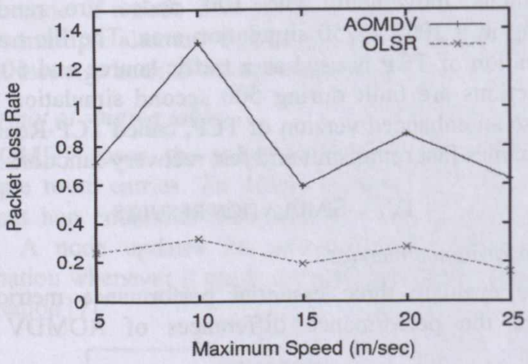
B. Varying node mobility

Firstly, we analyze the performance differences of AOMDV and OLSR under node mobility variations. We measure packet loss rate, average end-to-end delay and normalized routing load and discuss AOMDV and OLSR routing protocols that have different core routing mechanisms.

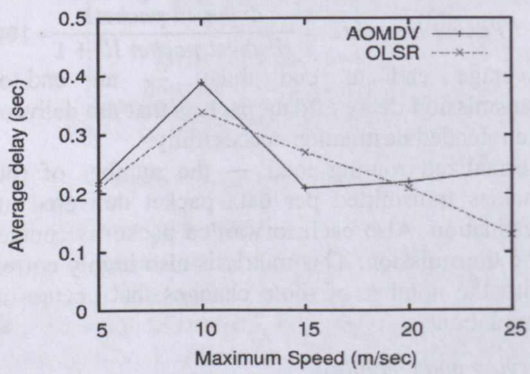
On average, AOMDV encounters a greater packet loss rate than OLSR by a factor of up to 3.4 as shown in Fig 3(a). Although AOMDV supports multiple paths between a source and destination, it is difficult to recover the packets during the time between the failure of a primary route and the finding of an alternative route. On the other hand, as OLSR nodes always have routes in hand due to its proactive nature, it reduces packet loss rates significanlty.

Moreover, Fig 3(b) shows that AOMDV incurs an unstable end to end delay with fluctuations. When node speed increases, the average end to end delay of OLSR decreases and becomes more stable. OLSR takes an advantage of link state routing and performs better due to well updated routes. On the other hand, during the recovery of route failures, TCP at the receiver side is unable to receive the packets. After waiting for a timeout period, TCP halves its congestion window and starts the slow start procedure again [8]. By doing so, routing protocols may incur a tremendous delay at high speed.

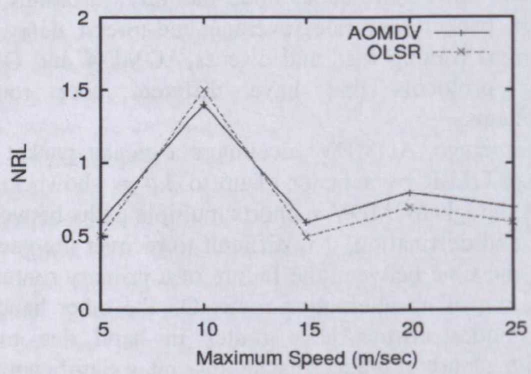
OLSR reduce the routing load selecting the MPR to propagate the updates of the link state. AOMDV also takes the advantage of maintaining multiple alternative paths in the routing tables of nodes.



(a) Packet loss rate over node speed



(b) Average end to end delay over node speed

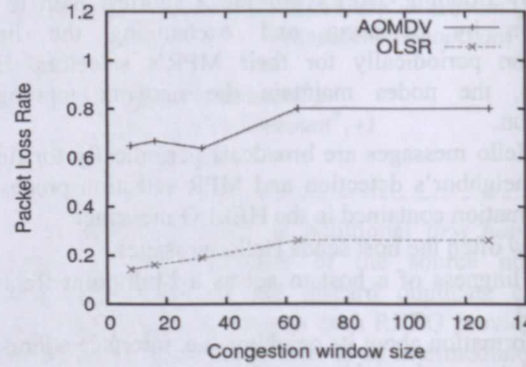


(c) Normalized routing load over node speed

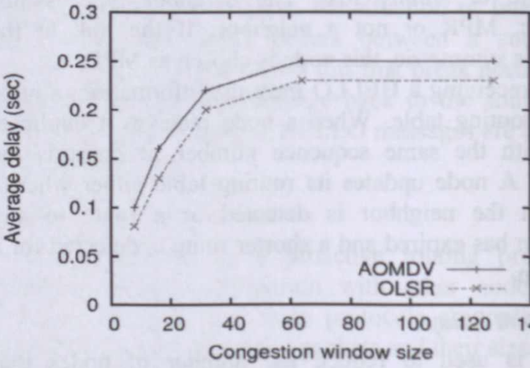
Figure 3: Performance comparions of routing protocols over node speed variations

Only when all alternative nodes are unavailable, a new route discovery procedure is initialized. Although the core mechanisms of both protocols are totally different, the routing load of both protocols are not significantly different as shown in Fig 3(c).

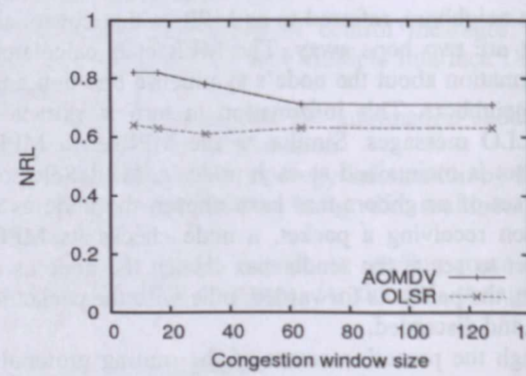
When we increase the node speed from 5 ms^{-1} to 10ms^{-1}, the packet loss rate of both protocols increase suddenly and the fluculation of AOMDV is very high compared to OLSR. At this point, the average end to end delay and routing load increases accordingly.



(a) Packet loss rate over congesiton window size



(b) Average end to end delay over congestion window size



(c) Normalized routing load over node speed

Figure 4: Performance comparison of routing protocols over congestion window size variations

C. *Varying TCP Congestion Window Size*

Finally, we examine the performance differences of AOMDV and OLSR under congestion window size variations.

Like in the mobility variations, Fig 4(a) shows that AOMDV encounters a greater average packet loss rate than OLSR by a factor of up to 3.5. In both protocols, as the congestion window size increases, the packet loss rate increases slightly. When the TCP window size is too big, it may have difficulties trying to recover from lost data. For example, when node A sees a window size of 128kB advertised by node B, node A eventually will try to send that exact amount of data. If the bandwidth for that connection only enables 64kB of in-flight data, the remaining data packets have to be queued. It is very difficult to recover the packets as well as the packet losses and queue overflows occur as the node mobility increases.

As a consequnce, the average end to end delay increases in both routing protocols as the window size increases. AOMDV suffers a somewhat higher delay than OLSR.

OLSR does not need to do the extra work for the route discovery but it consumes the usage of bandwith compared with AOMDV. As shown in Fig 4(c), OLSR has a lower routing load than AOMDV at all congestion window size variations. OLSR still has an advantage that the reactivity of the detection of topology changes can be improved by shortening the time interval of periodic control messages.

V. SUMMARY AND CONCULSION

We examine the performance differences of AOMDV and OLSR utilizing TCP-Reno as a traffic source. We measure the packet loss rate, delay and routing load as performance metrics.

The proactive routing protocols receive a higher route efficiency in the scattered traffic network because the route table updates come from periodic updates and no additional overhead occurs for finding a new route. However, the proactive routing protocols cannot be used in scarce resource networks.

Our simulation results show that OLSR outperforms AOMDV by a factor of up to 3.5 in the variations of node speed and TCP congestion window size.

Both protocols scalability is restricted due to their proactive or reactive natures. AOMDV meets the flooding overhead in the high mobility network and OLSR increments the size of the routing table and topological update messages, leading to the usage of more scarce resources.

REFERENCES

[1] BonnMotion v.3a, http://web.informatik.unibonn.de/IV/BonnMotion.

[2] C.E. Perkins, and T.J. Watson, "Highly dynamic destination sequenced distance vector routing (DSDV) for mobile computers," Proc. ACM Communications, Architectures, Protocols and Applications, vol. 24, pp. 234-244, 1994.

[3] C.-C. Chiang, H.-K Wu, W. Liu, and M. Gerla, "Routing in clustered multihop mobile wireless networks with fading channel," Proc. IEEE SICON'97, pp. 197-221, 1997.

[4] C. E. Perkins, E. M. Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561, (2003) IETF.

[5] D. Johnson, Y. Hu, and D. Maltz, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4," RFC 4728, 2007.

[6] G. Jayakumar, and G. Gopinath, "Performance Comparison of MANET Protocols Based on Manhattan Grid Mobility Model," Journal of Mobile Communication, vol. 2 pp. 18-26, 2008.

[7] Marina, M.K. and Das, S.R., "On-demand Multipath Distance Vector Routing for Ad Hoc Networks, "Proc. of 9th IEEE Int. Conf. on Network Protocols," pp. 14-23, 2001.

[8] M. Allman, V. Paxson, and W. Stevens, "RFC 2581-TCP Congestion Control," Internet Society, 1999.

[9] T. Clausen and P. Jacquet "Optimized Link State Routing Protocol (OLSR)." RFC 3626, IETF Network Working Group, October 2003.

[10] T. Camp, J. Boleng, and V. Davies, A survey of mobility models for ad hoc network research, Wireless Communication and Mobile Computing Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications. 2 (2002) 483-502.

[11] V.D. Park and M.S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," Proc. sixteenth Ann. Joint conf. of the IEEE computer and Communication Societies. vol. 3 pp. 1405-1413, 1997.

[12] VINT Group, "Network Simulator Ns-2, source codes available: http://www.isi.edu/nsnam/ns.