Mobile Ad-hoc Networks (manets)

By Donatas Sumyla
03/20/2006
Table of contents

1. Introduction ........................................................................................................ p.3

2. Mobile Ad-hoc Networks .................................................................................. p.4
   I. Ad-hoc networks versus Mobile Ad-hoc networks ........................................ p.4
   II. History of MANETs .................................................................................. p.7

3. Routing Protocols for MANETs ..................................................................... p.8
   I. Table-Driven Routing Protocols ................................................................ p.10
      1. Destination-Sequenced Distance-Vector Routing (DSDV) .................... p.10
      2. Clusterhead Gateway Switch Routing (CGSR) .................................... p.11
   II. Source-Initiated On-Demand Routing ....................................................... p.14
      1. Ad-Hoc On-Demand Distance Vector Routing (AODV) ..................... p.14
      2. Dynamic Source Routing (DSR) ....................................................... p.16
      3. Temporally-Ordered Routing Algorithm (TORA) ............................ p.18
      4. Associativity-Based Routing (ABR) ................................................. p.19
      5. Signal Stability Routing (SSR) ....................................................... p.20

Works Cited .......................................................................................................... p.22
1. Introduction

In today’s fast and rapidly growing world of technologies, more and more businesses understand the advantages of usage of computer networking. Depending on the firm’s size and resources it might be a small LAN containing only a few dozen computers; however in large corporations the networks can grow to enormous and complex mixture of computers and servers.

A computer network is a system for communication between computers. These networks may be fixed (cabled, permanent) or temporary (as via modems or null modems). Carrying instructions between calculation machines and early computers was done by human users. In September, 1940 George Stibitz used a teletype machine to send instructions for a problem set from his Model K at Dartmouth College in New Hampshire to his Complex Number Calculator in New York and received results back by the same means. Linking output systems like teletypes to computers was an interest at the Advanced Research Projects Agency ARPA when, in 1962, J.C.R. Licklider was hired and developed a working group he called the 'Intergalactic Network', a precursor to the ARPANet. In 1964 researchers at Dartmouth developed a time sharing system for distributed users of large computer systems. The same year, at MIT, a research group supported by General Electric and Bell Labs used a computer (DEC's PDP-8) to route and manage telephone connections. In 1968 Paul Baran proposed a network system consisting of datagrams or packets that could be used in a packet switching network between computer systems. In 1969 the University of California at Los Angeles, SRI (in Stanford), University of California at Santa Barbara, and the University of Utah were connected as the beginning of the
ARPANet network using 50 kbit/s circuits. Networks and the technologies needed to connect and communicate through and between them, continue to drive computer hardware, software, and peripherals industries. This expansion is mirrored by growth in the numbers and types of users of networks from researchers and businesses to families and individuals in everyday use.

Since their emergence in the 1970s, wireless networks have become increasingly popular in the computing industry. This is particularly true within the past decade which has seen wireless networks being adapted to enable mobility. There are currently two variations of mobile wireless networks. The first is known as infrastructured networks, i.e., those networks with fixed and wired gateways. The bridges for these networks are known as base stations. A mobile unit within these networks connects to, and communicates with, the nearest base station that is within its communication radius. As the mobile travels out of range of one base station and into the range of another, a “handoff” occurs from the old base station to the new, and the mobile is able to continue communication seamlessly throughout the network. Typical applications of this type of network include once wireless local area networks (WLANs).

The second type of mobile wireless network is the infrastructureless mobile network, commonly known as an ad-hoc network. Infrastructureless networks have no fixed routers; all nodes are capable of movement and can be connected dynamically in an arbitrary manner. Nodes of these networks function as routers which discover and maintain routes to other nodes in the network. Example applications of ad-hoc networks are emergency search-and-rescue operations, meetings or conventions in which persons wish to quickly share information, and data acquisition operations in inhospitable terrains.
2. Mobile Ad-hoc Networks

I. Ad-hoc Networks Versus Mobile Ad-hoc Networks

Ad-hoc networks form spontaneously without a need of an infrastructure or centralized controller. This type of peer-to-peer system infers that each node, or user, in the network can act as a data endpoint or intermediate repeater. Thus, all users work together to improve the reliability of network communications. These types of networks are also popularly known as "mesh networks" because the topology of network communications resembles a mesh.

The redundant communication paths provided by ad hoc mesh networks drastically improve fault tolerance for the network. Additionally, the ability for data packets to "hop" from one user to another effectively extends the network coverage area and provides a solution to overcome non-line of sight (LOS) issues.

Mobile applications present additional challenges for mesh networks as changes to the network topology are swift and widespread. Such scenarios require the use of Mobile Ad hoc Networking (MANET) technology to ensure communication routes are updated quickly and accurately. MANETs are self-forming, self-maintained, and self-healing, allowing for extreme network flexibility. While MANETs can be completely self-contained, they can also be tied to an IP-based global or local network (e.g. Internet or private networks). These are referred to as Hybrid MANETs.
Here is the illustration of the Hybrid MANET:

As you can see above we have three self-configuring mobile routers connected by wireless links creating MANET. However, as the routers approach the other two IP-based global
or local networks, they form a network which connects them all through those other networks, forming a hybrid MANET.

A mobile ad-hoc network (MANET) is a self-configuring network of mobile routers (and associated hosts) connected by wireless links - the union of which form a random topology. The routers are free to move randomly and organize themselves at random; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger Internet. Minimal configuration and quick deployment make ad hoc networks suitable for emergency situations like natural or human-induced disasters, military conflicts, emergency medical situations etc.

II. History of MANETs

The earliest MANETs were called “packet radio” networks, and were sponsored by DARPA in the early 1970s. BBN Technologies and SRI International designed, built, and experimented with these earliest systems. Experimenters included Jerry Burchfiel, Robert Kahn, and Ray Tomlinson of later TENEX, Internet and email fame. It is interesting to note that these early packet radio systems predated the Internet, and indeed were part of the motivation of the original Internet Protocol suite. Later DARPA experiments included the Survivable Radio Network (SURAN) project, which took place in the 1980s. Another third wave of academic activity started in the mid 1990s with the advent of inexpensive 802.11 radio cards for personal computer. Current MANETs are designed primary for military utility; examples include JTRS and NTDR.
The popular IEEE 802.11 ("Wi-Fi") wireless protocol incorporates an ad-hoc networking system when no wireless access points are present, although it would be considered a very low-grade ad-hoc protocol by specialists in the field. The IEEE 802.11 system only handles traffic within a local "cloud" of wireless devices. Each node transmits and receives data, but does not route anything between the network's systems. However, higher-level protocols can be used to aggregate various IEEE ad-hoc networks into MANETs.

The MIT Media Lab $100 laptop program hopes to develop a cheap laptop for mass distribution (>1 million at a time) to developing countries for education. The laptops will use ad-hoc wireless mesh networking to develop their own communications network out of the box.

3. Routing Protocols for MANETs

In order to facilitate communication within the network, a routing protocol is used to discover routes between nodes. The primary goal of such an ad-hoc network routing protocol is correct and efficient route establishment between a pair of nodes so that messages may be delivered in a timely manner. Route construction should be done with a minimum of overhead and bandwidth consumption. An Ad-hoc routing protocol is a convention or standard that controls how nodes come to agree which way to route packets between computing devices in a MANET. In ad-hoc networks, nodes do not have a priori knowledge of topology of network around them, they have to discover it. The basic idea is that a new node announces its presence and listens to broadcast announcements from its neighbors. The node learns about new near nodes and ways to reach them, and announces that it can also reach those nodes. As time goes on, each node knows about all other nodes and one or more ways how to reach them.
Routing algorithms have to:

- Keep routing table reasonably small;
- Choose best route for given destination (this can be the fastest, most reliable, highest throughput, or cheapest route);
- Keep table up-to-date when nodes die, move or join;
- Require small amount of messages/time to converge.

In a wider context, an ad-hoc protocol can also mean an improvised and often impromptu protocol established for a particular specific purpose.

Since the advent of DARPA packet radio networks in the early 1970s, numerous protocols have been developed for ad-hoc mobile networks. Such protocols must deal with the typical limitations of these networks, which include high power consumption, low bandwidth, and high error rates. As shown in Figure 1 below, these routing protocols may generally be categorized as: (a) table-driven and (b) source-initiated on-demand driven. Solid lines in this figure represent direct descendants while dotted lines depict logical descendants. Despite being designed for the same type of underlying network, the characteristics of each of these protocols are quite distinct.

Figure 1: Categorization of Ad-Hoc Routing Protocols
I. Table-Driven Routing Protocols

The table-driven routing protocols attempt to maintain consistent, up-to-date routing information from each node to every other node in the network. These protocols require each node to maintain one or more tables to store routing information, and they respond to changes in network topology by propagating updates throughout the network in order to maintain a consistent network view. The areas where they differ are the number of necessary routing-related tables and the methods by which changes in network structure are broadcast.

\textit{Destination-Sequenced Distance-Vector Routing (DSDV)}

The Destination-Sequenced Distance-Vector Routing protocol (DSDV) is a table-driven algorithm based on the classical Bellman-Ford routing mechanism. The improvements made to the Bellman-Ford algorithm include freedom from loops in routing tables.

Every mobile node in the network maintains a routing table in which all of the possible destinations within the network and the number of hops to each destination are recorded. Each entry is marked with a sequence number assigned by the destination node. The sequence numbers enable the 2 mobile nodes to distinguish stale routes from new ones, thereby avoiding the formation of routing loops. Routing table updates are periodically transmitted throughout the network in order to maintain table consistency. To help ease the potentially large amount of network traffic that such updates can generate, route updates can employ two possible types of packets. The first is known as a “full dump.” This type of packet carries all available routing information and can require multiple network protocol data units (NPDUs). During periods of occasional movement, these packets are transmitted infrequently. Smaller “incremental” packets
are used to relay only that information which has changed since the last full dump. Each of these broadcasts should fit into a standard size NPDU, thereby decreasing the amount of traffic generated. The mobile nodes maintain an additional table where they store the data sent in the incremental routing information packets.

New route broadcasts contain the address of the destination, the number of hops to reach the destination, the sequence number of the information received regarding the destination, as well as a new sequence number unique to the broadcast. The route labeled with the most recent sequence number is always used. In the event that two updates have the same sequence number, the route with the smaller metric is used in order to optimize (shorten) the path. Mobiles also keep track of the settling time of routes, or the weighted average time that routes to a destination will fluctuate before the route with the best metric is received. By delaying the broadcast of a routing update by the length of the settling time, mobiles can reduce network traffic and optimize routes by eliminating those broadcasts that would occur if a better route was discovered in the very near future.

**Clusterhead Gateway Switch Routing (CGSR)**

The Clusterhead Gateway Switch Routing (CGSR) protocol differs from the previous protocol in the type of addressing and network organization scheme employed. Instead of a “fat” network, CGSR is a clustered multihop mobile wireless network with several heuristic routing schemes. By having a cluster head controlling a group of ad-hoc nodes, a framework for code separation (among clusters), channel access, routing and bandwidth allocation can be achieved. A cluster head selection algorithm is utilized to elect a node as the cluster head using a distributed algorithm within the cluster. The disadvantage of having a cluster head scheme is that frequent
cluster head changes can adversely affect routing protocol performance since nodes are busy in
cluster head selection rather than packet relaying. Hence, instead of invoking cluster head
reselection every time the cluster membership changes, a Least Cluster Change (LCC) clustering
algorithm is introduced. Using LCC, cluster heads only change when two cluster heads come
into contact, or when a node moves out of contact of all other cluster heads.

CGSR uses DSDV as the underlying routing scheme, and hence has much of the same
overhead as DSDV. However, it modifies DSDV by using a hierarchical cluster head-to-gateway
routing approach to route traffic from source to destination. Gateway nodes are nodes that are
within communication range of two or more cluster heads. A packet sent by a node is first routed
to its cluster head, and then the packet is routed from the cluster head to a gateway to another
cluster head, and so on until the cluster head of the destination node is reached. The packet is
then transmitted to the destination. Using this method, each node must keep a “cluster member
table” where it stores the destination cluster head for each mobile node in the network. These
cluster member tables are broadcast by each node periodically using the DSDV algorithm. Nodes
update their cluster member tables on the reception of such a table from a neighbor.

In addition to the cluster member table, each node must also maintain a routing table,
which is used to determine the next hop in order to reach the destination. On receiving a packet,
a node will consult its cluster member table and routing table to determine the nearest cluster
head along the route to the destination. Next the node will check its routing table to determine
the node in order to reach the selected cluster head. It then transmits the packet to this node.
The Wireless Routing Protocol (WRP)

The Wireless Routing Protocol (WRP) is a table-based protocol with the goal of maintaining routing information among all nodes in the network. Each node in the network is responsible for maintaining four tables: (a) distance table, (b) routing table, (c) link-cost table, and (d) message retransmission list (MRL) table. Each entry of the MRL contains the sequence number of the update message, a retransmission counter, an acknowledgment-required flag vector with one entry per neighbor, and a list of updates sent in the update message. The MRL records which updates in an update message need to be retransmitted and which neighbors should acknowledge the retransmission.

Mobiles inform each other of link changes through the use of update messages. An update message is sent only between neighboring nodes and contains a list of updates (the destination, the distance to the destination, and the predecessor of the destination), as well as a list of responses indicating which mobiles should acknowledge (ACK) the update. Mobiles send update messages after processing updates from neighbors or detecting a change in a link to a neighbor. In the event of the loss of a link between two nodes, the nodes send update messages to their neighbors. The neighbors then update their distance table entries and check for new possible paths through other nodes. Any new paths are relayed back to the original nodes so that they can update their tables accordingly.

Nodes learn of the existence of their neighbors from the receipt of acknowledgments and other messages. If a node is not sending messages, it must send a hello message within a specified time period to ensure connectivity. Otherwise, the lack of messages from the node indicates the failure of that link; this may cause a false alarm. When a mobile receives a hello
message from a new node, that new node is added to the mobile's routing table, and the mobile sends the new node a copy of its routing table information.

Part of the novelty of WRP stems from the way in which it achieves loop freedom. In WRP, routing nodes communicate the distance and second-to-last hop information for each destination in the wireless networks. WRP belongs to the class of path finding algorithms with an important exception. It avoids the “count-to-infinity” problem by forcing each node to perform consistency checks of predecessor information reported by all its neighbors. This ultimately (though not instantaneously) eliminates looping situations and provides faster route convergence when a link failure event occurs.

II. Source-Initiated On-Demand Routing

A different approach from table-driven routing is source-initiated on-demand routing. This type of routing creates routes only when desired by the source node. When a node requires a route to a destination, it initiates a route discovery process within the network. This process is completed once a route is found or all possible route permutations have been examined. Once a route has been established, it is maintained by some form of route maintenance procedure until either the destination becomes inaccessible along every path from the source or until the route is no longer desired.

Ad-hoc On-Demand Distance Vector Routing (AODV)

The Ad-hoc On-Demand Distance Vector (AODV) routing protocol builds on the DSDV algorithm previously described. AODV is an improvement on DSDV because it typically
minimizes the number of required broadcasts by creating routes on an on-demand basis, as opposed to maintaining a complete list of routes as in the DSDV algorithm. The authors of AODV classify it as a pure on-demand route acquisition system, as nodes that are not on a selected path do not maintain routing information or participate in routing table exchanges.

When a source node desires to send a message to some destination node and does not already have a valid route to that destination, it initiates a Path Discovery process to locate the other node. It broadcasts a route request (RREQ) packet to its neighbors, which then forward the request to their neighbors, and so on, until either the destination or an intermediate node with a “fresh enough” route to the destination is located. AODV utilizes destination sequence numbers to ensure all routes are loop-free and contain the most recent route information. Each node maintains its own sequence number, as well as a broadcast ID. The broadcast ID is incremented for every RREQ the node initiates, and together with the node's IP address, uniquely identifies a RREQ. Along with its own sequence number and the broadcast ID, the source node includes in the RREQ the most recent sequence number it has for the destination. Intermediate nodes can reply to the RREQ only if they have a route to the destination whose corresponding destination sequence number is greater than or equal to that contained in the RREQ.

During the process of forwarding the RREQ, intermediate nodes record in their route tables the address of the neighbor from which the first copy of the broadcast packet is received, thereby establishing a reverse path. If additional copies of the same RREQ are later received, these packets are discarded. Once the RREQ reaches the destination or an intermediate node with a fresh enough route, the destination/intermediate node responds by unicasting a route reply (RREP) packet back to the neighbor from which it first received the RREQ. As the RREP is routed back along the reverse path, nodes along this path set up forward route entries in their
route tables which point to the node from which the RREP came. These forward route entries indicate the active forward route. Associated with each route entry is a route timer which will cause the deletion of the entry if it is not used within the specified lifetime. Because the RREP is forwarded along the path established by the RREQ, AODV only supports the use of symmetric links.

Routes are maintained as follows. If a source node moves, it is able to reinitialize the route discovery protocol to find a new route to the destination. If a node along the route moves, its upstream neighbor notices the move and propagates a link failure notification message (a RREP with infinite metric) to each of its active upstream neighbors to inform them of the erasure of that part of the route. These nodes in turn propagate the link failure notification to their upstream neighbors, and so on until the source node is reached. The source node may then choose to re-initiate route discovery for that destination if a route is still desired.

_Dynamic Source Routing (DSR)_

The Dynamic Source Routing (DSR) protocol is an on-demand routing protocol that is based on the concept of source routing. Mobile nodes are required to maintain route caches that contain the source routes of which the mobile is aware. Entries in the route cache are continually updated as new routes are learned.

The protocol consists of two major phases: route discovery and route maintenance. When a mobile node has a packet to send to some destination, it first consults its route cache to determine whether it already has a route to the destination. If it has an unexpired route to the destination, it will use this route to send the packet. On the other hand, if the node does not have such a route, it initiates route discovery by broadcasting a route request packet. This route
request contains the address of the destination, along with the source node's address and a unique identification number. Each node receiving the packet checks whether it knows of a route to the destination. If it does not, it adds its own address to the route record of the packet and then forwards the packet along its outgoing links. To limit the number of route requests propagated on the outgoing links of a node, a mobile only forwards the route request if the request has not yet been seen by the mobile and if the mobile's address does not already appear in the route record.

A route reply is generated when either the route request reaches the destination itself, or when it reaches an intermediate node which contains in its route cache an unexpired route to the destination. By the time the packet reaches either the destination or such an intermediate node, it contains a route record yielding the sequence of hops taken. If the node generating the route reply is the destination, it places the route record contained in the route request into the route reply. If the responding node is an intermediate node, it will append its cached route to the route record and then generate the route reply. To return the route reply, the responding node must have a route to the initiator. If it has a route to the initiator in its route cache, it may use that route. Otherwise, if symmetric links are supported, the node may reverse the route in the route record. If symmetric links are not supported, the node may initiate its own route discovery and piggyback the route reply on the new route request.

Route maintenance is accomplished through the use of route error packets and acknowledgments. Route error packets are generated at a node when the data link layer encounters a fatal transmission problem. When a route error packet is received, the hop in error is removed from the node's route cache and all routes containing the hop are truncated at that point. In addition to route error messages, acknowledgments are used to verify the correct
operation of the route links. Such acknowledgments include passive acknowledgments, where a mobile is able to hear the next hop forwarding the packet along the route.

_Temporally-Ordered Routing Algorithm (TORA)_

TORA (Temporally-Ordered Routing Algorithm) is a highly adaptive, loop-free, distributed routing algorithm based on the concept of link reversal. TORA is proposed to operate in a highly dynamic mobile networking environment. It is source-initiated and provides multiple routes for any desired source/destination pair. The key design concept of TORA is the localization of control messages to a very small set of nodes near the occurrence of a topological change. To accomplish this, nodes need to maintain routing information about adjacent (1-hop) nodes. The protocol performs three basic functions: (a) route creation, (b) route maintenance, and (c) route erasure.

During the route creation and maintenance phases, nodes use a “height” metric to establish a directed acyclic graph (DAG) rooted at the destination. Thereafter, links are assigned a direction (upstream or downstream) based on the relative height metric of neighboring nodes. This process of establishing a DAG is similar to the query/reply process proposed in LMR (Lightweight Mobile Routing). In times of node mobility, the DAG route is broken and route maintenance is necessary to re-establish a DAG rooted at the same destination. Upon failure of the last downstream link, a node generates a new reference level which results in the propagation of that reference level by neighboring nodes, effectively coordinating a structured reaction to the failure. Links are reversed to reflect the change in adapting to the new reference level. This has the same effect as reversing the direction of one or more links when a node has no downstream links.
Timing is an important factor for TORA because the “height” metric is dependent on the logical time of a link failure; TORA assumes all nodes have synchronized clocks (accomplished via an external time source such as Global Positioning System). TORA's metric is a quintuple comprised of five elements, namely: (a) logical time of a link failure, (b) the unique ID of the node that defined the new reference level, (c) a reflection indicator bit, (d) a propagation ordering parameter, and (e) the unique ID of the node. The first three elements collectively represent the reference level. A new reference level is defined each time a node loses its last downstream link due to a link failure. TORA's route erasure phase essentially involves flooding a broadcast “clear packet” (CLR) throughout the network to erase invalid routes.

Associativity-Based Routing (ABR)

The Associativity-Based Routing (ABR) protocol is free from loops, deadlock, and packet duplicates, and defines a new routing metric for ad-hoc mobile networks. This metric is known as the degree of association stability. In ABR, a route is selected based on the degree of association stability of mobile nodes. Each node periodically generates a beacon to signify its existence. When received by neighboring nodes, this beaconing causes their associativity tables to be updated. For each beacon received, the associativity tick of the current node with respect to the beacons node is incremented. Association stability is defined by connection stability of one node with respect to another node over time and space. A high degree of association stability may indicate a low state of node mobility, while a low degree may indicate a high state of node mobility. Associativity ticks are reset when the neighbors of a node or the node itself moves out of proximity. A fundamental objective of ABR is to derive longer-lived routes for ad-hoc mobile networks.
The three phases of ABR are: (a) route discovery, (b) route re-construction (RRC), and (c) route deletion. The route discovery phase is accomplished by a broadcast query and await-reply (BQ-REPLY) cycle. A node desiring a route broadcasts a BQ message in search of mobiles that have a route to the destination. All nodes receiving the query (that are not the destination) append their addresses and their associativity ticks with their neighbors along with QoS information to the query packet. A successor node erases its upstream node neighbors' associativity tick entries and retains only the entry concerned with itself and its upstream node. In this way, each resultant packet arriving at the destination will contain the associativity ticks of the nodes along the route to the destination. The destination is then able to select the best route by examining the associativity ticks along each of the paths. In the case where multiple paths have the same overall degree of association stability, the route with the minimum number of hops is selected. The destination then sends a REPLY packet back to the source along this path. Nodes propagating the REPLY mark their routes as valid. All other routes remain inactive and the possibility of duplicate packets arriving at the destination is avoided.

**Signal Stability Routing (SSR)**

Unlike the algorithms described so far, SSR selects routes based on the signal strength between nodes and on a node's location stability. This route selection criteria has the effect of choosing routes that have "stronger" connectivities. SSR can be divided into two cooperative protocols: the Dynamic Routing Protocol (DRP) and the Static Routing Protocol (SRP).

The DRP is responsible for the maintenance of the Signal Stability Table (SST) and the Routing Table (RT). The SST records the signal strength of neighboring nodes, which is obtained by periodic beacons from the link layer of each neighboring node. The signal strength
may be recorded as either a strong or weak channel. All transmissions are received by, and processed in, the DRP. After updating all appropriate table entries, the DRP passes a received packet to the SRP.

The SRP processes packets by passing the packet up the stack if it is the intended receiver or looking up the destination in the RT and then forwarding the packet if it is not. If no entry is found in the RT for the destination, a route-search process is initiated to find a route. Route requests are propagated throughout the network but are only forwarded to the next hop if they are received over strong channels and have not been previously processed (to prevent looping). The destination chooses the first arriving route-search packet to send back because it is most probable that the packet arrived over the shortest and/or least congested path. The DRP then reverses the selected route and sends a route-reply message back to the initiator. The DRP of the nodes along the path update their RTs accordingly.

Route-search packets arriving at the destination have necessarily chosen the path of strongest signal stability, as the packets are dropped at a node if they have arrived over a weak channel. If there is no route-reply message received at the source within a specific timeout period, the source changes the PREF field in the header to indicate that weak channels are acceptable, as these may be the only links over which the packet can be propagated.

When a failed link is detected within the network, the intermediate nodes send an error message to the source indicating which channel has failed. The source then initiates another route-search process to find a new path to the destination. The source also sends an erase message to notify all nodes of the broken link.
Works cited:

http://en.wikipedia.org/wiki/Mobile_ad-hoc_network


http://www.cwc.oulu.fi/~hernia/linkit.html