Ad-hoc Routing

Arvind Krishnamurthy
Fall 2003

Ad Hoc Routing

- Create multi-hop connectivity among set of wireless, possibly moving, nodes
- Mobile, wireless hosts act as forwarding nodes as well as end systems
- Need routing protocol to find multi-hop paths
  - Needs to be dynamic to adapt to new routes, movement
  - Low consumption of memory, bandwidth, power
  - Scalable with numbers of nodes
  - Localized effects of link failure

Problems Using DV or LS

- DV protocols may form loops
  - Very wasteful in wireless: bandwidth, power
  - Loop avoidance sometimes complex
- LS protocols: high storage and communication overhead
- More links in wireless (e.g., clusters) - may be redundant → higher protocol overhead
- Convergence may be slower in conventional networks but must be fast in ad-hoc networks and be done without frequent updates

Proposed Protocols

- Dynamic Source Routing (DSR)
  - On demand source route discovery
- Temporally-Ordered Routing Algorithm (TORA)
  - On demand creation of routes based on link-reversal
- Destination-Sequenced Distance Vector (DSDV)
  - DV protocol, destinations advertise sequence number to avoid loops, not on demand
- Ad Hoc On-Demand Distance Vector (AODV)
  - Combination of DSR and DSDV: on demand route discovery with routing

DSR Concepts

- Source routing
  - No need to maintain up-to-date info at intermediate nodes
- On-demand route discovery
  - No need for periodic route advertisements

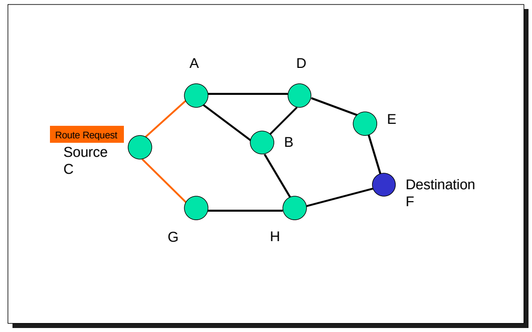
DSR Components

- Route discovery
  - The mechanism by which a sending node obtains a route to destination
- Route maintenance
  - The mechanism by which a sending node detects that the network topology has changed and its route to destination is no longer valid

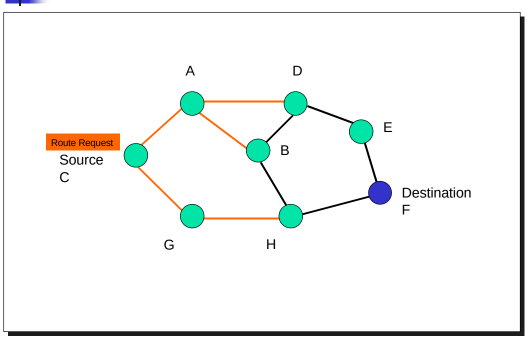
DSR Route Discovery

- Route discovery - basic idea
  - **Source** broadcasts route-request to **Destination**
  - Each node forwards request by adding own address and re-broadcasting
  - Requests propagate outward until:
    - Target is found, or
    - A node that has a route to Destination is found

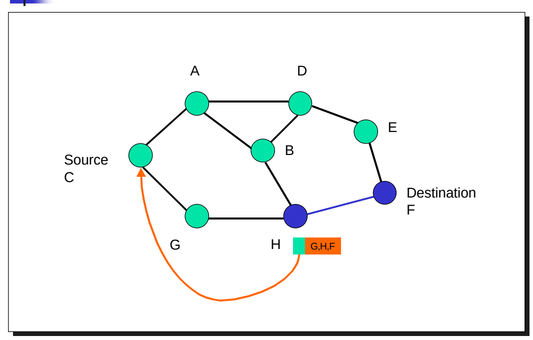
C Broadcasts Route Request to F



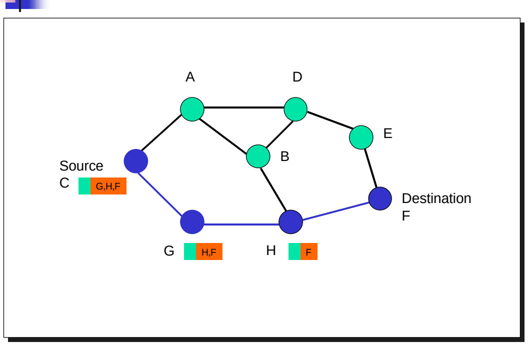
C Broadcasts Route Request to F



H Responds to Route Request



C Transmits a Packet to F



Forwarding Route Requests

- A request is forwarded if:
  - Node is not the destination
  - Node not already listed in recorded source route
  - Node has not seen request with same sequence number
  - IP TTL field may be used to limit scope
- Destination copies route into a Route-reply packet and sends it back to **Source**

Route Cache

- All source routes learned by a node are kept in Route Cache
  - Reduces cost of route discovery
- If intermediate node receives RR for destination and has entry for destination in route cache, it responds to RR and does not propagate RR further
- Nodes overhearing RR/RP may insert routes in cache

Sending Data

- Check cache for route to destination
- If route exists then
  - If reachable in one hop
    - Send packet
  - Else insert routing header to destination and send
- If route does not exist, buffer packet and initiate route discovery

Discussion

- Source routing is good for on demand routes instead of a priori distribution
- Route discovery protocol used to obtain routes on demand
  - Caching used to minimize use of discovery
- Required bidirectional links
  - Route reply sent in the reverse direction traversed by route request
- When a node cannot forward a packet:
  - Examines route cache; if route exists, replaces the broken source route with new route and retransmits
  - If no route, packet is dropped (and does not begin route discovery)
- Promiscuous mode optimization:
  - Node overhears a packet, packet not addressed to node, but node's identity is in the source route ➔ sends a gratuitous route reply to source
  - Cache all route replies broadcast to other nodes

Destination Sequenced Distance Vector

- Referred to DSDV
- Each node maintains a "next hop" for each reachable destination (just as in Distance vector)
- DSDV tags each route with a sequence number
- Considers routes with newer sequence numbers as more accurate information
- Each node advertises:
  - Monotonically increasing even-numbered sequence number
  - Propagated through the system
- When a node realizes that its route to a destination has broken:
  - It advertises the route to D with an infinite metric and a sequence number which is one greater than the previous route
  - Causes other nodes that route through this node to incorporate this new information
  - Routes restored when the destination advertises the next sequence number

Ad-Hoc On-Demand Distance Vector

- Combination of DSR and DSDV
  - On-demand mechanism of route discovery and route-maintenance from DSR
  - Plus the hop-by-hop routing, sequence numbers and periodic beacons from DSDV
- Nodes periodically ping their neighbors:
  - When a link is detected to be broken, upstream node is notified
  - Upon receipt of such a route reply, a node must reinitiate route discovery

Temporally Ordered Routing Algorithm (TORA)

- Quick route discovery on demand
- Localized adaptation to reduce overhead at the expense of longer routes
- Distributed protocol (logically per destination):
  - Query for destination
  - Node with a route returns "height" in Update
  - Nodes set their height above that of the neighbor from which Update was received
  - When route becomes invalid, broadcast update with local maximum to that destination
- "Link Reversal" Algorithms
  - When a node has no downstream links it reverses the direction of one or more links
  - Links are directed based on a metric maintained by the nodes in the network
    - Can be conceptually viewed as a "height" (i.e., link directed from the higher node to the lower node)

Three Basic Functions

- Creating Routes: demand driven “query/reply”
  - A query packet (QRY) is flooded through network
  - An update packet (UPD) propagates back if routes exist
- Maintaining routes: “link reversal” algorithm
  - UPD packet re-orient the route source
- Erasing routes:
  - A clear packet (CLR) is flooded through network to erase invalid routes

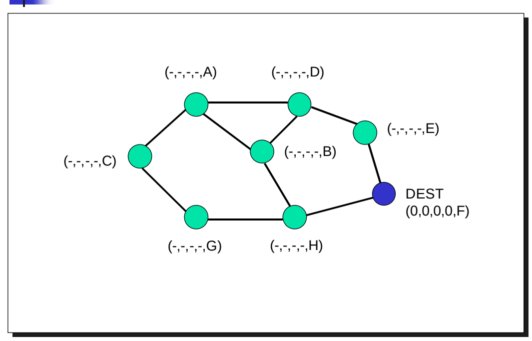
“Height” Metric

- Each node I contains a “height” – a quintuple (τ, *oid*, *r*, δ, *i*) where:
  - τ : the “logical time” of a link failure
  - *oid* : the unique ID of the node that defined the reference level
  - *r* : a “reflection” indication bit
  - δ : a “propagation” ordering parameter
  - *i* : the unique ID of the node

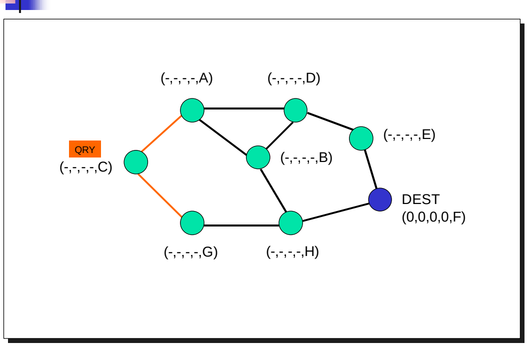
Link Direction Assignment

- Link direction assignment based on height of node *i* and height of corresponding neighbor *j*
  - if H_j == NULL then Unassigned
  - else if H_i == NULL then Down
  - else if $H_i > H_j$ then Down
  - else if $H_i < H_j$ then Up
  - The ordering of non-null heights always forms a Directed Acyclic Graph (DAG)

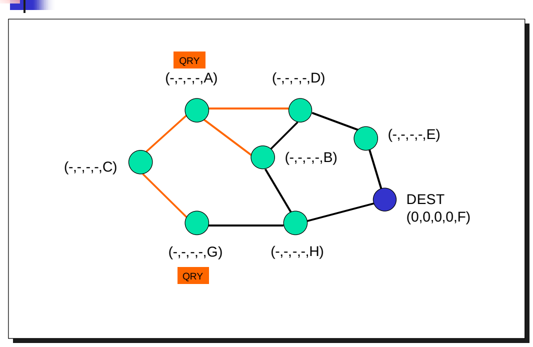
E.g. Creating Routes Process



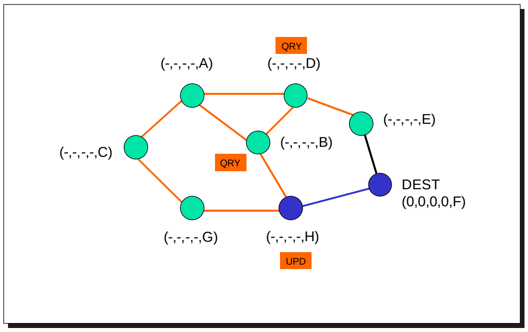
Node C Requires a Route; Sends QRY



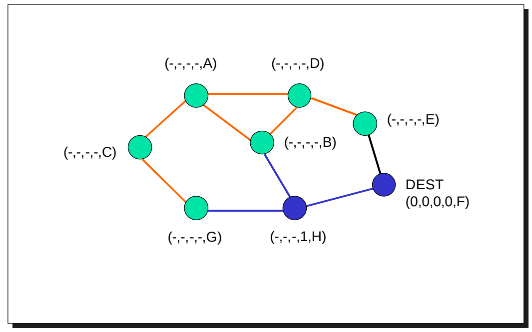
Nodes A and G send Query



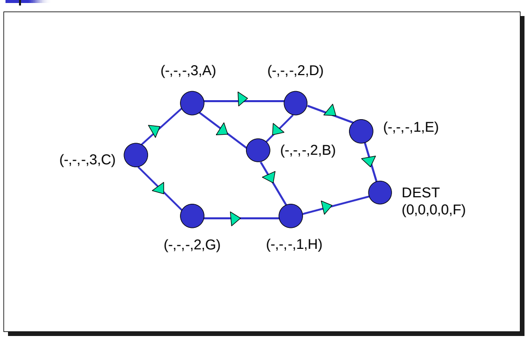
Nodes B & D send Query; H Responds



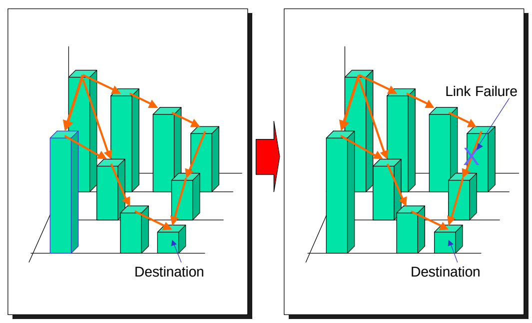
Nodes B & G Receive UPD



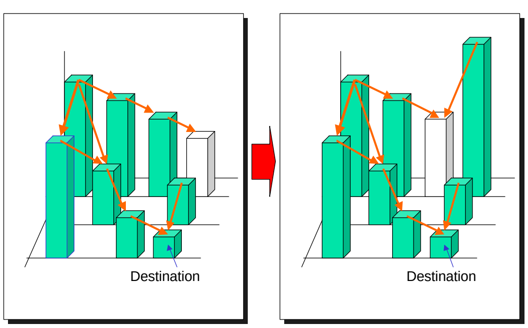
Create Route Process Complete



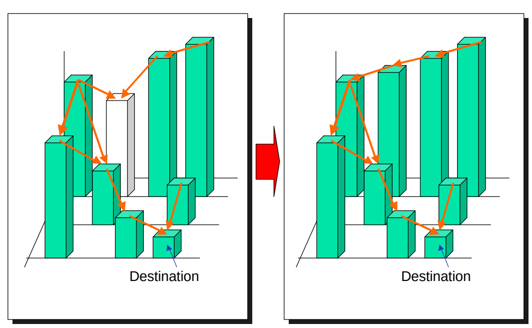
Conceptual Description



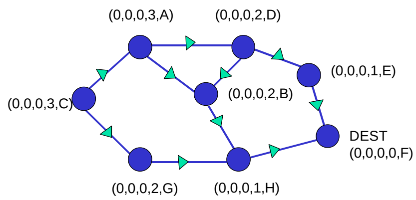
Conceptual Description



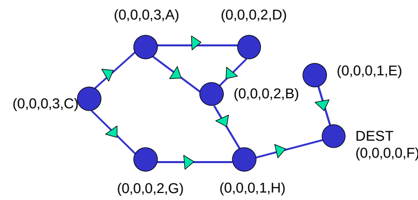
Conceptual Description



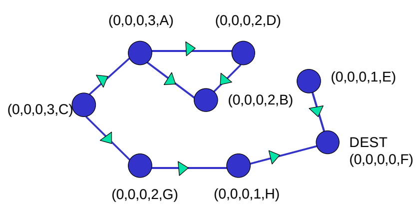
Link Failure 1



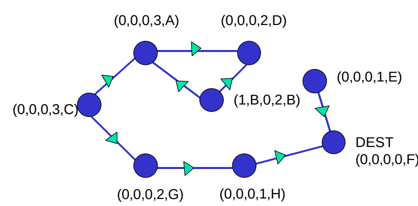
Link Failure 1



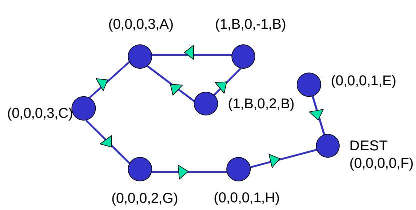
Link Failure 2



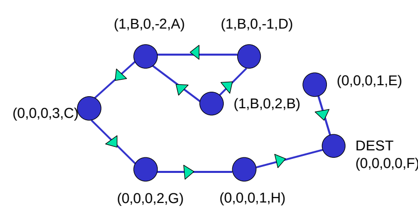
Link Failure 2



Link Failure 2



Link Failure 2



Simulation Results

- All protocols do fine at low mobility
- Packet delivery ratio: poor for DSDV somewhat less for Tora when small pause time
- Routing overhead high for TORA and AODV when small pause times
- TORA and AODV have more suboptimal routes

Packet Delivery Ratio

- DSR and AODV deliver 95-100% of packets
- DSDV fails to converge for high mobility
  - stale routes
- TORA loses packets due to short-lived loops, created during link reversal

Routing Overhead

- DSDV overhead constant, others reduce overhead as mobility decreases
- TORA sometimes suffers from congestion collapse with large number of sources
- AODV vs. DSR
  - Overhead lower for DSR when measuring packets
  - When measuring bytes, AODV performs better for lower mobility rates
  - Cost of sending packets vs cost of sending bytes? Need to bring MAC layer overhead into account

Path Optimality

- DSDV and DSR close to optimal regardless of mobility
- TORA and AODV do well for low mobility