

Lecture 16 Part A: Ad-hoc On-Demand Distance Vector Routing

Instructor: Soma Chaudhuri

Scribe: Danyu Liu

1 Lecture topic

This lecture is about the presentation of the paper *Ad-hoc On-Demand Distance Vector Routing* written by Charles E. Perkins and Elizabeth M. Royer. The presentation was done by Danyu Liu.

2 Introduction

Due to the enhanced capabilities of laptop computers, people can communicate with each other while moving without being constrained by wires. DARPA packet radio network initiated the idea of forming an on-the-fly ad-hoc network of mobile nodes. Many research works have focused on the algorithmic complexity of choosing the optimal set of ad-hoc routers while others have proposed new routing solutions leveraging features from the existing Internet routing algorithms. As a variant of the distance vector routing method by which mobile nodes cooperate to form an ad-hoc network, the Destination-Sequenced Distance Vector (DSDV) algorithm is effective for creating ad-hoc networks for small populations of mobile nodes, but it is a fairly brute force approach because it depends for its correct operation on the periodic advertisement and global dissemination of connectivity information. The application of DSDV has limitations when frequent system-wide broadcasts limit the size of ad-hoc networks. On the other hand, DSDV also requires each mobile node to maintain a complete list of routes, one for each destination within the ad-hoc network. Keeping a complete routing table does reduce route acquisition latency before transmission of the first packet to a destination. The proposed AODV algorithm can minimize broadcasts and transmission latency when new routes are needed. This algorithm works on wired media as well as wireless media, as long as links along which packets may be transmitted are available. The only requirement placed on the broadcast medium is that neighboring nodes can detect each others' broadcasts. Also AODV only uses the symmetric links between neighboring nodes to build a path from sources to destinations. In other words, AODV does not attempt to follow paths between nodes when one of the nodes cannot hear the other one.

3 The Ad-hoc On-Demand Distance Vector Algorithm

In the so called pure on-demand route acquisition system, nodes do not lie on active paths neither maintain any routing information nor participate in any periodic routing table exchanges.

A node only discovers a communication path to another node if and only if there exists requirements to transfer services between two nodes. The local broadcasts known as hello messages are used by the mobile node to detect other nodes in its neighborhood. The primary objectives of this algorithm are as follows:

- To broadcast discovery packets only when necessary
- To distinguish between local connectivity management and general topology maintenance
- To disseminate information about changes in local connectivity to those neighboring mobile nodes that are likely to need the information

Through the combination of several techniques, AODV can use bandwidth efficiently, respond to the link breakage in active routes quickly and ensure loop-free routing.

3.1 Path Discovery

The Path Discovery process is initiated whenever a source node needs to communicate with another node for which it has no routing information in its table. Both the node sequence number and the broadcast id are maintained in each node. The source node initiates path discovery by broadcasting a route request (RREQ) packet to its neighbors. The RREQ contains the following fields.

$\langle source_addr, source_sequence_#, broadcast_id, dest_addr, dest_sequence_#, hop_cnt \rangle$

Each RREQ is uniquely determined by $\langle source_addr, broadcast_id \rangle$ because *source_addr* is a unique IP-address and *broadcast_id* is incremented whenever the source issues a new RREQ.

If a neighbor satisfies the RREQ then it will send a route reply RREP back to the source. Otherwise, it broadcasts the RREQ to its own neighbors after increasing the *hop_cnt*. A node only rebroadcasts new receiving RREQ and it will drop the redundant RREQ and does not rebroadcast it. If a node cannot satisfy the RREQ, it keeps track of the following information in order to implement the reverse path setup, as well as the forward path setup that will accompany the transmission of the eventual RREP:

- Destination IP address
- Source IP address
- Broadcast id
- Expiration time for reverse path route entry
- Source node's sequence number

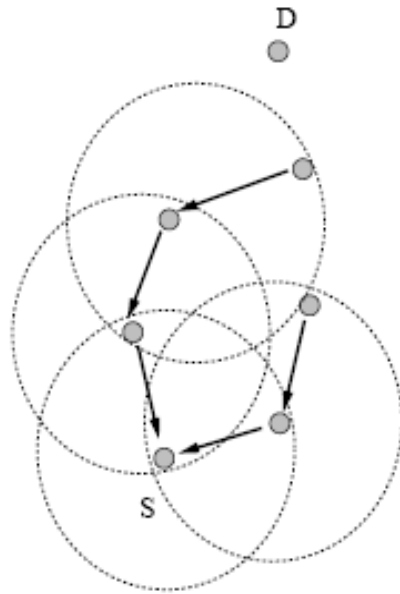


Figure 1. Reverse Path Formation

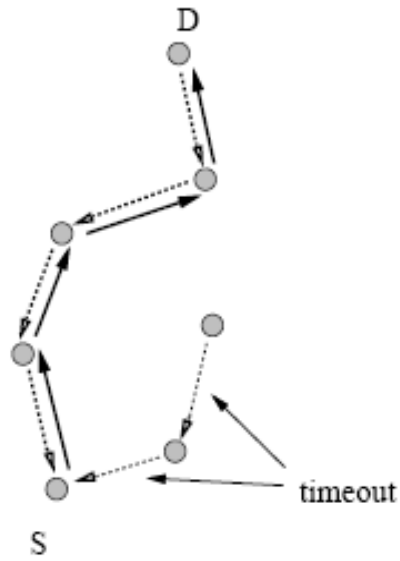


Figure 2. Forward Path Formation

3.1.1 Reverse path Setup

In addition to the *broadcast_id* included in a RREQ, there are two sequence numbers: the source sequence number and the last destination sequence number known to the source.

- The *source sequence number* is used to maintain freshness information about the reverse route to the source
- The *destination sequence number* specifies how fresh a route to the destination must be before it can be accepted by the source

As illustrated in Figure 1, RREQ automatically sets up the reverse path from all nodes back to the source as it travels from a source to various destinations. A node only sets up the reverse path to the node from which it receives the first copy of RREQ.

3.1.2 Forward Path Setup

If an intermediate node has a route entry for the desired destination, it determines whether the route is current by comparing the destination sequence number in its own route entry to the destination sequence number in the RREQ.

- If the RREQ's sequence number for the destination is greater than that recorded by the intermediate node, the intermediate node must not use its recorded route to respond to the RREQ. Instead, the intermediate node rebroadcasts the RREQ.

- If the RREQ's sequence number for the destination is less than or equal to that recorded by the intermediate node, the intermediate node can reply. Furthermore, If the intermediate node does have a current route to the destination and if the RREQ has not been processed previously, the node then unicasts a route reply packet RREP back to its neighbor from which it received the RREQ.

A RREP contains the following information:

$\langle source_addr, dest_addr, dest_sequence_#, hop_cnt, lifetime \rangle$

Illustrated in Figure 2, as the RREP travels back to the source, each node along the path sets up a forward pointer to the node from which the RREP came, updates its timeout information for route entries to the source and destination, and records the latest destination sequence number for the requested destination. A node receiving an RREP propagates the first RREP for a given source node towards that source. If it receives further RREPs, it updates its routing information and propagates the RREP only if the RREP contains either a greater destination sequence number than the previous RREP, or the same destination sequence number with a smaller hopcount. It suppresses all other RREPs it receives. This decreases the number of RREPs propagating towards the source while also ensuring the most up-to-date and quickest routing information. The source node can begin data transmission as soon as the first RREP is received, and can later update its routing information if it learns of a better route.

3.2 Route Table Management

In addition to the source and destination sequence numbers, other useful information is also stored in the route table entries, and is called the soft-state associated with the entry.

- *Route request expiration timer*: the purpose of this timer is to purge reverse path routing entries from those nodes that do not lie on the path from the source to the destination
- *Route caching timeout*: the time after which the route is considered to be invalid
- *Active timeout period*: this information is maintained so that all active source nodes can be noted when a link along a path to the destination breaks

A mobile node maintains a route table entry for each destination of interest. Each route table entry contains the following information:

- Destination
- Next Hop
- Number of hops (metric)
- Sequence number for the destination
- Active neighbors for this route
- Expiration time for the route table entry

3.3 Path Maintenance

Different path maintenance strategies have been applied to different nodes.

- *Nodes not on the active path*: do nothing because these nodes do not affect the routing to that path's destination
- *Source node*: re-initiate the route discovery procedure to establish a new route to the destination
- *Destination/Intermediate nodes*: a special RREP is sent to the affected source nodes

3.4 Local Connectivity Management

Hello messages with a time to live (TTL) value of 1 can enable nodes to learn of their neighbors. The local connectivity management with hello messages can also be used to ensure that only nodes with bidirectional connectivity are considered to be neighbors.

4 Conclusion

AODV avoids problems with previous proposals (notably DSDV) and has the following features:

- Nodes store only the routes that are needed
- Need for broadcast is minimized
- Reduces memory requirements and needless duplications
- Quick response to link breakage in active routes
- Loop-free routes maintained by use of destination sequence numbers
- Scalable to large populations of nodes