Improving AODV Protocol to Avoid Congested Areas in Mobile Ad hoc Networks

Cuong Do Dinh¹*, Tam Nguyen Van² and Hieu Nguyen Gia²

¹University of Information and Communication Technology, Thai Nguyen, Vietnam; ddcuong@ictu.edu.vn
²Information Technology Institute, Ha Noi, Vietnam; nvtam@ioit.ac.vn, nghieu@ioit.ac.vn

Abstract

Objectives: To improve routing mechanism of AODV protocol to enhance performance of Mobile Ad hoc network getting congested areas. Methods/Statistical Analysis: Proposing a method to estimate the link delay at MAC layer based on service time. This method includes estimations of frame error rate and channel utilization; Using cross-layer approach to get value of link delay estimated at MAC layer. This delay is utilized to calculate delay of route at Network layer; Using route delay as the routing metric in routing process. Findings: The simulation result of AODV and the proposed protocol AODV-DM in NS2 simulation shows that AODV-DM protocol archives better performance than AODV protocol in terms of 30% higher of packet delivery ratio, 27% higher of average throughput, 32% lower of average packet delay and 40% lower of routing overhead. Application/Improvements: The proposed AODV-DM protocol can be used in low performance Mobile Ad hoc Network due to congestion.

Keywords: AODV, Congestion, Delay, Mobile Ad hoc Networks, Routing, Service Time

1. Introduction

Ad hoc on Demand Distance Vector (AODV)¹ is a typical on-demand routing protocol for ad hoc networks with table driven mechanism, using hop count as a routing metric in distance vector routes. There have been a number of enhancements proposed to improve performance of AODV. These enhancements can be divided into two groups: enhancements made at Network layer²-⁶ and enhancements using cross-layer approach⁷-¹²

Regards to the group having enhancements of AODV made at Network layer, the propose in’ utilizes a method to update information about nodes which RREQ packets go through and intermediate nodes which will transmit this information in order to reduce the number of broadcasting RREQ packet when network topology changes and to speed up the process of finding a new path when the current path is down. The idea of removing paths whose time out value exceed the dynamic threshold is determined according to the route hop count combined with energy transmission control technique based on the source energy and load balancing proposed in³. To reduce the rate of packet lost when the current route is down, Bypass-AODV protocol⁴ is a modification of the AODV protocol to set up and use the bypass routes between the upstream and downstream nodes of the broken link. AODV-BR protocol⁵ inherited the whole operation mechanism of AODV protocol but the intermediate nodes on the path from the source to the destination is capable of looking more redundant routes towards the destination and inserting them into the backup routing table to use when the main road is down. This protocol does not need
to use new control packets but utilizes broadcasting ability of wireless nodes to get information about backup route from RREQ packages whose destination address is not receiver’s address. There are three suggestions for the improvement of AODV given in\(^4\), which includes AODV-QL protocol with flooding reduction technique of RREQ packet, AODV-LR protocol with route local repair technique and AODV-ERS protocol with search technique expanding circularly from the source node.

Regards to AODV enhancements based on cross-layer approach, each node in LEA_AODV\(^7\) protocol obtains information about source’s energy from MAC layer and establish a RREQ broadcasting threshold. If the energy of the source is smaller than the threshold, an intermediate node will not broadcast the received packet. Another enhancement of AODV, which is also related to source’s energy based routing approach, is PC-AODV\(^8\). This routing method uses an algorithm to estimate required energy level for each route in the routing table and then choose the route which has smallest required energy requirement. Furthermore, it also utilizes different energy management and control policies for data packets and control packets. RSEA-AODV\(^9\) protocol is also one of enhancement of AODV. This approach avoids choosing routes which have intermediate nodes with low energy level in order to increase the sustainability of the chosen route and trigger mechanism for a new route if MAC layer recognizes the indicators of link break, congestion, mobility and source’s energy threshold excess. This guarantees the availability of nodes when they forward data. To forecast when a link is broken\(^10\), proposed the 3rd Newton’s Difference Equation whose input parameters are signal strength and time data packets received in the past. When the current time is greater than the links break predicted time, a local repair technique will be used to find redundancy paths to replace when the main road is down.

However, these above enhancements have not proposed any change for routing metric in AODV based on cross-layer approach to achieve delay of links estimated using service time definition as a routing metric. Because AODV uses routing metric to be hop count and the route is chosen to be inserted into the routing table is the route having smallest hop count, therefore, if every node choose shortest route based on hop count, every traffic flows seem to go through the center of the network. As a result, congestion at central nodes of the network might happen and reduce the performance of the network. The question is raised here is to find out an appropriate routing metric for AODV routing protocol in order to avoid routing through congestion areas, while ensuring loop routing resistance by routing techniques using hop count and the sequence number of the AODV protocol.

This paper utilizes link delay estimation technique based on the principle of Distributed Coordination Function (DCF) at the MAC layer with the network multiple access method CSMA/CA. The delay parameters used to estimate the service time delay of links at the MAC layer include the back-off time, the transmission time and the deferring time. The delay value is then transmitted back to the network layer for use in calculating end-to-end delay of the route found. Techniques, which substitutes hop count by delay, are also recommended to ensure that the proposed protocol (AODV-DM) uses a new metric while still finding out a free-loop route using routing technique based on hop count of AODV.

The rest of this paper is organized as follows: The basis for estimating the value of the link delay is presented in Section 2. Section 3 will present all of the proposed improvements to the AODV in the introduced protocol named AODV-DM, which uses cross-layer approach based on the use of link service time as a routing metric. The performance evaluation of the two protocols based on NS2 simulation results are given in section 4 and finally, Section 5 will give the conclusion of this paper.

2. Link Delay Estimation

For networks using IEEE 802.11 technology at the Data Link layer, the average delay of successful sending a frame through a link is an important parameter in determining the data transfer capability of that link for data flows received from the Network layer. It is obvious that the average latency of links in congested areas of the network will be greater than that of links in other regions. Therefore, the average delay to successful transmit a packet through an end-to-end route will be increased if this route has links which are located in congested areas. If AODV routing protocol is improved to be able to choose a loop-free route with the smallest delay, this protocol will
have ability to prevent the routing of data through links in congested areas.

The average delay for successful transmitting a frame through a link suggested in this paper is the service time of source node of that links to send a frame through that link. When a node whose MAC layer operates under DCF mode based on the protocol for carrier transmission access CSMA/CA, service time is defined as the time to successfully transmit a packet or canceled a packet when the number of retransmissions exceeds the threshold allowed in the MAC layer. This value includes the deferring time when the channel is busy, back-off time of time to randomly re-access channel and transmission time to completely transmit a frame. Suppose $T_s$, $T_d$, $T_b$, and $T_t$ are service time, deferring time, back-off time and transmission time of a frame transmission process via a link respectively, the value of $T_s$ is determined as follows:

$$T_s = T_d + T_b + T_t$$

(1)

Assume that $R_n$ and $P_s$ are the number of retransmission and probability of successful transmission a frame through a link, respectively. The relationship between them is then determined by the following formula;

$$R_n = \frac{1}{P_s}$$

(2)

According to the DCF model based on CSMA/CA, a frame transmission time $T_t$ through a link is determined by the following formula;

$$T_t = R_n \left( SIFS + \frac{PH_s + MH_s}{B_r} + \frac{FD_s}{D_r} \right) + \frac{ACK_s}{B_r} + DIFS$$

(3)

From (2) and (3), the transmission time can be expressed as;

$$T_t = \frac{1}{P_s} \left( SIFS + \frac{PH_s + MH_s}{B_r} + \frac{FD_s}{D_r} \right) + \frac{ACK_s}{B_r} + DIFS$$

(4)

where $SIFS$ and $DIFS$ are Short Inter-Frame Space and DCF Inter-Frame Space respectively; $PH_s$, $MH_s$ and $FD_s$ are header size of the Physical layer, header size of the MAC layer and payload of the frame, respectively; $B_r$ and $D_r$ are Basic Rate and Data Rate respectively; $ACK_s$ is size of ACK packet.

According to\(^1\), formula (3) can be estimated as follows;

$$T_t = \frac{FD_s}{BW_{e}P_s}$$

(5)

where $BW_e$ is Efficient Bandwidth. This value is measured for each working speed of IEEE technology at the MAC layer\(^1\) and is summarized in Table 1.

**Table 1. Efficient Bandwidth**

| IEEE 802.11 Operating Rate (Mbps) | with RTS/CTS (Mbps) | without RTS/CTS (Mbps) |
|-----------------------------------|---------------------|------------------------|
| 1.0                               | 0.89                | 0.94                   |
| 2.0                               | 1.64                | 1.8                    |
| 5.5                               | 3.52                | 4.34                   |
| 11.0                              | 5.17                | 7.15                   |

When a node is in the back-off state, each channel is idle in a time slot, value of back-off counter is decreased by one unit. This value is randomly set in a range between zero and the Contention Window value minus one of each back-off stage. When a node completed sending a frame but has not received ACK after a SIFS period, it assumes that the frame transmission is false and resend that frame in a new back-off stage with a double Content Window value. Therefore, back-off time is total time of all frame re-transmissions based on successful transmission rate, average value of Content Window of each back-off stage and Slot Time value of IEEE 802.11 technology.

Assume that $CW_{\min}$ is original value of Contention Window, at $i$\(^{th}\) back-off stage, value of Contention Window ($CW_i$) is calculated based on $CW_{\min}$ as follows;

$$CW_i = 2^i CW_{\min}$$

(6)
At $i^{th}$ back-off stage, due to value of back-off counter is set according to random uniform distribution in $[0, CW_i - 1]$ so this value is estimated by mean value ($CW_i/2$). Therefore, reduction time for back-off counter to become zero would be ($T_{\text{slot}} CW_i/2$) with $T_{\text{slot}}$ is Slot Time.

The probability that the first back-off stage occurs is $P_s$. One frame is retransmitted $i^{th}$ times means that sending node is in $i^{th}$ back-off stage. Therefore, probability of $i^{th}$ back-off stage happened is ($P_s (1 - P_s)^i$).

Considering each occurrence of a back-off stage is an event. Probability of happening the following event depends on the probability of happening the previous event. Assume that $n$ is the number of back-off stage, according to\textsuperscript{11}, the value of back-off time $T_b$ is calculated as follows;

$$T_b = \frac{\sum_{i=0}^{n-1} T_{\text{slot}} (P_s (1 - P_s)^i \sum_{j=0}^{i} \frac{CW_j}{2})}{\sum_{i=0}^{n-1} P_s (1 - P_s)^i}$$  \hfill (7)

Assume that $F_e = 1 - P_s$ is frame error rate, the equivalent transformation of the formula (7) is described as follows;

$$T_b = T_{\text{slot}} \frac{CW_{\text{min}}}{2} \left( \frac{2P_s (1 - (2F_e)^n)}{(2P_e - 1)(1 - F_e^n)} - 1 \right)$$  \hfill (8)

CNU is called as channel utilization of a source node belonging to a link in a unit of time, so proportion of the channel idle is $(1 - \text{CNU})$. Deferring time of a node is calculated when the channel is busy, transmission time and back-off time of a node are calculated when the channel is free. Therefore, the ratio of the waiting time on the total transmission time and back-off time is the ratio between CNU and $(1 - \text{CNU})$. Consequently, deferring time is estimated using formula (9);

$$\frac{T_d}{T_b + T_t} = \frac{\text{CNU}}{1 - \text{CNU}}$$  \hfill (9)

Therefore:

$$T_d = \frac{\text{CNU}}{1 - \text{CNU}} (T_b + T_t)$$  \hfill (10)

From formula (1) and (10), we got;

$$T_s = \frac{\text{CNU}}{1 - \text{CNU}} (T_b + T_t) + T_b + T_t = \frac{T_b + T_t}{1 - \text{CNU}}$$  \hfill (11)

From formula (5), (8) and (11), we got a formula (12) to estimate value of the service time $T_s$.

$$T_s = \left( \frac{CW_{\text{min}}}{2} T_{\text{slot}} \left( \frac{2P_s (1 - (2F_e)^n)}{(2P_e - 1)(1 - F_e^n)} - 1 \right) + \frac{F_d}{B_{\text{\alpha}} P_s} \right) \frac{1}{1 - \text{CNU}}$$  \hfill (12)

3. Proposed Improvements for AODV protocol

3.1 Proposed Cross-Layer Routing Model

In this paper, we propose a cross-layer routing model shown in Figure 1 to calculate end-to-end delay of routes found by routing process using the concept of service time and to enhance the routing process of AODV protocol in order to select the best routes having smallest end-to-end delay instead of shortest hop count.

Improvements of AODV protocol are deployed in a new route protocol named AODV-DM (Ad hoc On Demand Distance Vector - Delay Metric). This protocol inherited almost the entire operating mechanism of AODV protocol, in addition, it is added two new modules which are Channel Utilization Measurement module used to measure the channel utilization (CNU)of a link's source node and Frame Error Rate Calculation module for calculating frame error rate ($F_e$) of a link. In formula (12), they are main parameters which need to be estimated in order to calculate service time of a link. End-to-end delay calculation and replacing hop count metric by delay metric in AODV-DM protocol are implemented.
via a mechanism sending and receiving RREQ and RREP packets of Routing Mechanism module.

### 3.2 Channel Utilization Measurement

To measure usage of channels at a node of the MAC layer in NS2, we utilize a channel sensing function and build a counter computing the proportion of number of times when the channel is busy and the number of times when the channel is idle. After 0.9ms, we will check the value of the channel sensing function and update the number of time the channel is busy. After 4.5s, we update value of the CNU. The period of checking channel sensing function is selected based on time to transmit one 1173-bytes frame through a IEEE.11b channel at 11Mbps speed. Period to update the value of CNU is selected based on the synchrony with update period of Frame Error Rate value at the Network layer. We are going to explain in more detail about this period in the section 3.3.

However, if using only this method, when channels are being used to transmit data flows having nodes are intermediate nodes on the end-to-end route at the Network layer, status of channels still be counted as busy. The bigger the load of the data stream passing through this node is, the greater the value CNU is. Obviously, this is an undesired effect when calculating the value of CNU. This value should be technically estimated to eliminate the situation that the channel status is busy when it is transmitting frames of the data flow passing through this node on the end-to-end route at the Network layer.

Assume that PNS (Passed Nodes Set) is a set of nodes that the frame has been passed through and SNS (Source Nodes Set) is a set of nodes has passed the frame through the current node. One frame will belong to the data thread passing through the current network node if it satisfies one of two following conditions:

i. Address of the current network node belongs to the PNS set.

ii. Source address of the received frame belongs to the SNS set.

In each period of calculating CNU value, at each update of the number of time the channel is busy, the value of this counter will be increased by one unit if the status of the channel is busy and both conditions (i) and (ii) are not satisfied. To implement the idea of calculating the CNU value of a channel at a sending node, we made two adjustments in AODV implementation as follows:

- Added the PNS field into the header of each data packet to contain address of nodes that the packet went through.
- Add the SNS list at each node to store the list of nodes forwarding the frame through the current node.

When a node receives a frame at the Data Link layer, it has to decapsulate the packet before sending the frame to the upper layer and this packet decapsulation need to be forwarded from the upper layer. Therefore we add the PNS field into the IP packet header to ensure simplicity when manipulating this field during processing forwarded packets. A network node will study the busy status of a channel when another network node is transmitting data if the distance from it to the transmitting node is in its carrier sensing range. Therefore, we only consider the range of influences on the busy status of a channel is from one node to other nodes within 2 hops radius. Therefore, the maximum number of elements in the PNS field is 3. In other words, the PNS field only need contain a maximum of 3 addresses of the nearest network nodes which the frame went through. Algorithms which set PNS field values, set the SNS list value, update the busy channel counter and update the CNU value of the sending node are represented in the Algorithm 1, 2, 3 and 4, respectively. After each update period of AODV routing table (1 second), we reset value of SNS array to ensure the accuracy of the list of source nodes sending the packet through the current node.
Algorithm 1. Update value of PNS field of a frame at MAC layer

Begin
   for each frame need to send do
      if not (my_address ∈ PNS) then
         for i := max_PNS_element - 1 downto 1 do
            PNS[i] := PNS[i-1]
         end for
         PNS[0] := my_address
         <AODV_works>
      end if
   end for
End

Algorithm 2. Update SNS array and in_flow_flag of a node at MAC layer

Node Initialization:

   in_flow_flag := false
   n_SNS := 0

Begin
   for each received frame do
      if (my_address == frame_destination) then
         #Updating value of SNS array
         for each node_addr in PNS do
            if not (node_addr ∈ SNS) then
               n_SNS := n_SNS + 1
               SNS[n_SNS] := node_addr
            end if
         end for
      else
         #Updating value of in_flow_flag
         if ((my_address ∈ PNS) or (frame_source ∈ SNS)) then
            in_flow_flag := true
         end if
      end if
      <AODV_works>
      in_flow_flag := false
   end for
End
### Algorithm 3. Update channel_busy_counter of a node at MAC layer

**Node Initialization:**

| busy_counter:= 0  |
|-------------------|
| total_counter:=0   |

**Begin**

```plaintext
for each channel sensing timer do
  #channel sensing period = 0.9ms
  if (in_flow_flag == false) and (channel_state() == busy) then
    busy_counter := busy_counter + 1
  end if
  total_counter:= total_counter+ 1
end for
End
```

### Algorithm 4. Updating CNU of the sending node at MAC layer

**Node Initialization:**

| CNU:= 0            |

**Begin**

```plaintext
for each CNU updating timer do
  #CNU updating period = 5s
  CNU := busy_counter / total_counter
  send Link_Quality_Update packet to Routing Layer
  busy_counter:= 0
  total_counter:=0
end for
End
```

### 3.3 Frame Error Rate Estimation

There are a number of methods to estimate Frame Error Rate (FER) at the MAC layer. In this paper, we use the Expected Transmission Count (ETX) method proposed by [13] to estimate the FER value for a link in at the MAC layer.

Consider a link between two nodes A and B. In an estimation period, assume that success transmission rate of a
frame from node A to node B is $d_f$ and success transmission rate of that frame from node B to node A is $d_r$. FER value of the link is calculated by the following formula:

$$FER = 1 - d_f \times d_r$$  \hspace{1cm} (13)

To estimate the value of $d_f$ and $d_r$, probe packets are transmitted periodically from node A to node B, and vice versa. Whenever a probe packet is received, one node will increase the value of counter for probe packet number of neighboring node by 1 unit. After one estimation period, value of FER would be $n$ probe packets, each node will send information about the number of probe packets it receives to the node on the other side of the link. Suppose that in one period, node A receives $i$ probe packets and got the information from node B that this node received $j$ probe packets from node A, FER value of the link from node A to node B will be calculated by following formula:

$$FER = 1 - \frac{i}{n} \times \frac{j}{n}$$  \hspace{1cm} (14)

This method was utilized in$^{11}$ to estimate FER value for links at the MAC layer. Probe packets used in here are HELLO packets of OLSR protocol. However, regards to AODV protocol that we choose to enhance, since there are no link break call back mechanism from the MAC layer to the Network layer, HELLO packet is utilized to determine the status of a link whether it is active. If after a period of time which is equal to the number of HELLO packets allowed to be missed multiply with HELLO packet cycle (ALLOWED_HELLO_LOSS x HELLO_INTERVAL), one node has not received a HELLO packet from its neighboring node, it will assume that the link between it and that neighboring node is broken. This node then uses ROUTE_ERROR procedure to informing the link error and reactivate ROUTE_REQUEST procedure. Default sending period of HELLO packet of AODV is 1 second, so, if using HELLO packet in AODV to become probe packet to estimate FER of the link, the estimation period is 10 HELLO packets, FER value of a link will be updated after every 10 seconds. One of value we need to calculate in order to be able to estimate the Service Time of the link using formula (12) is FER of links. Therefore, the estimation cycle of the Service Time need to be equal or greater than 10 seconds if the enhanced AODV protocol utilizes HELLO packet to be probe packet. In order to reduce the length of the estimation period of the Service Time, sending period of HELLO packet might be reduced. However, this will decrease maximum time life of the link. Therefore, the increment of number of calling ROUTE_ERROR procedure and ROUTE_REQUEST procedure will lead to bandwidth waste which is used for control packets.

In AODV-DM protocol, instead of using HELLO packets which is similar in$^{11}$, we build and use a new probe packet named FER_PROBE to calculate FER value of the link and contain the $CNU$ value of link measured at the MAC layer from the sending node to the receiving node of the link. The structure of this package is shown in Figure 2.

The meaning of the fields in the structure of the FER_PROBE packet as follows: TYPE field presents the type of package. LINKS field contains a list of links connected the current node to its neighboring nodes. Each link consists of two subfields which are NB_ID containing address of neighboring node and n_RECV holding the number of FER_PROBE packet the current node has received from the neighboring nodes in the nearest FER estimation period. nNB field contains the number of active links from the current node to its neighboring nodes. WND_ID field consists of value of the current estimation period of FER. CNU field contains the $CNU$ value of the link whose sending node is the current node.

Before sending a FER_PROBE packet, this node will read information from its Neighbor Table to write to LINKS va nNB fields. Value of WND_ID field is updated according to the estimation period of FER and the value
of CNU filed is updated based on the CNU value of the current node measured at the MAC layer. When a node receives a FER_PROBE packet, it will check whether IP source address of the packet is in its Neighbor Table. If not, it will add one new entry in the Neighbor Table. This node then read the element in the LINKS field which has NB_ID is identical with its address and record n_RECV value of this element. If the current FER estimation period of node is bigger than value of WND_ID field of packet, this node will update FER value and CNU of corresponding link with the node sending FER_PROBE packet which it has just received. It then calculates the Service Time value of the link according to formula (12) and store this information into the Neighbor Table.

When implementing AODV-DM protocol in NS2, we selected sending period of the FER_PROBE packet is 0.5 second and FER updating period is 5 seconds (10 FER_PROBE packets). As a result, CNU updating period is selected as 4.5 seconds to make sure that the value of CNU will be updated prior to FER updating period, to serve for the calculation of the value of service time of the link. To minimize sudden changes in values of FER and CNU in link changed suddenly, we use weighted of 0.7 for the current FER and CNU and weights 0.3 for their nearest corresponding values in the pass. More details as follows;

\[
FER = 0.7 \times {new}_{FER} + 0.3 \times FER
\]

(15)

\[
CNU = 0.7 \times {new}_{CNU} + 0.3 \times CNU
\]

(16)

3.4 Routing Mechanism

AODV-DM protocol is built based on the operating principle of AODV protocol. Therefore, the routing mechanism of AODV-DM need to be inherited the loop-free routing rule of AODV routing and replacing hop count by delay in routing metric. In this paper, we propose a method for estimating the delay of a route by summing up the service times of the links belonging to that route. According to the formula (12), service time of a link depends mainly on two parameters of that link: CNU and FER. Therefore, a route whose links traverse congested areas will have greater service time value. Selecting a route whose total service time value is the smallest will ensure that AODV-DM protocol sends data packets avoiding network congestion areas.

To ensure that the rule of using destination sequence number and hop count to avoid loop routing and integrated routing metric proposed in AODV-DM, we made some changes mentioned below in some control and data packets for the operation of AODV protocol.

- Adding RQ_DELAY field to RREQ packet.
- Adding RP_DELAY field to RREP packet.
- Adding PATH_DELAY field to the Routing Table.
- Adding SERVICE_TIME field to each entry of the Neighbor Table.

PQ_DELAY field and RP_DELAY field contain the delay of reverse route and forward route established in ROUTE_REQUEST and ROUTE_REPLY procedure, respectively. PATH_DELAY field of one entry of the routing table store the value of delay of route represented by the entry. SERVICE_TIME field contains service time value of the link from the neighboring node to the current node. Operating mechanism of routing procedure is presented in Algorithm 5 and 6. Algorithm 7 shows route reply mechanism of AODV-DM protocol.

To ensure the inheritance of the loop-free rule of AODV, we still remain the hop count parameter of routes. In AODV, one route in the routing table is only updated when the destination sequence number value of the new route is greater than that of the current route or this value of these two routes are equal and the hop count value of the new route is smaller than that of the current route. In AODV-DM protocol, in the route requesting and replying process, a route in the routing table is also updated when it noticed a newer route (the value of destination sequence number is greater). If the node received a new route whose destination sequence number is equal with that of the current node, this node only updates the new route if its hop count is equal and delay is smaller than those of the current route.
Algorithm 5. RREQ Packet Sending

Begin
  for each RREQ packet sent do
    <AODV_works>
    RQ_DELAY := 0
  end for
End

Algorithm 6. RREQ Packet Receiving

Begin
  for each RREQ packet received do
    #update path delay of RREQ packet
    NB_Addr := Address of node sent RREQ
    if (NB_Addr ∈ Neighbor Table) then
      read NB from Neighbor Table #link from NB to current node
      RQ_DELAY := NB.SERVICE_TIME + RQ_DELAY
    end if
    <AODV works>
    #update exist reverse path
    if (RREQ.sequence_number > reverse_path.sequence_number) or
      ((RREQ.sequence_number == reverse_path.sequence_number) and
      (RREQ.hop_count == reverse_path.hop_count) and
      (RQ_DELAY < reverse_path.PATH_DELAY)) then
      update reverse_path with new delay and other AODV parameters
    end if
    <AODV works>
    #send RREP if node is destination or having a forward path to destination
    if (current node is destination) then
      send RREP with RP_DELAY = 0 and other AODV parameters
    else if (current node have a forward path to destination) then
      send RREP with RP_DELAY = forward_path. DELAY and other AODV parameters
    end if
    <AODV works>
  end for
End
4. Performance Evaluation

4.1 Simulation Scenarios and Performance Metrics

To evaluate the performance of AODV-DM protocol in the comparison with AODV protocol, we use NS2 simulator to run a scenario shown in Figure 3. The matrix topology consists of 81 nodes distributed in a square space 350mx350m. The distance between the rows and columns of nodes is 40m. Technology is used in the Data Link layer and the Physical layer is IEEE 802.11b, propagation model is Two Ray Ground, the maximum transmission speed is 11 Mbps, maximum transmission distance up to 50m and maximum channel sensing distance is 110m. There are two CBR data flows transmitted between two nodes distributed in two opposite corners of the matrix with the source-destination node pairs are 0-16 and 24-8. The rate of data flows changes from 50kbps to 100kbps, 150kbps, 200kbps and 250kbps. To simulate interference flows, we used 4 CBR flows at the center of the matrix with the source node is node 80 and the destination nodes are 73, 75, 77 and 79. The rate of the interference flows are also changed from 0.1Mpbs to 0.2Mpbs, 0.4Mpbs, 0.6Mbps, 0.8Mpbs and 1.0Mpbs for testing its impact to the performance of the two protocols evaluated. The nodes distributed in the second round from the edge of the matrix (from node 32 to node 55) move constantly counterclockwise with the moving speed of 5m/s. After a period of moving, these nodes will break their links with the source or destination nodes from the edge to initialize the new route discovery.

Algorithm 7. RREP Packet Receiving

```
Begin
  for each RREP packet received do
    #update path delay of RREQ packet
    NB_Addr := Address of node sent RREP
    if (NB_Addr ∈ Neighbor Table) then
      read NB from Neighbor Table #link from NB to current node
      RP_DELAY := NB.SERVICE_TIME + RP_DELAY
    end if
    <!--AODV_works--> 
    #update exist forward path
    if (RREP.sequence_number > forward_path.sequence_number) or
      ((RREP.sequence_number == forward_path.sequence_number) and
      (RREP.hop_count == forward_path.hop_count) and
      (RP_DELAY < forward_path.PATH_DELAY)) then
      update forward_path with new delay and other AODV parameters
    end if
    <!--AODV_works-->
  end for
End
```
process. Link broken report mechanism directly from the MAC layer to the Routing layer in AODV is disable in order to use HELLO packets in determining whether a link is active or broken. Simulation process takes place in 200 seconds, in which the interference flows are started at 2nd second, data streams are started at 8th second, the movement of nodes in the moving circle started at 10th second. With each simulation parameter set including data rate and interference rate, we simulated 10 times with different seeds for each simulator running time.

There are four parameters used to assess the performance between AODV-DM protocol and AODV protocol, including:

- **Average Throughput**: The amount of data received at the destination node in the Application layer in a unit time (1 second). Unit is bps.

- **Packet Delivery Ratio**: Proportion of successful packet transmission which is calculated by the percentage of packets received at the destination node and the total packets sent at the source node.

- **Average Packet Delay**: The average end-to-end delay which is calculated from sending the packet from the source node until receiving the packet at the destination node. Unit is milliseconds.

- **Routing Overhead**: Percentage of control packet number of the routing protocol on the total number of packets.

### 4.2 Results

In the first experiment, we measured the average throughput at destination nodes of data flows having rate 50kbps and 250kbps and changing the rate of interference flows at the center from 0.1 to 0.2, 0.4, 0.6, 0.8 and 1.0Mpbs. Throughput measurement results illustrated in Figure 4 show that AODV-DM protocols achieve greater throughput than AODV protocol in both two data rates. At the 50kbps and 150 kbps data rate, the average throughput of AODV-DM protocol is approximately 29% and 25% higher, respectively compared with AODV protocol, so the difference between the average throughputs of these
protocols is 27%. AODV routing protocol chooses the smallest hop count route, so these routes tend to pass through the center of the network where is congested by interference flows. AODV-DM protocol prioritizes routes with minimal delay calculated based on service time, therefore selected routes will avoid congested areas at the network center. This leads to throughput at destination nodes in AODV-DM protocol is higher than that of AODV protocol.

![Average Throughput vs. Interference Flow Rate](image)

**Figure 4.** Average Throughput vs. Interference Flow Rate.

![Packet Delivery Ratio vs. Interference Flow Rate](image)

**Figure 5.** Packet Delivery Ratio vs. Interference Flow Rate.
We compared the performance of two protocols in terms of packet delivery ratio (PDR) when we fixed data transfer rate at 200kbps and changed the rate of the interference flows from 0.1Mbps to 1.0Mbps. The results of PDR of the two protocols in the scenario is given in Figure 5. It is possible to see that when the rate of the interference flows increase, the PDR of both protocols are reduced. However, at all six different rates of the interference flows used in simulation, the PDR of AODV-DM protocol is higher than that of AODV protocol. The average difference of PDR of the two protocols is approximately 30%. This can be explained as AODV-DM protocol selects routes which do not go through the center of interference flows (congestion area), so the effect of interference flows should be less on AODV-DM protocol than on AODV protocol.

In the third simulation, we fixed data rate at 50kbps and 150kbps while changing interference rate from 0.1Mbps to 1.0Mbps. Average delay of sending a packet from source to destination is measured. The result shown in Figure 6 shows that average delay of sending packets in AODV-DM protocol is lower than that in AODV protocol in both data rates. The difference between the average packet delays of these protocols is approximately 25% and 39% with 50kbps and 150kbps data rate, respectively; so the average difference is 32%. This can be explained by the routing mechanism of AODV-DM routing protocol compared with AODV protocol. Because of prioritizing routes with small delay so packet delay when sending packets from source to destination of AODV-DM protocol is smaller than that of AODV protocol which priories routes with smaller hop count.

The last simulation was used to evaluate the performance of Routing Overhead parameter of both two protocols and the results are shown in Figure 7. In this test, the rate of data flows is fixed at 150kbps and rate of interference flows is changed from 0.1Mbps to 1.0Mbps. For all interference rates, the routing overhead of AODV-DM protocols is smaller than that of AODV protocol. The difference between the routing overheads of these protocols is approximately 40%. This result is explained by the route choice of AODV-DM protocol, which tend to choose routes go through edges while AODV protocol tends to select routes going through the center of the matrix. Therefore, number of moving nodes belonging to data transmission routes of AODV-DM protocol is smaller than that of AODV protocol. It leads to the number of broken links on the path in the simulation duration is

![Figure 6. Average Packet Delay vs. Interference Flow Rate.](image-url)
smaller and as a result, the number of calling new routing procedure of AODV-DM protocol is smaller than that of AODV.

5. Conclusion

In this paper, we have proposed enhancements of AODV protocol in multi-hop ad hoc networks having congested areas. These enhancements are based on replacing route selection mechanism using hop count of AODV routing protocol by route selection mechanism using delay. We utilized cross-layer approach to get delay of links at the MAC layer and use it to calculate end-to-end delay of routes at the Network layer. Delay of each link in the MAC layer is estimated based on channel utilization of source nodes and frame error rate of links. Techniques used to measure channel utilization of one node and to estimate the frame error rate of links have been introduced in this paper. We have also proposed adjustments of AODV to implements a new protocol called AODV-DM. The results of the performance evaluation of the two protocols show that the protocol recommended by us achieves better performance than AODV protocol in terms of throughput, success packet transmission rate, average packet transmission delay and control packet rate.

6. References

1. Perkins C, Belding-Royer E, Das S. Ad hoc on-demand distance vector (AODV) routing. Network Working Group, Error! Hyperlink reference not valid. 2003; p. 1-37.
2. Barua G, Agarwal M. Caching of Routes in Ad hoc On-Demand Distance Vector (AODV) Routing for Mobile Ad hoc Networks. Mumbai, India: ICCC Proceedings of the 15th International Conference on Computer Communication. 2002 Aug; p. 1-12.
3. Tamilarasi M, Palanivelu T. Adaptive Link Timeout with Energy Aware Mechanism for On-Demand Routing in MANETs. Ubiquitous Computing and Communication Journal. 2009 Oct; 4(4):96-102.
4. Alshanyour A, Baroudi U. Bypass-AODV: improving performance of ad hoc on-demand distance vector (AODV) routing protocol in wireless ad hoc networks. Athens, Greece: Proceedings of International Conference on Ambient Media and Systems. 2008; p. 17.
5. Lee SJ, Gerla M. AODV-BR: backup routing in ad hoc networks. Chicago, USA: Proceedings of Wireless Communications and Networking Conference (WCNC. IEEE), WCNC’00. 2000; 3:1311-16.
6. Lai WK, Hsiao S-Y, Lin Y-C. Adaptive backup routing for ad-hoc networks. Journal of Computer Communications. 2007 Jan; 30(2):453-64.
7. AL-Gabri M, Chunlin L, Zhiyong Y, Hasan A, Xiaoqing Z. Improved the Energy of Ad Hoc On-Demand Distance
Vector Routing Protocol. Seoul, Korea: Proceeding of the International Conference on Future Computer Supported Education. 2012 Aug; p. 355-61.

8. Madhan Mohan R, Selvakumar K. Power controlled routing in wireless ad hoc networks using cross layer approach. Egyptian Informatics Journal. 2012 Jul; 13(2):95-101.

9. Srinivasan P, Kamalakkannan P. Enhancing Route Maintenance in RSEA-AODV for Mobile Ad Hoc Networks. Coimbatore, Tamil Nadu, India: Proceeding of the 7th International Conference on Intelligent Systems and Control (ISCO). 2003 Jan; p. 464-69.

10. Chib R, Sandhu AS. Enhancing Routing Performance of AODV Protocol using 3rd Ordered Newton’s Difference Equation. Indian Journal of Science and Technology. 2016 Apr; 9(15):1-9.

11. Nguyen LT, Beuran R, Shinoda Y. An interference and load aware routing metric for Wireless Mesh Networks. International Journal of Ad Hoc and Ubiquitous Computing. 2011; 7(1):25-37.

12. Awerbuch B, Holmer D, Rubens H. The medium time metric: high throughput route selection in multi-rate ad hoc wireless networks. Mobile Networks and Applications. 2006 Apr; 11(2):253-66.

13. De Couto DSJ, Aguayo D, Bicket J, Morris R. A high-throughput path metric for multi-hop wireless routing. San Diego, California, USA: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking. 2003; p. 134-46.