16

Routing in Mobile Ad Hoc Networks

Fenglien Lee
University of Guam
Guam 96923,
USA

1. Introduction

A mobile ad hoc network (MANET), sometimes called a mobile mesh network, is a self- configuring network of mobile devices connected by wireless links. In other words, a MANET is a collection of communication nodes that wish to communicate with each other, but has no fixed infrastructure and no predetermined topology of wireless links. Each node in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. Individual nodes are responsible for dynamically discovering other nodes that they can directly communicate with. Due to the limitation of signal transmission range in each node, not all nodes can directly communicate with each other. Each node must forward traffic unrelated to its own use, and therefore be a router.
The primary challenge in building a MANET is equipping each device to continuously maintain the information required to properly route traffic. Therefore, nodes are required to relay packets on behalf of other nodes in order to deliver data across the network. A significant feature of ad hoc networks is that changes in connectivity and link characteristics are introduced due to node mobility and power control practices.
Ad hoc networks can be built around any wireless technology, including infrared, radio frequency (RF), global positioning system (GPS), and so on. Usually, each node is equipped with a transmitter and a receiver to communicate with other nodes [Lee2009] [Wiki2010a].

1.1 Routing in a MANET

The absence of fixed infrastructure in a MANET poses several types of challenges. The biggest challenge among them is routing. Routing is the process of selecting paths in a network along which to send data packets. An ad hoc routing protocol is a convention, or standard, that controls how nodes decide which way to route packets between computing devices in a mobile ad-hoc network.
In ad hoc networks, nodes do not start out familiar with the topology of their networks; instead, they have to discover it. The basic idea is that a new node may announce its presence and should listen for announcements broadcast by its neighbors. Each node learns about nearby nodes and how to reach them, and may announce that it can reach them too. The routing process usually directs forwarding on the basis of routing tables which maintain a record of the routes to various network destinations. Thus, constructing routing tables, which are held in the router's memory, is very important for efficient routing [Lee2009][Wiki2010b].

1.2 Routing protocols for MANET

The growth of laptops and 802.11/Wi-Fi wireless networking has made MANETs a popular research topic since the 1990s. Many academic papers evaluate protocols and abilities assuming varying degrees of mobility within a bounded space, usually with all nodes within a few hops of each other and usually with nodes sending data at a constant rate. Different protocols are then evaluated based on the packet drop rate, average routing load, average end-to-end-delay, and other measures. The proposed solutions for routing protocols could be grouped in three categories: proactive (or table-driven), reactive (or on-demand), and hybrid protocols. Even the reactive protocols have become the main stream for MANET routing. In this chapter, we introduce some popular routing protocols in each of the three categories and for IPv6 networks [Lee2009][Wiki2010a][Wiki2010c].

1.3 Applications for MANET

Ad hoc networks are suited for use in situations where infrastructure is either not available or not trusted, such as a communication network for military soldiers in a field, a mobile network of laptop computers in a conference or campus setting, temporary offices in a campaign headquarters, wireless sensor networks for biological research, mobile social networks such as Facebook, MySpace and Twitter, and mobile mesh networks for Wi-Fi devices [Lee2009].

2. Proactive routing protocols

Every proactive routing protocol usually needs to maintain accurate information in their routing tables. It attempts to continuously evaluate all of the routes within a network. This means the protocol maintains fresh lists of destinations and their routes by periodically distributing routing tables throughout the network. So that when a packet needs to be forwarded, a route is already known and can be used immediately. Once the routing tables are setup, then data (packets) transmissions will be as fast and easy as in the tradition wired networks.

Unfortunately, it is a big overhead to maintain routing tables in the mobile ad hoc network environment. Therefore, the proactive routing protocols have the following common disadvantages:

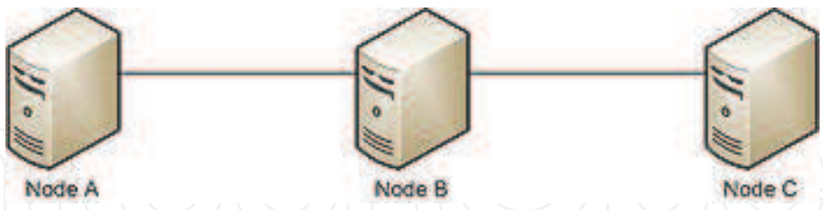
1. Respective amount of data for maintaining routing information.
2. Slow reaction on restructuring network and failures of individual nodes.

Proactive routing protocols became less popular after more and more reactive routing protocols were introduced. In this section, we introduce three popular proactive routing protocols – DSDV, WRP and OLSR. Besides the three popular protocols, there are many other proactive routing protocols for MNAET, such as CGSR, HSR, MMRP and so on [Wiki2010c][Sholander2002].

2.1 Destination-Sequenced Distance Vector (DSDV)

Destination-Sequenced Distance-Vector Routing (DSDV) is a table-driven routing scheme for ad hoc mobile networks based on the Bellman-Ford algorithm. It was developed by C. Perkins and P. Bhagwat in 1994. The main contribution of the algorithm was to solve the routing loop problem. Each entry in the routing table contains a sequence number. If a link presents the sequence numbers are even generally, otherwise an odd number is used. The

number is generated by the destination, and the emitter needs to send out the next update with this number. Routing information is distributed between nodes by sending *full dumps* infrequently and smaller incremental updates more frequently.



For example the routing table of Node A in the above network is

Destination	Next Hop	Number of Hops	Sequence Number	Install Time
A	A	0	A46	001000
B	B	1	B36	001200
C	B	2	C28	001500

Naturally the table contains description of all possible paths reachable by node A, along with the next hop, number of hops, sequence number and install time.

Selection of Route

If a router receives new information, then it uses the latest sequence number. If the sequence number is the same as the one already in the table, the route with the better metric is used. Stale entries are those entries that have not been updated for a while. Such entries as well as the routes using those nodes as next hops are deleted. Then new destination comes. This is how it works.

Influence

Since no formal specification of this algorithm is present, there is no commercial implementation of this algorithm. But some other protocols have used similar techniques. The best-known sequenced distance vector protocol is AODV, which, by virtue of being a reactive protocol, can use simpler sequencing heuristics. Besides, Babel is a distance-vector routing protocol for IPv4 and IPv6 with fast convergence properties. It was designed to make DSDV more robust, more efficient and more widely applicable for both wireless mesh networks and classical wired networks while staying within the framework of proactive protocols [Abohansen2009].

Advantages

DSDV was one of the early algorithms available. It is quite suitable for creating ad hoc networks with small number of nodes.

Disadvantages

DSDV requires a regular update of its routing tables, which uses up battery power and a small amount of bandwidth even when the network is idle. Also, whenever the topology of the network changes, a new sequence number is necessary before the network re-converges; thus, DSDV is not suitable for highly dynamic networks [Wiki2010d][Perkins94].

2.2 Wireless Routing Protocol (WRP)

The Wireless Routing Protocol (WRP) is a proactive unicast routing protocol for MANETs. WRP uses an enhanced version of the distance-vector routing protocol, which uses the Bellman-Ford algorithm to calculate paths. Because of the mobile nature of the nodes within the MANET, the protocol introduces mechanisms which reduce route loops and ensure reliable message exchanges.

The wireless routing protocol (WRP), similar to DSDV, inherits the properties of the distributed Bellman-Ford algorithm. To solve the count-to-infinity problem and to enable faster convergence, it employs a unique method of maintaining information regarding the shortest path to every destination node and the penultimate hop node on the path to every destination node in the network. Since WRP, like DSDV, maintains an up-to-date view of the network, every node has a readily available route to every destination node in the network. It differs from DSDV in table maintenance and in the update procedures. While DSDV maintains only one topology table, WRP uses a set of tables to maintain more accurate information. The tables that are maintained by a node are the following: distance table (DT), routing table (RT), link cost table (LCT), and a message retransmission list (MRL).

Distance Table

The DT contains the network view of the neighbors of a node. It contains a matrix where each element contains the distance and the penultimate node reported by a neighbor for a particular destination.

Routing Table

The RT contains the up-to-date view of the network for all known destinations. It keeps the shortest distance, the predecessor node (penultimate node), the successor node (the next node to reach the destination), and a flag indicating the status of the path. The path status may be a simple path (correct), or a loop (error), or the destination node not marked (null, invalid route). Note, storing the previous and successive nodes assists in detecting loops and avoiding the counting-to-infinity problem - a shortcoming of Distance Vector Routing.

Link Cost Table

The LCT contains the cost (e.g., the number of hops to reach the destination) of relaying messages through each link. The cost of a broken link is infinity. It also contains the number of update periods (intervals between two successive periodic updates) passed since the last successful update was received from that link. This is used to detect link breaks.

The LCT maintains the cost of the link to its nearest neighbors (nodes within direct transmission range), and the number of timeouts since successfully receiving a message from the neighbor. Nodes periodically exchange routing tables with their neighbors via update messages, or whenever the link cost table changes.

Message Retransmission List

The MRL contains an entry for every update message that is to be retransmitted and maintains a counter for each entry. This counter is decremented after every retransmission of an update message. Each update message contains a list of updates. A node also marks each node in the RT that has to acknowledge the update message it transmitted. Once the counter reaches zero, the entries in the update message for which no acknowledgments have been received are to be retransmitted and the update message is deleted. Thus, a node

detects a link break by the number of update periods missed since the last successful transmission. After receiving an update message, a node not only updates the distance for transmission neighbors but also checks the other neighbors' distance, hence convergence is much faster than DSDV. The MRL maintains a list of which neighbors are yet to acknowledge an update message, so they can be retransmitted if necessary. If there is no change in the routing table, a node is required to transmit a "hello" message to affirm its connectivity. When an update message is received, a node updates its distance table and reassesses the best route paths. It also carries out a consistency check with its neighbors, to help eliminate loops and speed up convergence.

Advantages

WRP has the same advantage as that of DSDV. In addition, it has faster convergence and involves fewer table updates.

Disadvantages

The complexity of maintenance of multiple tables demands a larger memory and greater processing power from nodes in the wireless ad hoc network. At high mobility, the control overhead involved in updating table entries is almost the same as that of DSDV and hence is not suitable for a highly dynamic and for a very large ad hoc wireless network as it suffers from limited scalability [Wiki2010e][Murthy1996].

2.3 Optimized Link State Routing (OLSR)

The Optimized Link State Routing Protocol (OLSR) is an IP routing protocol optimized for mobile ad-hoc networks, which can also be used on other wireless ad-hoc networks. OLSR is a proactive link-state routing protocol, which uses Hello and Topology Control (TC) messages to discover and then disseminate link state information throughout the mobile ad-hoc network. Individual nodes use this topology information to compute next hop destinations for all nodes in the network using shortest hop forwarding paths.

Features Specific to OLSR

Link-state routing protocols such as OSPF and IS-IS elect a *designated router* on every link to perform flooding of topology information. In wireless ad-hoc networks, there is different notion of a link, packets can go out the same interface; hence, a different approach is needed in order to optimize the flooding process. Using Hello messages the OLSR protocol at each node discovers 2-hop neighbor information and performs a distributed election of a set of *multipoint relays* (MPRs). Nodes select MPRs such that there is a path to each of its 2-hop neighbors via a node selected as an MPR. These MPR nodes then forward TC messages that contain the MPR selectors. This functioning of MPRs makes OLSR unique from other link state routing protocols in a few different ways: The forwarding path for TC messages is not shared among all nodes but varies depending on the source, only a subset of nodes source link state information, not all links of a node are advertised but only those that represent MPR selections.

Since link-state routing requires the topology database to be synchronized across the network, OSPF (Open Shortest Path First) and IS-IS (Intermediate System to Intermediate System) perform topology flooding using a reliable algorithm. Such an algorithm is very difficult to design for ad-hoc wireless networks, so OLSR doesn't bother with reliability; it simply floods topology data often enough to make sure that the database does not remain unsynchronized for extended periods of time.

Messages Used in OLSR

OLSR uses the "Hello" messages to find its one hop neighbors and its two hop neighbors through their responses. The sender can then select its multipoint relays (MPR) OLSR uses the "Hello" messages to find its one hop neighbors and its two hop neighbors through their responses. The sender can then select its multipoint relays (MPR) based on the one hop node that offers the best routes to the two hop nodes. Each node has also an MPR selector set, which enumerates nodes that have selected it as an MPR node. OLSR uses Topology Control (TC) messages along with MPR forwarding to disseminate neighbor information throughout the network. Host and Network Association (HNA) messages are used by OLSR to disseminate network route advertisements in the same way TC messages advertise host routes. Below are the formats of Topology and Hello Control messages.

1. Topology Control Message

0 (bits 0-9)				1 (bits 10-19)										2 (bits 20-29)						3	
0	1	…….	9	0	1	2	3	4	5	6	7	8	9	0	1	…….	9	0	1		
ANSN										Reserved											
Advertised Neighbor Main Address																					
Advertised Neighbor Main Address																					

Note: Each row has 32 bits.

2. Hello Control Message

0 (bits 0-9)				1 (bits 10-19)										2 (bits 20-29)							3	
0	1	…….	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	….	9	0	1
Reserved										Htime								Willingness				
Link Code				Reserved						Link Message Size												
Neighbor Interface Address																						
Neighbor Interface Address																						
…. ….																						
Link Code				Reserved						Link Message Size												
Neighbor Interface Address																						
Neighbor Interface Address																						

Advantages

Being a proactive protocol, routes to all destinations within the network are known and maintained before use. Having the routes available within the standard routing table can be useful for some systems and network applications as there is no route discovery delay associated with finding a new route. The routing overhead generated, while generally greater than that of a reactive protocol, does not increase with the number of routes being used.

Default and network routes can be injected into the system by HNA (Host and Network Association) messages allowing for connection to the internet or other networks within the OLSR MANET cloud. Network routes using reactive protocols do not currently execute well. Timeout values and validity information is contained within the messages conveying information allowing for differing timer values to be used at differing nodes.

Disadvantages

The original definition of OLSR does not include any provisions for sensing of link quality; it simply assumes that a link is up if a number of hello packets have been received recently. This assumes that links are bi-modal (either working or failed), which is not necessarily the case on wireless networks, where links often exhibit intermediate rates of packet loss.

Implementations such as the open source OLSRD (OLSR Daemon, commonly used on Linux-based mesh routers) have been extended (as of v. 0.4.8) with link quality sensing. Being a proactive protocol, OLSR uses power and network resources in order to propagate data about possibly unused routes. While this is not a problem for wired access points, and laptops, it makes OLSR unsuitable for sensor networks that try to sleep most of the time. For small scale wired access points with low CPU power, the open source OLSRD project showed that large scale mesh networks can run with OLSRD on thousands of nodes with very little CPU power on 200 MHz embedded devices.

Being a link-state protocol, OLSR requires a reasonably large amount of bandwidth and CPU power to compute optimal paths in the network. In the typical networks where OLSR is used (which rarely exceed a few hundreds of nodes), this does not appear to be a problem. By only using MPRs to flood topology information, OLSR removes some of the redundancy of the flooding process, which may be a problem in networks with moderate to large packet loss rates - however the MPR mechanism is self-pruning (which means that in case of packet losses, some nodes that would not have retransmitted a packet may do so).

OLSR Version 2

OLSRv2 is currently being developed within the IETF. It maintains many of the key features of the original including MPR selection and dissemination. Key differences are the flexibility and modular design using shared components: packet format, and neighborhood discovery protocol (NHDP). These components are being designed to be common among next generation IETF MANET protocols. Differences in the handling of multiple address and interface enabled nodes is also present between OLSR and OLSRv2 [Abohansen2009] [Wiki2010f][Clausen2003].

3. Reactive routing protocols

In bandwidth-starved and power-starved environments, it is interesting to keep the network silent when there is no traffic to be routed. Reactive routing protocols do not maintain routes, but build them on demand. A reactive protocol finds a route on demand by flooding the network with Route Request packets. These protocols have the following advantages:

1. No big overhead for global routing table maintenance as in proactive protocols.
2. Quick reaction for network restructure and node failure.

Even reactive protocols have become the main stream for MANET routing, they still have the following main disadvantages:

1. High latency time in route finding.
2. Excessive flooding can lead to network clogging.

There are many reactive routing protocols for MANET. We only introduce three popular (AODV, DSR and DYMO) and one new (ODCR) protocols in this section [Wiki2010c].

3.1 Ad hoc On-demand Distance Vector (AODV)

Ad hoc On-Demand Distance Vector (AODV) Routing is a routing protocol for mobile ad hoc networks (MANETs) and other wireless ad-hoc networks. It is jointly developed in Nokia Research Center, University of California, Santa Barbara and University of Cincinnati by C. Perkins, E. Belding-Royer and S. Das. AODV is capable of both unicast and multicast routing. It is a reactive routing protocol, meaning that it establishes a route to a destination only on demand. In contrast, the most common routing protocols of the Internet are proactive, meaning they find routing paths independently of the usage of the paths. AODV is, as the name indicates, a distance-vector routing protocol. AODV avoids the *counting-to-infinity* problem of other distance-vector protocols by using sequence numbers on route updates, a technique pioneered by DSDV.

In AODV, the network is silent until a connection is needed. At that point the network node that needs a connection broadcasts a request for connection. Other AODV nodes forward this message, and record the node that they heard it from, creating an explosion of temporary routes back to the needy node. When a node receives such a message and already has a route to the desired node, it sends a message backwards through a temporary route to the requesting node. The needy node then begins using the route that has the least number of hops through other nodes. Unused entries in the routing tables are recycled after a time. When a link fails, a routing error is passed back to a transmitting node, and the process repeats.

Much of the complexity of the protocol is to lower the number of messages to conserve the capacity of the network. For example, each request for a route has a sequence number. Nodes use this sequence number so that they do not repeat route requests that they have already passed on. Another such feature is that the route requests have a "time to live" number that limits how many times they can be retransmitted. The third feature is that if a route request fails, another route request may not be sent until twice as much time has passed as the timeout of the previous route request.

Technical Description

The AODV Routing protocol uses an on-demand approach for finding routes, that is, a route is established only when it is required by a source node for transmitting data packets. It employs destination sequence numbers to identify the most recent path. The major difference between AODV and Dynamic Source Routing (DSR) is that DSR uses source routing in which a data packet carries the complete path to be traversed; however, in AODV, the source node and the intermediate nodes store the next-hop information corresponding to each flow for data packet transmission.

In an on-demand routing protocol, the source node floods the *RouteRequest* packet in the network when a route is not available for the desired destination. It may obtain multiple routes to different destinations from a single *RouteRequest*. The major difference between AODV and other on-demand routing protocols is that it uses a *destination sequence number* (DestSeqNum) to determine an up-to- date path to the destination. A node updates its path information only if the *DestSeqNum* of the current packet received is greater than the last *DestSeqNum* stored at the node.

A RouteRequest carries the source identifier (SrcID), the destination identifier (DestID), the source sequence number (SrcSeqNum), the destination sequence number (DestSeqNum), the broadcast identifier (BcastID), and the *time to live* (TTL) field. DestSeqNum indicates the

freshness of the route that is accepted by the source. When an intermediate node receives a RouteRequest, it either forwards it or prepares a RouteReply if it has a valid route to the destination. The validity of a route at the intermediate node is determined by comparing the sequence number at the intermediate node with the destination sequence number in the RouteRequest packet. If a RouteRequest is received multiple times, which is indicated by the BcastID-SrcID pair, the duplicate copies are discarded. All intermediate nodes having valid routes to the destination, or the destination node itself, are allowed to send RouteReply packets to the source.

Every intermediate node, while forwarding a RouteRequest, enters the previous node address and its BcastID. A timer is used to delete this entry in case a RouteReply is not received before the timer expires. This helps in storing an active path at the intermediate node as AODV does not employ source routing of data packets. When a node receives a RouteReply packet, information about the previous node from which the packet was received is also stored in order to forward the data packet to this next node as the next hop toward the destination.

Advantages

The main advantage of this protocol is that routes are established on demand and destination sequence numbers are used to find the latest route to the destination. The connection setup delay is lower. It creates no extra traffic for communication along existing links. Also, distance vector routing is simple, and doesn't require much memory or calculation.

Disadvantages

AODV requires more time to establish a connection, and the initial communication to establish a route is heavier than some other approaches. Also, intermediate nodes can lead to inconsistent routes if the source sequence number is very old and the intermediate nodes have a higher but not the latest destination sequence number, thereby having stale entries. Also multiple RouteReply packets in response to a single RouteRequest packet can lead to heavy control overhead. Another disadvantage of AODV is that the periodic beaconing leads to unnecessary bandwidth consumption [Wiki2010g] [Perkins2003].

3.2 Dynamic Source Routing

Dynamic Source Routing (DSR) is a routing protocol for wireless mesh networks. It is similar to AODV in that it forms a route on-demand when a transmitting computer requests one. However, it uses source routing instead of relying on the routing table at each intermediate device. Many successive refinements have been made to DSR, including DSRFLOW.

Determining source routes requires accumulating the address of each device between the source and destination during route discovery. The accumulated path information is cached by nodes processing the route discovery packets. The learned paths are used to route packets. To accomplish source routing, the routed packets contain the address of each device the packet will traverse. This may result in high overhead for long paths or large addresses, like IPv6 (Internet Protocol version 6). To avoid using source routing, DSR optionally defines a flow id option that allows packets to be forwarded on a hop-by-hop basis.

This protocol is truly based on source routing whereby all the routing information is maintained (continually updated) at mobile nodes. It has only two major phases which are Route Discovery and Route Maintenance. Route Reply would only be generated if the message has reached the intended destination node (route record which is initially contained in Route Request would be inserted into the Route Reply).

To return the Route Reply, the destination node must have a route to the source node. If the route is in the Destination Node's route cache, the route would be used. Otherwise, the node will reverse the route based on the route record in the Route Reply message header (this requires that all links are symmetric). In the event of fatal transmission, the Route Maintenance Phase is initiated whereby the Route Error packets are generated at a node.

The erroneous hop will be removed from the node's route cache, all routes containing the hop are truncated at that point. Again, the Route Discovery Phase is initiated to determine the most viable route.

Dynamic source routing protocol (DSR) is an on-demand protocol designed to restrict the bandwidth consumed by control packets in ad hoc wireless networks by eliminating the periodic table-update messages required in the table-driven approach. The major difference between this and the other on-demand routing protocols is that it is beacon-less and hence does not require periodic hello packet (beacon) transmissions, which are used by a node to inform its neighbors of its presence. The basic approach of this protocol (and all other on-demand routing protocols) during the route construction phase is to establish a route by flooding RouteRequest packets in the network. The destination node, on receiving a RouteRequest packet, responds by sending a RouteReply packet back to the source, which carries the route traversed by the RouteRequest packet received.

Consider a source node that does not have a route to the destination. When it has data packets to be sent to that destination, it initiates a RouteRequest packet. This RouteRequest is flooded throughout the network. Each node, upon receiving a RouteRequest packet, rebroadcasts the packet to its neighbors if it has not forwarded it already, provided that the node is not the destination node and that the packet's *time to live* (TTL) counter has not been exceeded. Each RouteRequest carries a sequence number generated by the source node and the path it has traversed.

A node, upon receiving a RouteRequest packet, checks the sequence number on the packet before forwarding it. The packet is forwarded only if it is not a duplicate RouteRequest. The sequence number on the packet is used to prevent loop formations and to avoid multiple transmissions of the same RouteRequest by an intermediate node that receives it through multiple paths. Thus, all nodes except the destination forward a RouteRequest packet during the route construction phase. A destination node, after receiving the first RouteRequest packet, replies to the source node through the reverse path the RouteRequest packet had traversed.

Nodes can also learn about the neighboring routes traversed by data packets if operated in the promiscuous mode (the mode of operation in which a node can receive the packets that are neither broadcast nor addressed to itself). This route cache is also used during the route construction phase. If an intermediate node receiving a RouteRequest with a route to the destination node in its route cache, then it replies to the source node by sending a RouteReply with the entire route information from the source node to the destination node.

Advantages

This protocol uses a reactive approach which eliminates the need to periodically flood the network with table update messages which are required in a table-driven approach. In a reactive (on-demand) approach such as this, a route is established only when it is required and hence the need to find routes to all other nodes in the network as required by the table-driven approach is eliminated. The intermediate nodes also utilize the route cache information efficiently to reduce the control overhead.

Disadvantages

The disadvantage of this protocol is that the route maintenance mechanism does not locally repair a broken link. Stale route cache information could also result in inconsistencies during the route reconstruction phase. The connection setup delay is higher than in table-driven protocols. Even though the protocol performs well in static and low-mobility environments, the performance degrades rapidly with increasing mobility. Also, considerable routing overhead is involved due to the source-routing mechanism employed in DSR. This routing overhead is directly proportional to the path length [Wiki2010h][Au-Yong2006][Johnson1994][Johnson2001].

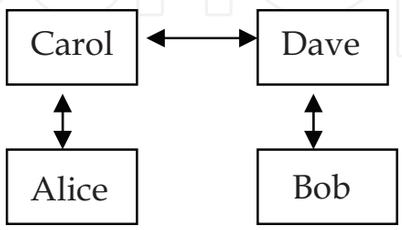
3.3 Dynamic MANET On-Demand Routing (DYMO)

The DYMO routing protocol is a successor to the popular Ad hoc On-Demand Distance Vector (AODV) routing protocol and shares many of its benefits. It is, however, slightly easier to implement and designed with future enhancements in mind. DYMO can work as both a proactive and as a reactive routing protocol, i.e. routes can be discovered just when they are needed. In any way, to discover new routes the following two steps take place:

1. A special "Route Request" (RREQ) messages is broadcast through the MANET. Each RREQ keeps an ordered list of all nodes it passed through, so every host receiving an RREQ message can immediately record a route back to the origin of this message.
2. When an RREQ message arrives at its destination, a "Routing Reply" (RREP) message will immediately get passed back to the origin, indicating that a route to the destination was found. On its way back to the source, an RREP message can simply backtrace the way the RREQ message took and simultaneously allow all hosts it passes to record a complementary route back to where it came from.

So as soon as the RREP message reaches its destination, a two-way route was successfully recorded by all intermediate hosts, and exchange of data packets can commence.

Example



Step 1.

- Alice wants to exchange data with Bob
- Alice does not know a route to Bob yet, so it broadcasts a new RREQ for a route to Bob containing only information about itself.

Step 2.
- Carol receives Alice's RREQ, remembers the contained information about how to reach Alice (directly), then appends information about itself and re-broadcasts the packets.

Step 3.
- Dave receives Carol's RREQ, remembers the contained information about how to reach Carol (directly) and Alice (via Carol), then appends information about itself and re-broadcasts the packet.
- At the same time, Alice also receives Carol's RREQ. Closer examination of the contained information reveals that even the very first information block - how to reach itself, Alice - is of no use. It thus discards the RREQ and does not re- broadcast it as Dave did.

Step 4.
- Bob receives Dave's RREQ and remembers the contained information about how to reach Dave (directly), Carol (via Dave) and Alice (also via Dave). Realizing that he is the target of the RREQ he creates an RREP containing information about itself. He marks the RREP bound for Alice and - knowing that Dave can somehow reach Alice - sends it to Dave.
- Again, at the same time, Carol also receives Dave's RREQ, but - following the same logic as Alice before - ignores it.

Step 5.
- Dave receives the RREP to Alice sent by Bob, remembers the information on how to reach Bob (directly), appends information about itself and - knowing that Alice can be reached via Carol, sends it to Carol.

Step 6.
- Carol receives the RREP to Alice sent by Dave, remembers the contained information on how to reach Dave (directly) and Bob (via Dave), then appends information about itself and - knowing that Alice can be reached directly, sends it to Alice.

Step 7.
- Alice receives the RREP sent to her by Carol and remembers all information on how to reach Carol (directly), Dave (via Carol) and - most importantly - Bob (also via Carol). Now knowing how to reach Bob she can finally send her data packet for him to Carol.

Step 8.
- Carol receives the data packet for Bob from Alice. Because she knows Dave can reach Bob she forwards the packet to him.

Step 9.
- Dave receives the data packet for Bob. Because he knows Bob can be directly reached by him, he forwards the packet to him.

Step 10.
- Bob receives the data packet. Still knowing how to reach Alice, he could now respond with one of his own, and the process repeats until communications are complete or the network changes (e.g. Carol leaves or Eileen joins), where it may be necessary to search the network again for a route [Wiki2010i] [Chakeres2008].

3.4 On-Demand Cache Routing protocol

This protocol presents an efficient algorithm for route discovery, route management and mobility handling for on-demand routing. It is called as “on-demand cache routing” (ODCR) algorithm since it applies caches in each node to improve the routing performance.

In the MANET, each node equips L-1 (level 1 or primary) and L-2 (level 2 or secondary) caches. Usually, the size of L-1 cache is about 64 to 256 KB and L-2 cache is about 256 KB to 2MB). For memory address mapping, they use 2-, 4- or 8-way set associative scheme. Each data entry in a cache is called a "cache line". Most caches use the least-recently-used (LRU) policy for cache line replacement. All cache lines can be searched in parallel in a few processor cycles. This is an important reason why many routing protocols adopted cache for route management. This cache is called as "route cache" because it stores the routing information in the network.

For the initial settings of a MANET, this protocol assumes (1) the communication media among nodes (e.g. laptop computers) is RF; (2) each node has an identification (ID) number; (3) each node keeps an ID list in its own cache (see Figure 1); (4) the wireless links in the network are symmetric (i.e. bi-directional transmission); and (5) the network is scalable and heterogeneous. This means the number of nodes in the network is changeable anytime, and the processor architecture, transmission radius and battery life of each node can be different. In this section, we only present the main algorithm (ODCR). For detail operations of sub-algorithms RDA and MHA mentioned in Algorithm ODCR below, please refer to [Lee2009].

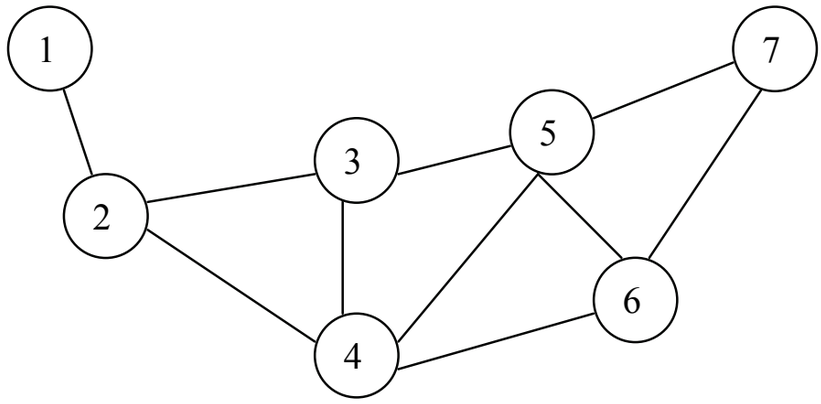


Fig. 1. A simple MANET, where 1, 2, 3, 4, 5, 6 and 7 are node IDs and solid edges are wireless links within the RF transmission radius of each node. For example, node 5 can transmits packets to nodes 3, 4, 6 and 7. In this MANET, each node has an ID list (1, 2, 3, 4, 5, 6, 7).

Algorithm: On-Demand Cache Routing (ODCR)
Inputs: Node identifications (IDs) in the MANET.
Outputs: Transmitted data packets on the network.
Begin

1. If any node in the network wants to send a data packet, at first it has to search the best route (usually the least hop-count route) from its cache. If the route does not exist, go to Step 2. Otherwise (i.e. the route exists) go to Step 3.
2. The source node looks up the destination node in its ID list (as in Figure 1). Then it executes the Route Discovery Algorithm (RDA) to create the best route to its destination node in the network. For instance, the best route from node 1 to node 6 is {1,2,4,6}.
3. The source node attaches its ID, destination node ID and the packet number to eachdata packet, and sends the packet to the destination node along the best route.
4. Each intermediate node uses the best route to the destination node in its cache toforward the data packet to the next or destination node.

5. If any node leaves from, joins to, or moves around the network, it has to execute the Mobility Handling Algorithm (MHA) to notify other nodes about this change and to update their own route information in their caches.
6. Repeat Steps 1 to 5 until the whole network is terminated.

End of On-Demand Cache Routing.

In conclusion, this protocol proposed an efficient on-demand routing algorithm, called ODCR, for route discovery and management, and mobility handling. The ODCR algorithm applied the content-addressable and LRU replacement features in L-1 and L-2 caches for route table creation, updating, and maintenance. The ODCR algorithm with duel-level route caches solved most problems in on-demand routing, such as route tables in "slow" main memory, long connection setup delay, broken link repairing, huge routing overhead for long routes, lengthy data packet in source routing, sending beacons ("hello packets") periodically, control overhead for complicated IDs in data packets, to setup TTL (time-to-live) in a packet or a route path, and to update the stale routes in the route table or cache frequently.

The simulation results showed that the ODCR algorithm outperforms AODV, DSR (Dynamic Source Routing) and CSOR (Cache Scheme in On-Demand Routing) in packet delivery rate, average end-to-end delay and average routing load [Lee2009].

4. Hybrid routing protocols

This type of protocols combines the advantages of proactive and reactive routings. The routing is initially established with some proactively prospected routes and then serves the demand from additionally activated nodes through reactive flooding. The choice for one or the other method requires predetermination for typical cases. The main disadvantages of such algorithms are:

1. Advantage depends on amount of nodes activated.
2. Reaction to traffic demand depends on gradient of traffic volume [Wiki2010j].

4.1 Zone Routing Protocol

Zone Routing Protocol (ZRP) was the first hybrid routing protocol with both a proactive and a reactive routing component. ZRP was first introduced by Haas in 1997. ZRP is proposed to reduce the control overhead of proactive routing protocols and decrease the latency caused by routing discover in reactive routing protocols. ZRP defines a zone around each node consisting of its k-neighborhood (e.g. k=3). That is, in ZRP, all nodes within k-hop distance from node belong to the routing zone of node.

ZRP is formed by two sub-protocols, a proactive routing protocol: Intra-zone Routing Protocol (IARP), is used inside routing zones and a reactive routing protocol: Inter-zone Routing Protocol (IERP), is used between routing zones, respectively. A route to a destination within the local zone can be established from the proactively cached routing table of the source by IARP. Therefore, if the source and destination is in the same zone, the packet can be delivered immediately. Most of the existing proactive routing algorithms can be used as the IARP for ZRP.

For routes beyond the local zone, route discovery happens reactively. The source node sends a route requests to its border nodes, containing its own address, the destination

address and a unique sequence number. Border nodes are nodes which are exactly the maximum number of hops to the defined local zone away from the source. The border nodes check their local zone for the destination. If the requested node is not a member of this local zone, the node adds its own address to the route request packet and forwards the packet to its border nodes. If the destination is a member of the local zone of the node, it sends a route reply on the reverse path back to the source. The source node uses the path saved in the route reply packet to send data packets to the destination [Wiki2010k] [Haas2002].

4.2 Order One Network Protocol

The Order One MANET Routing Protocol (OORP) is an algorithm for computer communicating by digital radio in a mesh network to find each other, and send messages to each other along a reasonably efficient path. It was designed for, and promoted as working with wireless mesh networks. OORP can handle hundreds of nodes, where most other protocols handle less than a hundred. OORP uses hierarchical algorithms to minimize the total amount of transmissions needed for routing. Routing overhead is only about 1% to 5% of node to node bandwidth in any network and does not grow as the network size grows.

The basic idea is that a network organizes itself into a tree. Nodes meet at the root of the tree to establish an initial route. The route then moves away from the root by cutting corners, as ant-trails do. When there are no more corners to cut, a nearly optimum route exists. This route is continuously maintained. Each process can be performed with localized minimal communication, and very small router tables. OORP requires about 200K of memory. A simulated network with 500 nodes transmitting at 200 bytes/second organized itself in about 20 seconds. As of 2004, OORP was patented or had other significant intellectual property restrictions.

Assumptions

Each computer or "node" of the network has a unique name. At least one network link and a computer with some capacity hold a list of neighbors.

Organizing a Tree

The network nodes form a hierarchy by having each node select a parent. The parent is a neighbor node that is the next best step to the most other nodes. This method creates a hierarchy around nodes that are more likely to be present, and which have more capacity, and which are closer to the topological center of the network. The memory limitations of a small node are reflected in its small routing table, which automatically prevents it from being a preferred central node. At the top, one or two nodes are unable to find nodes better-connected than themselves, and therefore become parents of the entire network. The hierarchy-formation algorithm does not need a complex routing algorithm or large amounts of communication.

Routing

All nodes push a route to themselves to the root of the tree. A node wanting a connection can therefore push a request to the root of the tree, and always find a route. The commercial protocol uses Dijkstra's algorithm to continuously optimize and maintain the route. As the network moves and changes, the path is continually adjusted.

Advantages

Assuming that some nodes in the network have enough memory to know of all nodes in the network, there is no practical limitation to network size. Since the control bandwidth is defined to be less than 5% regardless of network size, the amount of control bandwidth required is not supposed to increase as network size grows. The system can use nodes with small amounts of memory.

The network has a reliable, low-overhead way to establish that a node is not in the network. This is a valuable property in ad-hoc mesh networks. Most routing protocols scale either by reducing proactive link-state routing information or reactively driving routing by connection requests. OORP mixes the proactive and reactive methods. Properly configured, an OORP net can theoretically scale to 100,000's of nodes and can often achieve reasonable performance even though it limits routing bandwidth to 5%.

Disadvantages

Central nodes have an extra burden because they need to have enough memory to store information about all nodes in the network. At some number of nodes, the network will therefore cease to scale. If all the nodes in the network are low capacity nodes the network may be overwhelmed with change. This may limit the maximum scale. However, in real world networks, the farther away from the edge nodes the more the bandwidth grows.

These critiques may have no practical effect. For example, consider a low bandwidth 9.6Kbit/second radio. If the protocol was configured to send one packet of 180 bytes every 5 seconds, it would consume 3% of overall network bandwidth. This one packet can contain up to 80 route updates. Thus even in very low bandwidth network the protocol is still able to spread a lot of route information. Given a 10Mbit connection, 3% of the bandwidth is 4,100 to 16,000 route updates per second. Since the protocol only sends route updates for changes, it is rarely overwhelmed.

The other disadvantage is that public proposals for OORP do not include security or authentication. Security and authentication may provided by the integrator of the protocol. Typical security measures include encryption or signing the protocol packets and incrementing counters to prevent replay attacks [Wiki2010l][Orderone2010].

4.3 Global On-Demand Routing protocol

The Global On-Demand Routing (GOR) is a clever hybrid routing protocol for the MANET. To simplify simulations in GOR, it assumes (1) all nodes are homogeneous; (2) the transmission range of each node is k; and (3) each node has an ID and a pair of positive x and y coordinates to represent its location in the network. The main algorithm for the GOR protocol is described below. For detail operations of sub-algorithms DFA and NRA in GOR protocol, please refer to [Lee2007].

Algorithm GOR Protocol

Inputs: The ID and (x, y) coordinates of each node.

Outputs: Destination nodes receive data packets from sources nodes.

Begin

1. Select a center or near-center node in the initial network as the root node (RN).
2. The RN runs the Double-Flooding Algorithm (DFA) to create the location table (LT), sorts the LT by IDs in ascending order, and broadcasts the LT to each node in the network.

3. Each node uses the LT to generate its own distance table (DT) concurrently. Then, each node marks any distance that is longer than the transmission range k in the DT as "∞" (infinity).
4. Each node calls the Dijkstra's Algorithm to generate the one-to-all shortest-path table (SPT) concurrently (see Figure 2 below).
5. If a new node joined to the network, an existing node moved out of the transmission range of its any neighbor nodes, or an existing node left from the network, then it calls the Node-Reorganization Algorithm (NRA) to ask other nodes to update (or mark as "new" nodes if any) their own LT for these changes consequently.
6. If any node wants to send packets via or to the above joined or moved nodes, it has to (1) use the updated LT in Step 5 to update its DT (or mark the "∞" distances if any); (2) run the Dijkstra's algorithm again to update its SPT; (3) reset all nodes in the LT to "old" nodes; and (4) follows the paths in the new SPT to send packets to its destination nodes.
7. If network topology changed again, repeat steps 5 and 6 until the whole network dismissed.

End of GOR Protocol.

Figure 2 below shows some shortest paths within the transmission range k for node 1. In this figure, the shortest path between nodes 1 and 6 is (1, 3, 6) not (1, 6) because node 6 locates outside the circular transmission range k of node 1. Note we have marked all "∞" distances in steps 3 and 6 respectively in the main algorithm (Algorithm GOR Protocol).

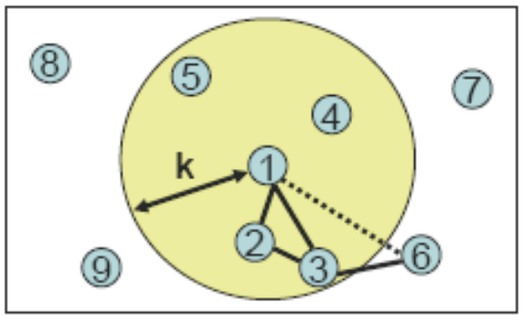


Fig. 2. Sample shortest paths in a MANET.

This algorithm proposed a hybrid global on-demand routing (called GOR) protocol for mobile ad hoc networks. This protocol does not update the routing tables immediately if any node changed its status in the network, such as movement, addition or deletion. Instead, it only handles a node whose move changed the MANET topology or whose move distance is greater than the transmission range k. This critical strategy prevents other nodes from updating the routing tables frequently and hence reducing unnecessary computation and node-reorganization overheads dramatically.

The GOR protocol not only keeps the advantages of proactive and reactive protocols, but also improves the sub-optimal routing overhead and memory consuming problems in local hybrid protocols. Because this protocol retains high packet delivery rate and low end-to-end delay as the DSDV and WRP protocols, and low routing load as the AODV and DSR protocols [Lee2007].

5. MANET routing protocols for IPv6

It is possible that all the IP version 4 (IPv4) addresses will be allocated in next decade. The transition from IP version 4 to IP version 6 (IPv6) will become an important issue in computer networks and Internet in recent years. Therefore, in this section, we introduce IPv6, mobile IPv6, and two popular MANET routing protocols, OLSR and AODV, for IPv6 networks.

5.1 Introduction to IPv6 and mobile IPv6

Internet is built upon a protocol suite called TCP/IP. This abbreviation stands for Transmission Control Protocol, and Internet Protocol. When your computer communicates with the Internet a unique IP address is used to transfer and receive information. Yesterdays IP standard is called IPv4. Each IPv4 address contains 32 binary bits. That is the total address in IPv4 is 2^32 only. Sadly most ISPs and services still only deliver this ancient technology standardized in September 1981. So far, most of IPv4 addresses are already tied up and the Internet is simply running out of IPs. The address shortage problem is aggravated by the fact that portions of the IP address space have not been efficiently allocated.

IPv6 (Internet Protocol version 6) gives citizens the opportunity to become real Internet participants. IPv4 makes citizens into passive consumers who are only able to connect to compartmentalized networks run by companies or governments. This is why the establishment does not want IPv6. Each IPv6 address contains 128 binary bits. This means there are 2^128 unique addresses in IPv6. This huge amount of IP addresses may be able to serve the Internet till the end of this century [Linux2010a].

Mobile IPv6 is the implementation of the IP mobility (Mobile IP) methods and protocols on an Internet Protocol version 6 (IPv6) network. Like its IPv4 counterpart, it is designed to permit IP devices to roam between different networks without losing IP connectivity by maintaining a permanent Internet Protocol (IP) address. Mobile IPv6 is described in RFC3775.

The key benefit of Mobile IPv6 is that even though the mobile node changes locations and addresses, the existing connections through which the mobile node is communicating are maintained. To accomplish this, connections to mobile nodes are made with a specific address that is always assigned to the mobile node, and through which the mobile node is always reachable. Mobile IPv6 provides Transport layer connection survivability when a node moves from one link to another by performing address maintenance for mobile nodes at the Internet layer [Wiki2010m].

5.2 OLSR for IPv6 networks

In this section, we summarize the proposed issues and necessary changes to adapt OLSR to IPv6 from the paper "OLSR for IPv6 Networks" by Laouiti, etc [Laouiti2004]. In order to present a complete IPv6 solution for OLSR, there are several issues to address:

1. Addressing: IPv6 introduce several changes, some more conceptual than others. Changes include the diffusion of data packets and existing multiple addresses of Interfaces.

2. Protocol changes: The OLSR specification gives the protocol format message for IPv4 packets, but some additional changes are proposed.
3. Neighbor discovery: It is described how the neighbor discovery mechanism of IPv6 still operates properly.
4. Autoconfiguration: It is loosely related to addressing, the ability for an IPv6 node to self-configure its addressed yields numerous challenges and had been the subject of elaborate research as seen previously.

IPv6 Ad Hoc Addressing Issues

Several changes are required due to various novelties introduced by IPv6 itself.

1. Interface Addresses: The chosen solution in this paper is to consider a MANET as a single site-local network, and to use site-local prefix with a fixed 16 bits subnet called OLSR_SUBNET. Then, an OLSR node will perform link-local address autoconfiguration, and upon success, will automatically configure for each of its OLSR interfaces. The site-local address with that subnet (FEC0:0:0:OLSR_ SUBNET::/64) will run the OLSR protocol using it.
2. OSLR Diffusion Addresses: In order to reach all the nodes present on the link to get the same effect as in IPv4, this paper proposed that a multicast address ALL_OLSR_NODES is used for the destination address. The ALL_OLSR_NODES could be taken as ALL_LINK_NODES (FF01::1). Also since a node has several interface addresses, the paper proposed that the site-local addresses are used as source addresses.

Diffusing Non-OLSR Packets

Since MANETs are multi-hop routing networks, in order to flood packets to all nodes, retransmissions are usually needed. With OLSR, packets are retransmitted hop by hop to the direct neighborhood by using MPRs (multipoint relays). In the other hand, for any applications, a direct multicast on the local "link" is performed and such packets are never routed. For instance, it is also in the case for most of IPv6 messages for neighbor discovery and autoconfiguration. This relies on the assumption that being on the same network is equivalent to being on same link, an assumption which doesn't hold in MANET networks. As a result, in a multi-hop network, by default, this kind of messages will not be delivered to all nodes. This paper proposed two solutions to diffuse non-OLSR packets to all nodes:

1. Encapsulate the packets in specific OLSR messages, and use the MPR flooding.
2. Use of a new multicast address called ALL-MANET_NODES, instead of the ALL_LINK_NODES.

Changes to the OLSR Routing Protocol

1. OLSR Packet format: The essential change needed for the existing OLSR packet format is to replace the IPv4 addresses with the IPv6 addresses in all messages, as highlighted in the OLSR specification [Clausen2003].
2. Multiple Interface Addresses: In IPv6, an interface can have several addresses. This paper proposed an OLSR node, for each interface, will have:
   - A link-local address: This address is usually obtained by autoconfiguration. It is temporary used as the source address for OLSR packets before autoconfiguration is completed.

- A site-local address: This is derived from the link-local address, in the fixed subnet OLSR_SUBNET for site-local prefix. This address is permanently used as the source for all OLSR packets, once autoconfiguration is completed.
- Any number (possibly zero) of additional global or site local unicast addresses, which are automatically or manually configured.

Neighbor Discovery

In IPv6, nodes (hosts and routers) use Neighbor Discovery [Narten1998] to determine the MAC addresses for neighbors on attached links and to quickly purge invalid cache values. Hosts also use Neighbor Discovery to find neighboring routers that are willing to forward packets on their behalf. Finally, nodes use the protocol to actively keep track of which neighbors are reachable and which are not, and to detect changed MAC addresses.
Routing table in the OLSR indicates the next hop for each reachable destination in the network. This next hop is one of the direct neighbors. This means that the neighbor solicitation for address resolution will work without any modification. In OLSR, gateways declare themselves to the entire network periodically. The neighbor discovery is adapted to OLSR. Consequently it is not necessary to do any modification to the classical procedure.

Autoconfiguration

IPv6 Stateless Address Autoconfiguration is based on several steps: after the creation of a link local address, the node must check whether the address is already in use by another interface of another node, somewhere in the network. In wired network, this means that all the links of the attached interfaces of the node are probed. If the address is not unique the process is interrupted, otherwise the autoconfiguration was successful and the address may be safely used.
In a MANET, the nodes on the links of the attached interfaces would include only the nodes with an interface within radio reach of the transmitter and not all the participating nodes. Hence, the uniqueness of the address is not guaranteed if the classical DAD (Duplicate Address Detection) procedure is applied. This paper proposed an algorithm, following the philosophy of the IPv6 DAD, to perform autoconfiguration in an OLSR network. The algorithm includes reactive probing (i.e. sending a request to the whole network and waiting for a possible answer), proactive checking (i.e. checking periodically for duplicate addresses) and collision resolution (i.e. what should be done upon detection of duplicate addresses) [Laouiti2004][Linux2010b].

5.3 Ad hoc On-demand Distance Vector routing for IPv6 (AODV6)

The operation of AODV for IPv6 is intended to mirror the operation of AODV for IPv4, with changes necessary to allow for transmission of 128-bit addresses in IPv6 instead of the traditional 32-bit addresses in IPv4.

Route Request (RREQ) Message Format

The format of the IPv6 Route Request message (RREQ) contains the same fields with the same functions as the RREQ message defined for IP version 4, except as follows:

1. Destination IP Address: The 128-bit IPv6 address of destination for which a route is desired.

2. Source IP Address: The 128-bit IPv6 address of the node which originated the Route Request.

Note, the order of the fields has been changed to enable alignment along the 128-bit boundaries.

Route Reply (RREP) Message Format

The format of the IPv6 Route Reply message (RREP) contains the same fields with the same functions as the RREP message defined for IP version 4, except as follows:

1. Prefix Size: The Prefix Size is 7 bits instead of 5, to account for the 128-bit IPv6 address space.
2. Destination Sequence Number: The destination sequence number associated to the route.
3. Destination IP Address: The 128-bit IP address of the destination for which a route is supplied.
4. Source IP Address: The 128-bit IP address of the source node which issued the RREQ for which the route is supplied.

Note, the order of the fields has been changed for better alignment.

Route Error Message Format

The format of the Route Error (RERR) message is identical to the format for the IPv4 RERR message except that the IP addresses are 128 bits, not 32 bits.

Route Reply Acknowledgment (RREP-ACK) Message Format

The RREP-ACK message is used to acknowledge receipt of an RREP message. It is used in cases where the link over which the RREP message is sent may be unreliable. It is identical in format to the RREP-ACK message for IPv4.

AODV for IPv6 Operation

The handling of AODV for IPv6 messages analogous to the operation of AODV for IPv4, except that the RREQ, RREP, RERR, and RREP-ACK messages described above are to be used instead; these messages have the formats appropriate for use with 128-bit IPv6 addresses [Perkins2000].

6. Conclusion

In this chapter, we introduced the general concepts of mobile ad hoc networks (MANET), routing in a MANET, and routing protocols for MANETs. For routing protocols, we summarized the key concepts of some popular proactive, reactive and hybrid protocols. We also introduced two popular MANET routing protocols for IPv6 networks, because more and more networks will adopt IPv6 addresses in the near future.

Each protocol introduced in this chapter has its own advantage and disadvantages in different MANET settings or environments. Therefore, it is hard to say which one is the best among them. So far, AODV is the most popular one for both IPv4 and IPv6 networks because it has more advantages than other protocols and it has been implemented successfully. In fact, the ODCR or the GOR algorithm could be a better choice.

7. References

[Abohasan2009] Abolhasan, Hagelstein and Wang, "Real-world Performance of Current Proactive Multi- hop Mesh Protocols", *Proceedings of IEEE APCC2009*, Shanghai, China, 10/2009.

[Au-Yong2006] Au-Yong, "Comparison of On-Demand Mobile Ad Hoc Network Routing Protocols under On/Off Source Traffic Effect", *Proceedings of NCS2006*, Chiang-Mai, Thailand, 3/2006.

[Chakeres2008] Chakeres and Perkins, "Dynamic MANET On-demand (DYMO) Routing", in: *Mobile Ad Hoc Networks Working Groups (draft-ietf-manet-dymo-12)*, available from: http://ianchak.com/dymo/draft-ietf-manet-dymo-12.txt, 2/2008.

[Clausen2003] Clausen and Jacquet, "Optimized Link State Routing Protocols (OLSR)", in: *Network Working Group – Request for Comments 3626*, available from: http://tools.ietf.org/html/rfc3626, 10/2003.

[Haas2002] Haas, Pearlman and Samar, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks" in: *Internet Draft (draft-ietf-manet-zone-zrp-04.txt)*, available from: http://people.ece.cornell.edu/~haas/wnl/Publications/draft-ietf-manet-zone-zrp-04.Txt, 7/2002.

[Johnson] Johnson, Maltz and Broch, "DSR: The Dynamic Source Routing Protocol for Multi-HopWireless Ad Hoc networks", in: *Ad Hoc Networking*, publisher: Edison Wesley, 2001.

[Johnson1994] David Johnson, "Routing in Ad Hoc Networks for Mobile Hosts", *Proceedings of IEEE WMCSA1994*, Santa Cruz, CA, 12/1994.

[Laouiti2004] Laouiti, Boudjit, Minet and Adjih, "OLSR for IPv6 Networks", *Proceedings of Med-Hoc-Net-2004*, available from http://www2.ece.ohio-state.edu/ medhoc04, 7/2004.

[Lee2007] Lee, Kimm and Reinhart, "A Global On-Demand Routing Protocol for Mobile Ad Hoc Networks", *Proceedings of IEEE NCA2007*, Boston, MA, 7/2007.

[Lee2009] Lee, Swanson and Liu, "An Efficient On-Demand Cache Routing Algorithm for Mobile Ad Hoc Networks", *Proceedings of IEEE ICCSIT2009*, Bejing, China, 8/2009.

[Linux2010a] Linux Reviews, "Why You Want IPv6 (Background – The IP Shortage)", available from: http://en.linuxreviews.org/Why_you_want_IPv6#Background:_The_IP_shortage, 8/2010.

[Linux2010b] Linux Reviews, "Linux Optimized Link State Routing Protocol (OLSR) IPv6 HOWTO", available from: http://linuxreviews.org/howtos/networking/ OLSR-IPv6-HOWTO/en/index.html, 8/2010.

[Murthy1996] Murthy and Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks", *Mobile Networks and Applications*, Volume 1, Issue 2, pp183-197, 1996.

[Narten1998] Narten, Noedmark and Simpson, , "Neighbor Discovery for IP Version 6 (IPv6)", in: *Network Working Group – Request for Comments 2461*, 12/1998, available from: http://tools.ietf.org/html/rfc2461

[Orderone2010] Orderone Networks 2010, "Mesh Network Routing Protocol", available from:
http://www.orderonenetworks.com/

[Perkins1994] Perkins and Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", *Proceedings of ACM SIGCOMM94*, pp234-244, London, 8/1994.

[Perkins2000] Perkins, Royer and Das, "Ad hoc On-Demand Distance Vector (AODV) Routing for IP version 6", in: *Mobile Ad Hoc Networking Working Group, Internet Draft*, available from:
http://members.shaw.ca/aodv6-sfu/aodv-ipv6-ietf-1.txt, 11/2000.

[Perkins2003] Perkins, Belding-Royer and Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", in: *Network Working Group – Request for Comments 3561*, available from:
http://tools.ietf.org/html/rfc3561, 7/2003.

[Sholander2002] Sholander, Yankopolus, Coccoli and Tabrizi, "Experimental Comparison of Hybrid and Proactive MANET Routing Protocols", *Proceedings of MILCOM2002*, Anaheim, CA, 10/2002.

[Wiki2010a] Wikipedia, "Mobile Ad Hoc Networks", available from: http://en.wikipedia.org/wiki/Mobile_ad_hoc_network, 8/2010.

[wiki2010b] Wikipedia, "Routing", available from: http://en.wikipedia.org/wiki/ Routing, 8/2010.

[Wiki2010c] Wikipedia, "List of Ad Hoc Routing Protocols", available from: http://en.wikipedia.org/wiki/List_of_ad-hoc_routing_protocols, 8/2009.

[Wiki2010d] Wikipedia, "Destination-Sequenced Distance-Vector Routing" available from: http://en.wikipedia.org/wiki/DSDV, 8/2010.

[Wiki2010e] Wikipedia, "Wireless Routing Protocols", available from: http://en.wikipedia.org/wiki/Wireless_Routing_Protocol, 8/2010.

[Wiki2010f] Wikipedia, "Optimized Link State Routing Protocols", available from: http://en.wikipedia.org/wiki/Optimized_Link_State_Routing_Protocol, 8/2010.

[Wiki2010g] Wikipedia, "Ad Hoc On-Demand Distance Vector Routing", available from: http://en.wikipedia.org/wiki/Ad-hoc_On-demand_Distance_Vector, 8/2010.

[Wiki2010h] Wikipedia, "Dynamic Source Routing", available from: http://en.wikipedia.org/wiki/Dynamic_Source_Routing, 8/2010.

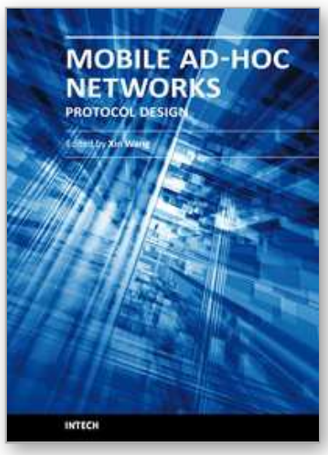
[Wiki2010i] Wikipedia, "DYMO", available from: http://en.wikipedia.org/wiki/ DYMO, 8/2010.

[Wiki2010j] Wikipedia, "Hybrid (both pro-active and reactive) Routing:, available from: http://en.wikipedia.org/wiki/List_of_ad-hoc_routing_protocols#Hybrid_.28both_ pro- active_and_reactive.29_routing, 8/2010.

[Wiki2010k] Wikipedia, "Zone Routing Protocol", available from: http://en.wikipedia.org/wiki/Zone_Routing_Protocol, 8/2010.

[Wiki2010l] Wikipedia, "Order One Network Protocol", available from: http://en.wikipedia.org/wiki/Order_One_Network_Protocol, 8/2010.

[Wiki2010m] Wikipedia, “Mobile IP”, available from: http://en.wikipedia.org/wiki/Mobile_IPv6, 8/2010.



Mobile Ad-Hoc Networks: Protocol Design

Edited by Prof. Xin Wang

ISBN 978-953-307-402-3
Hard cover, 656 pages
Publisher InTech
Published online 30, January, 2011
Published in print edition January, 2011

Being infrastructure-less and without central administration control, wireless ad-hoc networking is playing a more and more important role in extending the coverage of traditional wireless infrastructure (cellular networks, wireless LAN, etc). This book includes state-of-the-art techniques and solutions for wireless ad-hoc networks. It focuses on the following topics in ad-hoc networks: quality-of-service and video communication, routing protocol and cross-layer design. A few interesting problems about security and delay-tolerant networks are also discussed. This book is targeted to provide network engineers and researchers with design guidelines for large scale wireless ad hoc networks.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Fenglien Lee (2011). Routing in Mobile Ad hoc Networks, Mobile Ad-Hoc Networks: Protocol Design, Prof. Xin Wang (Ed.), ISBN: 978-953-307-402-3, InTech, Available from: http://www.intechopen.com/books/mobile-ad-hoc-networks-protocol-design/routing-in-mobile-ad-hoc-networks

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen