A Review of Current Routing Protocols for MANET

Variations of mobile wireless networks:
- Infrastructured Networks: Networks with fixed wired gateways.
- Infrastructureless Mobile Networks or Ad-Hoc Networks.

Application areas
- Military environments
  - Battle field: sensors, soldiers, vehicles
- Emergency operations
  - search-and-rescue
  - policing and fire fighting
- Civilian environments
  - conference halls
  - sports stadiums, Library, etc.
- Personal area networking
  - laptop, PDA, cell phone, ear phone, wrist watch

Why is routing difficult in Ad Hoc Networks?
- Dynamic nature of the network
- No specific devices to do routing
- Limitations of Ad Hoc Networks like
  -- high power consumption
  -- low bandwidth
  -- high error rates

Ad Hoc Protocol Routing Requirements
- Simple, reliable and efficient
- Distributed but lightweight in nature
- Quickly adapts to changes in topology
- Protocol reaction to topology changes should result in minimal control overhead
- Bandwidth efficient
- Mobility management involving user location and hand-off management
**Categorization of Ad-Hoc Routing Protocols**

- **Table Driven Based Routing (Proactive)**
  - Maintain table of all active links in network
  - Calculate the shortest path from table
  - Update table whenever nodes move
  - Immediate tell if node is reachable
  - Data can be sent immediately
  - Very high overheads

- **Distance Vector**
  - Basic Routing Protocol
    - known also as Distributed Bellman-Ford or RIP
  - Every node maintains a routing table
    - all available destinations
    - the next node to reach to destination
    - the number of hops to reach the destination
  - Periodically send table to all neighbors to maintain topology
  - Bi-directional links are required!

**Distance Vector (Tables)**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Next Dest.</th>
<th>Metric</th>
<th>Next Dest.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>AA</td>
<td>1</td>
<td>BB</td>
</tr>
<tr>
<td>1</td>
<td>CC</td>
<td>1</td>
<td>BB</td>
</tr>
<tr>
<td>2</td>
<td>CC</td>
<td>2</td>
<td>BA</td>
</tr>
<tr>
<td>3</td>
<td>BC</td>
<td>3</td>
<td>BC</td>
</tr>
</tbody>
</table>

**Distance Vector (Update)**

- B broadcasts the new routing information to his neighbors
- Routing table is updated

<table>
<thead>
<tr>
<th>Metric</th>
<th>Next Dest.</th>
<th>Metric</th>
<th>Next Dest.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>AA</td>
<td>1</td>
<td>BB</td>
</tr>
<tr>
<td>1</td>
<td>CC</td>
<td>1</td>
<td>BB</td>
</tr>
<tr>
<td>2</td>
<td>CC</td>
<td>2</td>
<td>BA</td>
</tr>
<tr>
<td>3</td>
<td>BC</td>
<td>3</td>
<td>BC</td>
</tr>
</tbody>
</table>

**Distance Vector (New Node)**

- D broadcasts to update tables of C, B, A with new entry for D

<table>
<thead>
<tr>
<th>Metric</th>
<th>Next Dest.</th>
<th>Metric</th>
<th>Next Dest.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>AA</td>
<td>1</td>
<td>BB</td>
</tr>
<tr>
<td>1</td>
<td>CC</td>
<td>1</td>
<td>BB</td>
</tr>
<tr>
<td>2</td>
<td>CC</td>
<td>2</td>
<td>BA</td>
</tr>
<tr>
<td>3</td>
<td>BC</td>
<td>3</td>
<td>BC</td>
</tr>
</tbody>
</table>

- **Distance Vector (Update)**
- B broadcasts the new routing information to his neighbors
- Routing table is updated
Distance Vector (Broken Link)

Distance Vector (Loops)

Distance Vector (Count to Infinity)

Distance Vector

- DV not suited for ad-hoc networks!
  - Loops
    - Bandwidth reduction in network
    - Unnecessary work for loop nodes
  - Count to Infinity
    - Very slow adaptation to topology changes.
- Solution -> DSDV

DSDV Protocol

- Keep the simplicity of Distance Vector
- Guarantee Loop Freeness
  - New Table Entry for Destination Sequence Number
- Allow fast reaction to topology changes
  - Make immediate route advertisement on significant changes in routing table
  - but wait with advertising of unstable routes (damping fluctuations)

DSDV (Table Entries)

- Sequence number originated from destination. Ensures loop freeness.
- Install Time when entry was made (used to delete stale entries from table)
- Stable Data Pointer to a table holding information on how stable a route is. Used to damp fluctuations in network.
DSDV (Route Advertisements)

- Advertise to each neighbor own routing information
  - Destination Address
  - Metric = Number of Hops to Destination
  - Destination Sequence Number
  - Other info (e.g. hardware addresses)

- Rules to set sequence number information
  - On each advertisement increase own destination sequence number (use only even numbers)
  - If a node is no more reachable (timeout) increase sequence number of this node by 1 (odd sequence number) and set metric = \(\infty\).

DSDV (Route Selection)

- Update information is compared to own routing table
  1. Select route with higher destination sequence number (This ensures to use always newest information from destination)
  2. Select the route with better metric when sequence numbers are equal.

DSDV (Tables)

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Seq</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

DSDV (Route Advertisement)

- B increases Seq.Nr from 100 -> 102
- B broadcasts routing information to Neighbors A, C including destination sequence numbers

DSDV (New Node)

1. D broadcast for first time
   Send Sequence number D-000
   Then immediately broadcast own table.

2. Insert entry for D with sequence number D-000
   Then immediately broadcast own table.

3. C increases its sequence number to C-592 then broadcasts its new table.

4. B gets this new information and updates its table.

DSDV (New Node cont.)

4. B gets this new information and updates its table.

(C, 1, C-588)

(D, 0, D-000)
DSDV (no loops, no count to infinity)

- B does its broadcast
  - No effect on C (C knows that it has stale information because C has higher seq. number for destination D)
  - No loop -> no count to infinity

1. Node C detects broken link:
   - Increase Seq. Nr. by 1
   (Only case where not the destination sets the seq. number -> odd number)

2. B does its broadcast
   - No affect on C (C knows that B has stale information because C has higher seq. number for destination D)
   - No loop -> no count to infinity

DSDV (Immediate Advertisement)

- Immediate propagation
  - B to A: (route information has higher seq. Nr. -> replace table entry)
  - C to B: (route information has higher seq. Nr. -> replace table entry)

DSDV (Respond to Topology Changes)

- Immediate advertisements
  - Information on new Routes, broken Links, metric change is immediately propagated to neighbors.

- Full/Incremental Update:
  - Full Update: Send all routing information from own table.
  - Incremental Update: Send only entries that has changed. (Make it fit into one single packet)

Route selection criteria

- Routes are preferred if the sequence numbers are newer
- If the sequence numbers are the same, the one with better metric is preferred
- Keep track of the settling time of routes - the weighted average time that routes to a destination will fluctuate before the route with best metric is received

Summary of DSDV Routing

- Essentially a modification to Bellman-Ford routing algorithm
- Using sequence number to guarantee loop-free paths
- Relies on periodic exchange of routing information.
- Inefficient due to periodic update transmissions even no changes in topology
- Overhead grows as O(n^2), limiting scalability

CGSR (Clusterhead Gateway Switch Routing)

- A clustered multihop mobile wireless network with several heuristic routing schemes.
- A cluster head selection algorithm is utilized to elect a node as the cluster head using a distributed algorithm within the cluster
- Disadvantage: Cluster head changes can adversely affect routing protocol performance since nodes are busy in cluster head selection
CGSR (Clusterhead Gateway Switch Routing)

- A packet sent by a node is first routed to its cluster head and then the packet is routed from the cluster head to gateway to another cluster head and so on until the cluster head of the destination node is reached.
- CGSR uses DSDV as the underlying routing scheme. Same overhead as DSDV

WRP (Wireless Routing Protocol)

- Each node in the network is responsible for maintaining four tables
  - Distance table
  - Routing table
  - Link-cost table
  - Message retransmission list table
- Mobile inform each other of link changes through the use of update messages.
- An update message is sent only between neighboring nodes.

WRP (Wireless Routing Protocol)

- In event of the loss of a link between two nodes, the nodes send update messages to their neighbors.
- Nodes learn of the existence of their neighbors from the receipt of acknowledgements and other messages.
- Routing nodes communicate the distance and second-to-last hop information for each destination in the wireless networks

On-demand Routing (Reactive)

- Find routes as needed
- Cache information from other nodes’ requests
- No static overhead
- Slow start before transmitting data
- AODV, DSR
AODV (Ad Hoc On-Demand Distance Vector) routing

• AODV is an improvement on DSDV because it typically minimizes the number of required broadcast by creating routes on demand basis.
• If valid route doesn’t existing, it initiates a path discovery process to locate other path.
• It broadcast route request packet to its neighbors, which then forward request to their neighbors, and so on, until either destination or intermediate node with “fresh enough” route is located.
• Each node maintains its own sequence number to ensure all routes are loop-free and contains most recent route information.
• If source node moves, it is able to reinitiate route discovery protocol to find a new route to the destination.
• Periodic broadcast “hello” messages can be used to maintain local connectivity.

Route Table Information in AODV

• The destination address
• The next hop IP address
• The destination sequence number
• A list of precursor nodes to reach the destination(for the purpose of route maintenance if link breaks)
• A lifetime for each route(expire the unused route)

Route Discovery in AODV

Source Node:
• Initiate RREQ, packet contains:
  – Source node IP address and current sequence #
  – Destination IP address and last known sequence #
  – A broadcast ID
• Set a timer to wait for reply

Intermediate nodes
• Check the unique identifier (Source IP address & broadcast ID)of the RREQ
• If already seen, discards the packet
• If not,
  – Set up a reverse route for the source node, associated with a lifetime
  – Increase the RREQ’s hop count
  – Broadcast the RREQ to its neighbors

Node responds to the RREQ (not necessary destination node)
• Must have an unexpired entry for destination
• The sequence # associated >= the sequence # indicated in the RREQ
• Unicasts a RREP back to the source, using the node from which it received the RREQ as the next hop

C responds to the RREQ with RREP, which includes
• Its record of the destination’s sequence #.
• The hop count from C to D
• The lifetime of the route from C to D
• Unicasts the RREP to source node through A
Forward Path Setup

A receives RREP, sets up a *forward path* entry to D in its route table, includes:
- IP address of D, IP address of the neighbor this RREP received (C)
- Its hop count to D
- The associated life time as contained in RREP

Summary of AODV Routing

- On demand, distance-vector routing protocol for highly mobile wireless nodes
- Support both *unicast* and *multicast*
- Good chance to scale to large node population
  (Reported results for
  - AODV: node population as great as 10,000 nodes
  - Others: of 100 to 200 nodes)

Dynamic Source Routing (DSR)

- When node S wants to send a packet to node D, but does not know a route to D, node S initiates a route discovery
- Source node S floods Route Request (RREQ)
- Each node appends its own identifier when forwarding RREQ

Route Discovery in DSR

- Node H receives packet RREQ from two neighbors: potential for collision

![Route Discovery in DSR](image)
Route Discovery in DSR

• Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once

Route Discovery in DSR

• Nodes J and K both broadcast RREQ to node D
• Since nodes J and K are hidden from each other, their transmissions may collide

Route Discovery in DSR

• Node D does not forward RREQ, because node D is the intended target of the route discovery

Route Discovery in DSR

• Destination D on receiving the first RREQ, sends a Route Reply (RREP)
• RREP is sent on a route obtained by reversing the route appended to received RREQ
• RREP includes the route from S to D on which RREQ was received by node D

Route Reply in DSR

• Route Reply can be sent by reversing the route in Route Request (RREQ) only if links are guaranteed to be bi-directional
  – To ensure this, RREQ should be forwarded only if it received on a link that is known to be bi-directional
• If unidirectional (asymmetric) links are allowed, then RREP may need a route discovery for S from node D
  – Unless node D already knows a route to node S
  – If a route discovery is initiated by D for a route to S, then the Route Reply is piggybacked on the Route Request from D.

Represents RREP control message
Dynamic Source Routing (DSR)

- Node S on receiving RREP, caches the route included in the RREP
- When node S sends a data packet to D, the entire route is included in the packet header — hence the name source routing
- Intermediate nodes use the source route included in a packet to determine to whom a packet should be forwarded

Data Delivery in DSR

Packet header size grows with route length

DSR Optimization: Route Caching

- Each node caches a new route it learns by any means
- When node S finds route [S,E,F,J,D] to node D, node S also learns route [S,E,F] to node F
- When node F forwards Route Reply RREP [S,E,F,J,D], node F learns route [F,J,D] to node D
- When node E forwards Data [S,E,F,J,D] it learns route [E,F,J,D] to node D
- A node may also learn a route when it overhears Data packets

Route Cache

<table>
<thead>
<tr>
<th>Dest.</th>
<th>Route</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>X</td>
</tr>
<tr>
<td>B</td>
<td>direct</td>
</tr>
<tr>
<td>C</td>
<td>B</td>
</tr>
<tr>
<td>D</td>
<td>B</td>
</tr>
<tr>
<td>E</td>
<td>BD</td>
</tr>
</tbody>
</table>

Use of Route Caching

- When node S learns that a route to node D is broken, it uses another route from its local cache, if such a route to D exists in its cache. Otherwise, node S initiates route discovery by sending a route request
- Node X on receiving a Route Request for some node D can send a Route Reply if node X knows a route to node D
- Use of route cache
  - can speed up route discovery
  - can reduce propagation of route requests

Use of Route Caching

[DSR maintains the cached routes in a tree format]
**Use of Route Caching:**
Can Speed up Route Discovery

When node Z sends a route request for node C, node K sends back a route reply [Z,K,G,C] to node Z using a locally cached route.

**Dynamic Source Routing:**
Advantages

- Routes maintained only between nodes who need to communicate – reduces overhead of route maintenance
- Route caching can further reduce route discovery overhead
- A single route discovery may yield many routes to the destination, due to intermediate nodes replying from local caches

**ABR (Associativity-Based Routing):**
- Protocol is free from loops, deadlocks and packet duplicates.
- Each node periodically generates a beacon to signify its existence. When received by neighboring nodes, this beacon causes their associativity tables to updated and associativity tick of the current node with respect to beaconing node is incremented.
- The protocol performs three basic functions
  - Route discovery
  - Broadcast query and await-reply
  - Partial route discovery, invalid route erasure, valid route updates and new updates.
  - Route deletion
  - When route is not longer needed, the source node send route deletion broadcast so that nodes along to route update their routing tables.

**SSR (Signal Stability Routing):**
Protocol selects routes based on the signal strength between nodes and a node’s location stability
- SSR can be divided two cooperative protocols
- Dynamic Routing Protocol
  - Responsible for maintenance of the Signal Stability Table (signal strength of neighboring nodes) and Routing Table
- Static Routing Protocol
  - Process packets by passing the packet up the stack
  - Route-search packets arriving at destination have necessarily chosen the path of strongest signal stability
  - When a failed links is detected within the network, intermediate nodes send an error message to the source indicating which channel has failed and initiates another route-search process

**Key Features**
- Table-driven:
- DSDV is inefficient because of periodic update transmissions.
- CGSR can be employed several heuristic methods and improve protocol performance
- WRP requires lot of memory and uses “hello” packets.

**Comparison**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>On-Demand</th>
<th>Table-Driven</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability of routing information</td>
<td>Available when needed</td>
<td>Always available regardless of need</td>
</tr>
<tr>
<td>Routing philosophy</td>
<td>Flat</td>
<td>Mostly flat, except for CGRS</td>
</tr>
<tr>
<td>Periodic route updates</td>
<td>Not required</td>
<td>Required</td>
</tr>
<tr>
<td>Coping with mobility</td>
<td>Use local route discovery as in ABR and SSR</td>
<td>Inform other nodes to achieve a consistent routing table</td>
</tr>
<tr>
<td>Signaling traffic generated</td>
<td>Grows with increasing mobility of active routes (as in ABR)</td>
<td>Greater than that of on-demand routing</td>
</tr>
<tr>
<td>Quality of service support</td>
<td>Fair can support QoS, although most support shortest path</td>
<td>Mainly shortest path as the QoS metric</td>
</tr>
</tbody>
</table>