IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 13, Issue 6, Ver. I (Nov.-Dec. 2018), PP 46-53
www.iosrjournals.org

Ad hoc On-Demand Distance Vector (AODV) Routing

Shivaleela I. Hubli[1], Prof Hemanth Kelagadi[2], Dr Priyatamkumar[3]

[1]*(Electronics and Communication Engineering, B. V. Bhoomaraddi College of Engineering and Technology, India)*
[2]*(Electronics and Communication Engineering, B. V. Bhoomaraddi College of Engineering and Technology, India)*
[3]*(Electronics and Communication Engineering, B. V. Bhoomaraddi College of Engineering and Technology, India)*
Corresponding Author: Shivaleela I. Hubli

Abstract: *The Ad hoc On-Demand Distance Vector (AODV) routing protocol is intended for use by mobile nodes in an ad hoc network. It offers quick adaptation to dynamic link conditions, low processing and memory overhead, low network utilization, and determines unicast routes to destinations within the ad hoc network. It uses destination sequence numbers to ensure loop freedom at all times, even in the face of anomalous delivery of routing control messages, avoiding problems such as "counting to infinity" associated with classical distance vector protocols.*

Keywords - *Ad hoc On-Demand Distance Vector, ad hoc network, distance vector protocol, mobile, routing.*

Date of Submission: 21-12-2018 Date of acceptance: 05-01-2019

I. Introduction

The Ad hoc On-Demand Distance Vector (AODV) algorithm enables dynamic, self-starting, multi-hop routing between participating mobile nodes wishing to establish and maintain an ad hoc network. AODV allows mobile nodes to obtain routes quickly for new destinations, and does not require nodes to maintain routes to destinations that are not in active communication. AODV allows mobile nodes to respond to link breakages and changes in network topology in a timely manner. The operation of AODV is loop-free, and by avoiding the Bellman-Ford "counting to infinity" problem offers quick convergence when the ad hoc network topology changes (typically, when a node moves in the network). When links break, AODV causes the affected set of nodes to be notified so that they are able to invalidate the routes using the lost link.

One distinguishing feature of AODV is its use of a destination sequence number for each route entry. The destination sequence number is created by the destination to be included along with any route information it sends to requesting nodes. Using destination sequence numbers ensures loop freedom and is simple to program. Given the choice between two routes to a destination, a requesting node is required to select the one with the greatest sequence number.

II. Overview

Route Requests (RREQs), Route Replies (RREPs), and Route Errors (RERRs) are the message types defined by AODV. These message types are received via UDP, and normal IP header processing applies. So, for instance, the requesting node is expected to use its IP address as the Originator IP address for the messages. For broadcast messages, the IP limited broadcast address (255.255.255.255) is used. This means that such messages are not blindly forwarded. However, AODV operation does require certain messages (e.g., RREQ) to be disseminated widely, perhaps throughout the ad hoc network. The range of dissemination of such RREQs is indicated by the TTL in the IP header. Fragmentation is typically not required. AODV is a routing protocol, and it deals with route table management. Route table information must be kept even for short-lived routes, such as are created to temporarily store reverse paths towards nodes originating RREQs. AODV uses the following fields with each route table entry.

2.1. Destination IP Address
1.2. Valid Destination Sequence Number flag
1.3. Other state and routing flags (e.g., valid, invalid, repairable, being repaired)
1.4. Network Interface
1.5. Hop Count (number of hops needed to reach destination)
1.6. Next Hop
1.7. List of Precursors
1.8. Lifetime (expiration or deletion time of the route)

III. AODV Terminology

This protocol specification uses conventional meanings [1] for capitalized words such as MUST, SHOULD, etc., to indicate requirement levels for various protocol features. This section defines other terminology used with AODV that is not already defined in [3].

3.1. **active route -** A route towards a destination that has a routing table entry that is marked as valid. Only active routes can be used to forward data packets.
3.2. **broadcast -** Broadcasting means transmitting to the IP Limited Broadcast address, 255.255.255.255. A broadcast packet may not be blindly forwarded, but broadcasting is useful to enable dissemination of AODV messages throughout the ad hoc network.
3.3. **destination -** An IP address to which data packets are to be transmitted. Same as "destination node". A node knows it is the destination node for a typical data packet when its address appears in the appropriate field of the IP header. Routes for destination nodes are supplied by action of the AODV protocol, which carries the IP address of the desired destination node in route discovery messages.
3.4. **forwarding node -** A node that agrees to forward packets destined for another node, by retransmitting them to a next hop that is closer to the unicast destination along a path that has been set up using routing control messages.
3.5. **forward route -** A route set up to send data packets from a node originating a Route Discovery operation towards its desired destination.
3.6. **invalid route -** A route that has expired, denoted by a state of invalid in the routing table entry. An invalid route is used to store previously valid route information for an extended period of time. An invalid route cannot be used to forward data packets, but it can provide information useful for route repairs, and also for future RREQ messages.
3.7. **originating node -** A node that initiates an AODV route discovery message to be processed and possibly retransmitted by other nodes in the ad hoc network. For instance, the node initiating a Route Discovery process and broadcasting the RREQ message is called the originating node of the RREQ message.
3.8. **reverse route -** A route set up to forward a reply (RREP) packet back to the originator from the destination or from an intermediate node having a route to the destination.
3.9. **sequence number -** A monotonically increasing number maintained by each originating node. In AODV routing protocol messages, it is used by other nodes to determine the freshness of the information contained from the originating node.

IV. Application

The AODV routing protocol is designed for mobile ad hoc networks with populations of tens to thousands of mobile nodes. AODV can handle low, moderate, and relatively high mobility rates, as well as a variety of data traffic levels. AODV is designed for use in networks where the nodes can all trust each other, either by use of preconfigured keys, or because it is known that there are no malicious intruder nodes. AODV has been designed to reduce the dissemination of control traffic and eliminate overhead on data traffic, in order to improve scalability and performance.

V. Message Formats

The AODV routing protocol is designed for mobile ad hoc networks with populations of tens to thousands of mobile nodes. AODV can handle low, moderate, and relatively high mobility rates, as well as a variety of data traffic levels. AODV is designed for use in networks where the nodes can all trust each other, either by use of preconfigured keys, or because it is known that there are no malicious intruder nodes. AODV has been designed to reduce the dissemination of control traffic and eliminate overhead on data traffic, in order to improve scalability and performance.

3.1. Route Request (RREQ) - The format of the Route Request message is illustrated in Table 5.1, and fields are explained in the Table 5.2.
3.2. Route Reply (RREP) - The format of the Route Reply message is illustrated in Table 5.3, and fields are explained in the Table 5.4.
3.3. Route Error (RERR) - The format of the Route Error message is illustrated in Table 5.5, and fields are explained in the Table 5.6.
3.4. Route Reply Acknowledgement (RREP-ACK) - The format of the Route Reply Acknowledgement message is illustrated in Table 5.7, and fields are explained in the Table 5.8.

VI. AODV Operation

This section describes the scenarios under which nodes generate Route Request (RREQ), Route Reply (RREP) and Route Error (RERR) messages for unicast communication towards a destination, and how the message data are handled. In order to process the messages correctly, certain state information has to be maintained in the route table entries for the destinations of interest.

6.1. Maintaining Sequence Numbers

Every route table entry at every node MUST include the latest information available about the sequence number for the IP address of the destination node for which the route table entry is maintained. This sequence number is called the "destination sequence number". It is updated whenever a node receives new (i.e., not stale) information about the sequence number from RREQ, RREP, or RERR messages that may be received related to that destination. AODV depends on each node in the network to own and maintain its destination sequence number to guarantee the loop-freedom of all routes towards that node. A destination node increments its own sequence number in two circumstances.

One, immediately before a node originates a route discovery, it MUST increment its own sequence number. This prevents conflicts with previously established reverse routes towards the originator of a RREQ. Second, immediately before a destination node originates a RREP in response to a RREQ, it MUST update its own sequence number to the maximum of its current sequence number and the destination sequence number in the RREQ packet.

In order to ascertain that information about a destination is not stale, the node compares its current numerical value for the sequence number with that obtained from the incoming AODV message. This comparison MUST be done using signed 32-bit arithmetic, this is necessary to accomplish sequence number rollover. If the result of subtracting the currently stored sequence number from the value of the incoming sequence number is less than zero, then the information related to that destination in the AODV message MUST be discarded, since that information is stale compared to the node's currently stored information.

6.2. Generating Route Requests

A node disseminates a RREQ when it determines that it needs a route to a destination and does not have one available. This can happen if the destination is previously unknown to the node, or if a previously valid route to the destination expires or is marked as invalid. The Destination Sequence Number field in the RREQ message is the last known destination sequence number for this destination and is copied from the Destination Sequence Number field in the routing table. If no sequence number is known, the unknown sequence number flag MUST be set. The Originator Sequence Number in the RREQ message is the node's own sequence number, which is incremented prior to insertion in a RREQ. The RREQ ID field is incremented by one from the last RREQ ID used by the current node. Each node maintains only one RREQ ID. The Hop Count field is set to zero.

Before broadcasting the RREQ, the originating node buffers the RREQ ID and the Originator IP address (its own address) of the RREQ for PATH_DISCOVERY_TIME. In this way, when the node receives the packet again from its neighbors, it will not reprocess and re-forward the packet.

An originating node often expects to have bidirectional communications with a destination node. In such cases, it is not sufficient for the originating node to have a route to the destination node; the destination must also have a route back to the originating node. In order for this to happen as efficiently as possible, any generation of a RREP by an intermediate node for delivery to the originating node SHOULD be accompanied by some action that notifies the destination about a route back to the originating node. The originating node selects this mode of operation in the intermediate nodes by setting the 'G' flag.

A node SHOULD NOT originate more than RREQ_RATELIMIT RREQ messages per second. After broadcasting a RREQ, a node waits for a RREP (or other control message with current information regarding a route to the appropriate destination). If a route is not received within NET_TRAVERSAL_TIME milliseconds, the node MAY try again to discover a route by broadcasting another RREQ, up to a maximum of RREQ_RETRIES times at the maximum TTL value. Each new attempt MUST increment and update the RREQ ID. For each attempt, the TTL field of the IP header is set according to the mechanism, in order to enable control over how far the RREQ is disseminated for the each retry.

To reduce congestion in a network, repeated attempts by a source node at route discovery for a single destination MUST utilize a binary exponential backoff. The first time a source node broadcasts a RREQ, it waits NET_TRAVERSAL_TIME milliseconds for the reception of a RREP. If a RREP is not received within that time, the source node sends a new RREQ. When calculating the time to wait for the RREP after sending the second RREQ, the source node MUST use a binary exponential backoff. Hence, the waiting time for the RREP corresponding to the second RREQ is 2 * NET_TRAVERSAL_TIME milliseconds. If a RREP is not received within this time period, another RREQ may be sent, up to RREQ_RETRIES additional attempts after the first

RREQ. For each additional attempt, the waiting time for the RREP is multiplied by 2, so that the time conforms to a binary exponential backoff. The hop count is copied from the Hop Count in the RREQ message.

Whenever a RREQ message is received, the Lifetime of the reverse route entry for the Originator IP address is set to be the maximum of (ExistingLifetime, MinimalLifetime), where

MinimalLifetime = (current time + 2*NET_TRAVERSAL_TIME -
2*HopCount*NODE_TRAVERSAL_TIME).

The current node can use the reverse route to forward data packets in the same way as for any other route in the routing table. Processing and Forwarding Route Requests

When a node receives a RREQ, it first creates or updates a route to the previous hop without a valid sequence number then checks to determine whether it has received a RREQ with the same Originator IP Address and RREQ ID within at least the last PATH_DISCOVERY_TIME. If such a RREQ has been received, the node silently discards the newly received RREQ.

6.3. Processing and Forwarding Route Requests

When a node receives a RREQ, it first creates or updates a route to the previous hop without a valid sequence number then checks to determine whether it has received a RREQ with the same Originator IP Address and RREQ ID within at least the last PATH_DISCOVERY_TIME. If such a RREQ has been received, the node silently discards the newly received RREQ.

First, it first increments the hop count value in the RREQ by one, to account for the new hop through the intermediate node. Then the node searches for a reverse route to the Originator IP Address, using longest-prefix matching. If need be, the route is created, or updated using the Originator Sequence Number from the RREQ in its routing table. This reverse route will be needed if the node receives a RREP back to the node that originated the RREQ (identified by the Originator IP Address). When the reverse route is created or updated, the following actions on the route are also carried out.

6.3.1. The Originator Sequence Number from the RREQ is compared to the corresponding destination sequence number in the route table entry and copied if greater than the existing value there.
6.3.2. The valid sequence number field is set to true.
6.3.3. The next hop in the routing table becomes the node from which the RREQ was received (it is obtained from the source IP address in the IP header and is often not equal to the Originator IP Address field in the RREQ message).

6.4. Generating Route Replies

A node generates a RREP if either - it is itself the destination, or it has an active route to the destination, the destination sequence number in the node's existing route table entry for the destination is valid and greater than or equal to the Destination Sequence Number of the RREQ (comparison using signed 32-bit arithmetic), and the "destination only" ('D') flag is NOT set.

When generating a RREP message, a node copies the Destination IP Address and the Originator Sequence Number from the RREQ message into the corresponding fields in the RREP message. Processing is slightly different, depending on whether the node is itself the requested destination, or instead if it is an intermediate node with a fresh enough route to the destination.

Once created, the RREP is unicast to the next hop toward the originator of the RREQ, as indicated by the route table entry for that originator. As the RREP is forwarded back towards the node which originated the RREQ message, the Hop Count field is incremented by one at each hop. Thus, when the RREP reaches the originator, the Hop Count represents the distance, in hops, of the destination from the originator.

6.5. Receiving and Forwarding Route Replies

When a node receives a RREP message, it searches (using longest-prefix matching) for a route to the previous hop. If needed, a route is created for the previous hop, but without a valid sequence number. Next, the node then increments the hop count value in the RREP by one, to account for the new hop through the intermediate node. Call this incremented value the "New Hop Count". Then the forward route for this destination is created if it does not already exist. Otherwise, the node compares the Destination Sequence Number in the message with its own stored destination sequence number for the Destination IP Address in the RREP message. Upon comparison, the existing entry is updated only in the following circumstances as follows.
6.5.1. The sequence number in the routing table is marked as invalid in route table entry.
6.5.2. The Destination Sequence Number in the RREP is greater than the node's copy of the destination sequence number and the known value is valid, or
6.5.3. The sequence numbers are the same, but the route is marked as inactive, or

6.5.4. The sequence numbers are the same, and the New Hop Count is smaller than the hop count in route table entry.

If the route table entry to the destination is created or updated, then the following actions occur.

i. The route is marked as active.
ii. The destination sequence number is marked as valid,
iii. The next hop in the route entry is assigned to be the node from which the RREP is received, which is indicated by the source IP address field in the IP header,
iv. The hop count is set to the value of the New Hop Count,
v. The expiry time is set to the current time plus the value of the Lifetime in the RREP message,
vi. The destination sequence number is the Destination Sequence Number in the RREP message.

The current node can subsequently use this route to forward data packets to the destination. If the current node is not the node indicated by the Originator IP Address in the RREP message AND a forward route has been created or updated as described above, the node consults its route table entry for the originating node to determine the next hop for the RREP packet, and then forwards the RREP towards the originator using the information in that route table entry. If a node forwards a RREP over a link that is likely to have errors or be unidirectional, the node SHOULD set the 'A' flag to require that the recipient of the RREP acknowledge receipt of the RREP by sending a RREP-ACK message back.

When any node transmits a RREP, the precursor list for the corresponding destination node is updated by adding to it the next hop node to which the RREP is forwarded. Also, at each node the (reverse) route used to forward a RREP has its lifetime changed to be the maximum of (existing-lifetime, (current time + ACTIVE_ROUTE_TIMEOUT). Finally, the precursor list for the next hop towards the destination is updated to contain the next hop towards the source.

6.6. Route Error (RERR) Messages, Route Expiry and Route Deletion

Generally, route error and link breakage processing requires the following steps.

6.6.1. Invalidating existing routes
6.6.2. Listing affected destinations
6.6.3. Determining which, if any, neighbors may be affected
6.6.4. Delivering an appropriate RERR to such neighbors

A Route Error (RERR) message MAY be either broadcast (if there are many precursors), unicast (if there is only 1 precursor), or iteratively unicast to all precursors (if broadcast is inappropriate). Even when the RERR message is iteratively unicast to several precursors, it is considered to be a single control message for the purposes of the description in the text that follows. With that understanding, a node SHOULD NOT generate more than RERR_RATELIMIT RERR messages per second.

A node initiates processing for a RERR message in three situations.

vii. If it detects a link break for the next hop of an active route in its routing table while transmitting data (and route repair, if attempted, was unsuccessful), or
viii. If it gets a data packet destined to a node for which it does not have an active route and is not repairing (if using local repair), or
ix. If it receives a RERR from a neighbor for one or more active routes.

For case i, the node first makes a list of unreachable destinations consisting of the unreachable neighbor and any additional destinations in the local routing table that use the unreachable neighbor as the next hop. In this case, if a subnet route is found to be newly unreachable, an IP destination address for the subnet is constructed by appending zeroes to the subnet prefix as shown in the route table entry. This is unambiguous, since the precursor is known to have route table information with a compatible prefix length for that subnet.

For case ii, there is only one unreachable destination, which is the destination of the data packet that cannot be delivered. For case iii, the list should consist of those destinations in the RERR for which there exists a corresponding entry in the local routing table that has the transmitter of the received RERR as the next hop.

Some of the unreachable destinations in the list could be used by neighboring nodes, and it may therefore be necessary to send a new RERR. The RERR should contain those destinations that are part of the created list of unreachable destinations and have a non-empty precursor list.

Just before transmitting the RERR, certain updates are made on the routing table that may affect the destination sequence numbers for the unreachable destinations. For each one of these destinations, the corresponding routing table entry is updated as follows.

The destination sequence number of this routing entry, if it exists and is valid, is incremented for cases i and ii above, and copied from the incoming RERR in case iii above. The entry is invalidated by marking the route entry as invalid. The Lifetime field is updated to current time plus DELETE_PERIOD. Before this time, the entry SHOULD NOT be deleted.

6.7. Interfaces

Because AODV should operate smoothly over wired, as well as wireless, networks, and because it is likely that AODV will also be used with multiple wireless devices, the particular interface over which packets arrive must be known to AODV whenever a packet is received. This includes the reception of RREQ, RREP, and RERR messages.

Whenever a packet is received from a new neighbor, the interface on which that packet was received is recorded into the route table entry for that neighbor, along with all the other appropriate routing information. Similarly, whenever a route to a new destination is learned, the interface through which the destination can be reached is also recorded into the destination's route table entry.

When multiple interfaces are available, a node retransmitting a RREQ message rebroadcasts that message on all interfaces that have been configured for operation in the ad-hoc network, except those on which it is known that all of the nodes neighbors have already received the RREQ For instance, for some broadcast media (e.g., Ethernet) it may be presumed that all nodes on the same link receive a broadcast message at the same time. When a node needs to transmit a RERR, it SHOULD only transmit it on those interfaces that have neighboring precursor nodes for that route.

VII. Figures and Tables

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Type								J	R	G	D	U	Reserved											Hop count							
RREQ ID																															
Destination IP address																															
Destination sequence number																															
Originator IP address																															
Originator sequence number																															

Table 5.1 Route Request message format

Type	1
J	Join flag; reserved for multicast.
R	Repair flag; reserved for multicast.
G	Gratuitous RREP flag; indicates whether a gratuitous RREP should be unicast to the node specified in the Destination IP Address field
D	Destination only flag; indicates only the destination may respond to this RREQ
U	Unknown sequence number; indicates the destination sequence number is unknown
Reserved	Sent as 0; ignored on reception.
Hop count	The number of hops from the Originator IP Address to the node handling the request.
RREQ ID	A sequence number uniquely identifying the particular RREQ when taken in conjunction with the originating node's IP address.
Destination IP Address	The IP address of the destination for which a route is desired.
Destination Sequence Number	The latest sequence number received in the past by the originator for any route towards the destination.
Originator IP Address	The IP address of the node which originated the Route Request.
Originator Sequence Number	The current sequence number to be used in the route entry pointing towards the originator of the route request.

Table 5.2 Route Request message format fields

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Type								R	A	Reserved									Prefix size					Hop count							
Destination IP address																															
Destination sequence number																															
Originator IP address																															
Lifetime																															

Table 5.3 Route Reply message format

Type	2
R	Repair flag; reserved for multicast.
A	Acknowledgement required
Reserved	Sent as 0; ignored on reception.
Prefix Size	If nonzero, the 5-bit Prefix Size specifies that the indicated next hop may be used for any nodes with the same routing prefix as the requested destination.
Hop count	The number of hops from the Originator IP Address to the Destination IP Address. For multicast route requests this indicates the number of hops to the multicast tree member sending the

	RREP.
Destination IP Address	The IP address of the destination for which a route is supplied.
Destination Sequence Number	The destination sequence number associated to the route.
Originator IP Address	The IP address of the node which originated the Route Request for which the route is supplied.
Lifetime	The time in milliseconds for which nodes receiving the RREP consider the route to be valid.

Table 5.4 Route Reply message format fields

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Type								N	Reserved															Dest count							
Unreachable Destination IP address																															
Unreachable Destination sequence number																															
Additional Unreachable Destination IP address (if needed)																															
Additional Unreachable Destination sequence address (if needed)																															

Table 5.5 Route Error message format

Type	3
N	No delete flag; set when a node has performed a local repair of a link, and upstream nodes should not delete the route.
Reserved	Sent as 0; ignored on reception.
DestCount	The number of unreachable destinations included in the message; MUST be at least 1.
Unreachable Destination IP Address	The IP address of the destination that has become unreachable due to a link break.
Unreachable Destination Sequence Number	The sequence number in the route table entry for the destination listed in the previous Unreachable Destination IP Address field.

Table 5.6 Route Error message format fields

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Type								Reserved							

Table 5.7 Route Reply Acknowledgement message format

Type	4
Reserved	Sent as 0; ignored on reception.

Table 5.8 Route Reply Acknowledgement message format fields

VIII. Conclusion

AODV has been designed for use by mobile nodes with IP addresses that are not necessarily related to each other, to create an ad hoc network. However, in some cases a collection of mobile nodes MAY operate in a fixed relationship to each other and share a common subnet prefix, moving together within an area where an ad hoc network has formed. Call such a collection of nodes a "subnet". In this case, it is possible for a single node within the subnet to advertise reachability for all other nodes on the subnet, by responding with a RREP message to any RREQ message requesting a route to any node with the subnet routing prefix. Call the single node the "subnet router". In order for a subnet router to operate the AODV protocol for the whole subnet, it has to maintain a destination sequence number for the entire subnet. In any such RREP message sent by the subnet router, the Prefix Size field of the RREP message MUST be set to the length of the subnet prefix. Other nodes sharing the subnet prefix SHOULD NOT issue RREP messages, and SHOULD forward RREQ messages to the subnet router.

Acknowledgements

This research on Ad hoc On-demand Distance Vector (AODV) routing is the outcome of guidance, moral support of guide and reviewers. For this, I, acknowledge and express who have been a source of inspiration, guidance and teaching during the project. First and foremost, I thank my research guides Dr PriyatamKumar and Prof Hemanth Kelagadi for guiding me throughout the research constantly and supporting me through their knowledge. My humble thanks to the review committee for instructing me to modify the things in the research and making the research a successful one. I thank all the teaching and non-teaching staff of the department for their co-operation during the course of this research. Lastly, I thank my parents, friends and finally the Almighty for helping me directly or indirectly in making this project successful.

References

[1]. Bradner, S. "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[2]. Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[3]. Manner, J., et al., "Mobility Related Terminology", Work in Progress, July 2001.

[4]. Karthikeyan Bhargavan, Carl A. Gunter, and Davor Obradovic. Fault Origin Adjudication. In Proceedings of the Workshop on Formal Methods in Software Practice, Portland, OR, August 2000.

[5]. IEEE 802.11 Committee, AlphaGraphics #35, 10201 N.35th Avenue, Phoenix AZ 85051. Wireless LAN Medium Access Control MAC and Physical Layer PHY Specifications, June 1997. IEEE Standard 802.11-97.

[6]. Perkins, C., Royer, E. and S. Das, "Ad hoc on demand distance vector (AODV) routing for ip version 6", Work in Progress.

Shivaleela I. Hubli. " Ad hoc On-Demand Distance Vector (AODV) Routing." IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) 13.6 (2018): 46-53.