EE5132 / EE5023

Routing Protocols

Chapter 3

Routing

- Routing involves determining a path from a source node to a destination node through the network based on criteria such as
  - minimum no. of hops
  - least cost, etc.
- There are two main families of routing algorithms:
  - **Link state routing**, based on Dijkstra's algorithm
  - **Distance-vector routing**, based on the Bellman-Ford algorithm
- **Overview of Lecture**
- Routing Concepts
- Dijkstra's Algorithm
- Bellman-Ford Algorithm
- Infrastructure-free or Ad Hoc Networks
- Routing in Mobile Ad Hoc Networks (MANETs)
  - Table-driven Routing Protocols
    - Destination Sequenced Distance-Vector (DSDV)
  - Source Initiated On-demand Routing Protocols
    - Dynamic Source Routing Protocol (DSR)
    - Ad hoc On-demand Distance Vector Routing (AODV)
  - Hybrid Routing Protocol
    - Zone Routing Protocol (ZRP)

Routing Concepts

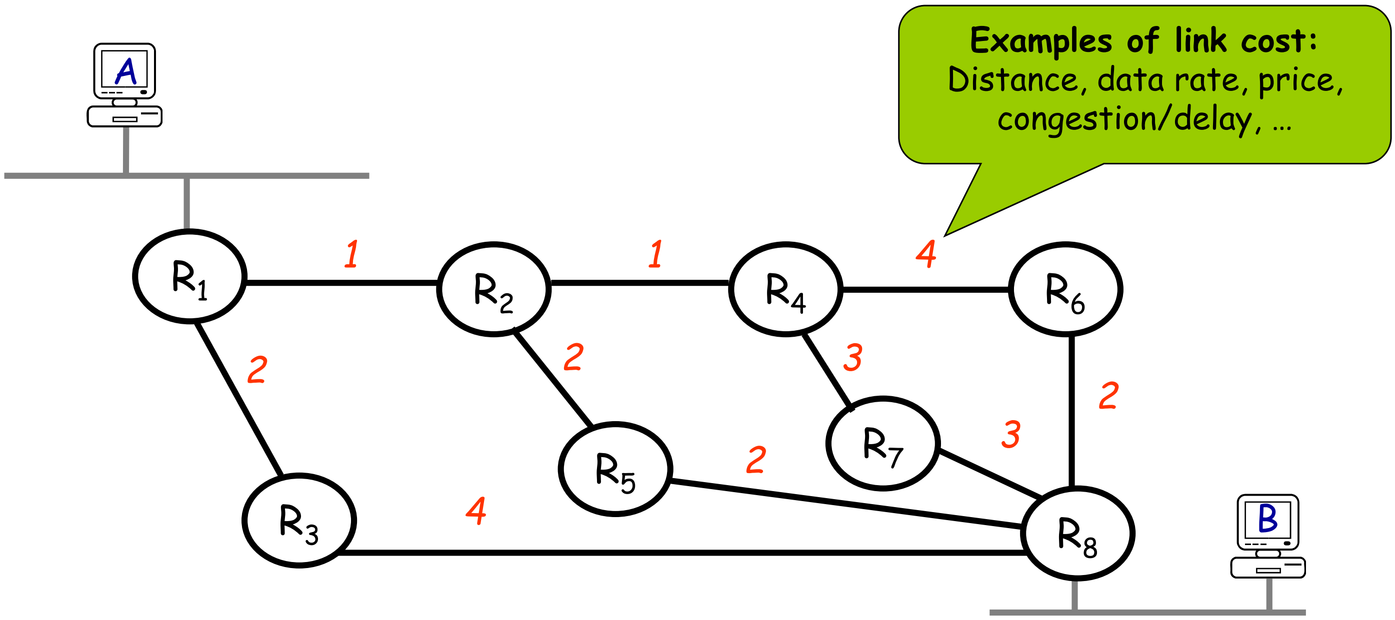
- Performance criteria used for selection of route:
  - Minimum hop
  - Least cost
- To select a path involves 2 issues:
  - The path selection algorithm itself
    - The cost of a path is a function of: *hop count* and *available bandwidth*. Each interface has associated a metric which indicates the amount of remaining available bandwidth.
    - This metric is combined with the hop count to provide a cost value, whose goal is to *pick the path with the minimum number of hops supporting the requested bandwidth*.
    - When several paths are available, the path whose availability is maximal is selected. This way the balance load is maximized. Observe that this algorithm has double objective optimization.
  - When the algorithm is actually invoked. 2 options:
    - On-demand, i.e. computation is triggered for each new request. Could be computationally expensive
    - Using some form of pre-computation. This option amortizes computational cost over multiple requests, but each computation instance is usually more expensive because paths to all destination should be recomputed. Also the final accuracy of the selected path may be lower. Another important issue is when pre-computation should take place. There are two options:
      - Periodic re-computation
      - Pre-computation after 'N' number of updates have been received

Least Cost Algorithms

- Basis for routing decisions
  - Can minimize hop with each link cost
  - Can have link cost inversely proportional to capacity
- Given network of nodes connected by bi-directional links
- Each link has a cost in each direction
- Define cost of path between two nodes as sum of costs of links traversed
- For each pair of nodes, find a path with the least cost
- Link costs in different directions may be different

Example Network

Objective: Determine the route from A to B that minimizes the path cost.



Dijkstra's Algorithm Definitions

- Find shortest paths from given source node to all other nodes, by developing paths in order of increasing path length
- *N* = set of nodes in the network
- *s* = source node
- *T* = set of nodes so far incorporated by the algorithm
- $w(i, j)$ = link cost from node *i* to node *j*
  - $w(i, i) = 0$
  - $w(i, j) = \infty$ if the two nodes are not directly connected
  - $w(i, j) \geq 0$ if the two nodes are directly connected
- $L(n)$ = cost of least-cost path from node *s* to node *n* currently known
  - At termination, $L(n)$ is cost of least-cost path from *s* to *n*

Edsger W. Dijkstra

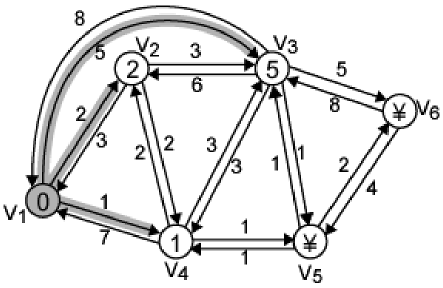
Dijkstra's Algorithm Method

- Step 1 [Initialization]
  - $T = \{s\}$ Set of nodes so far incorporated consists of only source node
  - $L(n) = w(s, n)$ for $n \neq s$
  - Initial path costs to neighboring nodes are simply link costs
- Step 2 [Get Next Node]
  - Find neighboring node x not in T with least-cost path from s
  - Incorporate node x into T
  - Also incorporate the edge that is incident on node x and a node in T that contributes to the path
- Step 3 [Update Least-Cost Paths]
  - $L(n) = \min[L(n), L(x) + w(x, n)]$ for all $n \notin T$
  - If latter term is minimum, path from s to n is path from s to x concatenated with edge from x to n
- Algorithm terminates when all nodes have been added to T

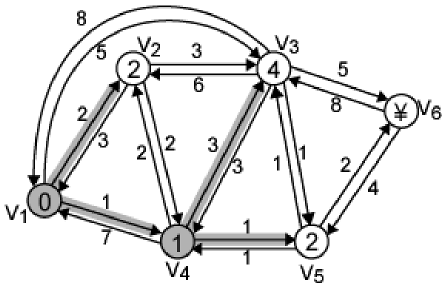
Dijkstra's Algorithm Notes

- At termination, value $L(x)$ associated with each node x is cost (length) of least-cost path from s to x.
- In addition, T defines least-cost path from s to each other node
- One iteration of steps 2 and 3 adds one new node to T
  - Defines least cost path from s to that node

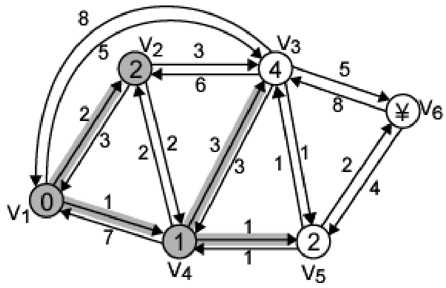
Example of Dijkstra’s Algorithm



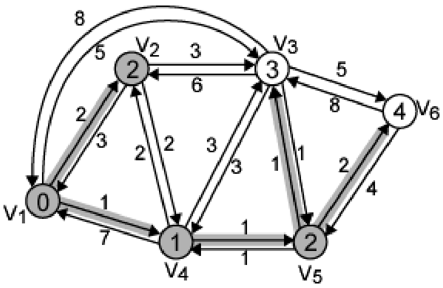
T = {1}



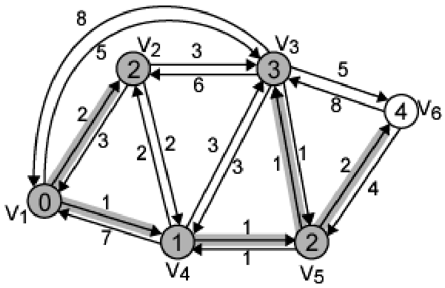
T = {1, 4}



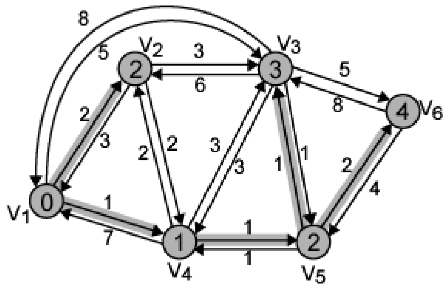
T = {1, 2, 4}



T = {1, 2, 4, 5}



T = {1, 2, 3, 4, 5}



T = {1, 2, 3, 4, 5, 6}

Results of Example Dijkstra’s Algorithm

Iteration	*T*	*L*(2)	Path	*L*(3)	Path	*L*(4)	Path	*L*(5)	Path	*L*(6)	Path
1	{1}	2	1–2	5	1-3	1	1–4	∞	-	∞	-
2	{1,4}	2	1–2	4	1-4-3	1	1–4	2	1-4–5	∞	-
3	{1, 2, 4}	2	1–2	4	1-4-3	1	1–4	2	1-4–5	∞	-
4	{1, 2, 4, 5}	2	1–2	3	1-4-5–3	1	1–4	2	1-4–5	4	1-4-5–6
5	{1, 2, 3, 4, 5}	2	1–2	3	1-4-5–3	1	1–4	2	1-4–5	4	1-4-5–6
6	{1, 2, 3, 4, 5, 6}	2	1-2	3	1-4-5-3	1	1-4	2	1-4–5	4	1-4-5-6

See also:
http://weierstrass.is.tokushima-u.ac.jp/ikeda/suuri/dijkstra/Dijkstra.shtml

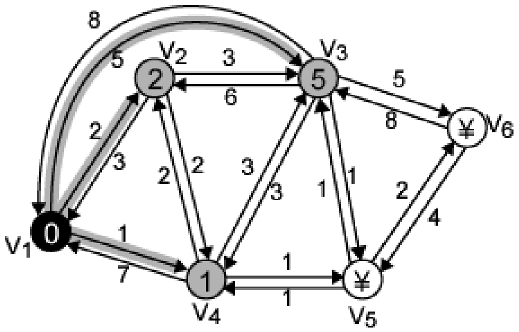
Bellman-Ford Algorithm Definitions

- Find shortest paths from given node, subject to constraint that paths contain at most one link
- Find the shortest paths with a constraint of paths of at most two links
- And so on
- s = source node
- $w(i, j)$ = link cost from node i to node j
  - $w(i, i) = 0$
  - $w(i, j) = \infty$ if the two nodes are not directly connected
  - $w(i, j) \geq 0$ if the two nodes are directly connected
- h = maximum number of links in path at current stage of the algorithm
- $L_h(n)$ = cost of least-cost path from s to n under constraint of no more than h links

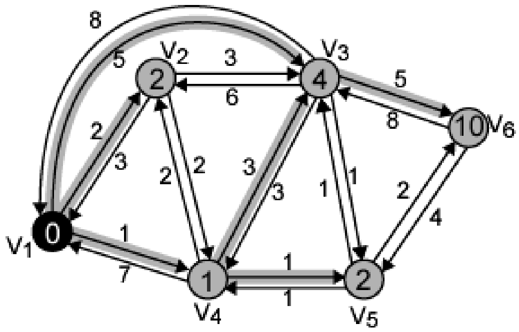
Bellman-Ford Algorithm

- Step 1 [Initialization]
  - $L_0(n) = \infty$, for all $n \neq s$
  - $L_h(s) = 0$, for all h
- Step 2 [Update]
- For each successive $h > 0$
  - For each $n \neq s$, compute
  - $L_h(n)=\min_j[L_{h-1}(j)+w(j,n)]$
- Connect n with predecessor node j that achieves minimum path cost
- Eliminate any connection of n with different predecessor node formed during an earlier iteration
- Path from s to n terminates with link from j to n

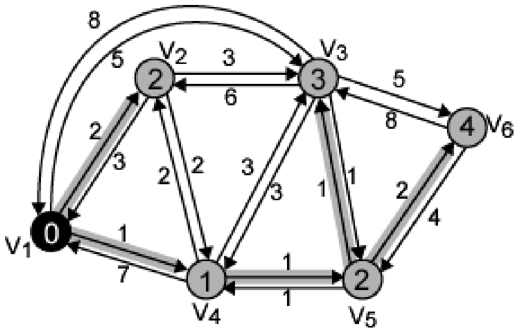
Example of Bellman-Ford Algorithm



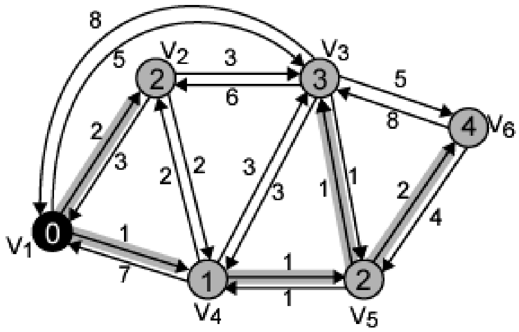
h = 1



h = 2



h = 3



h = 4

Results of Bellman-Ford Example

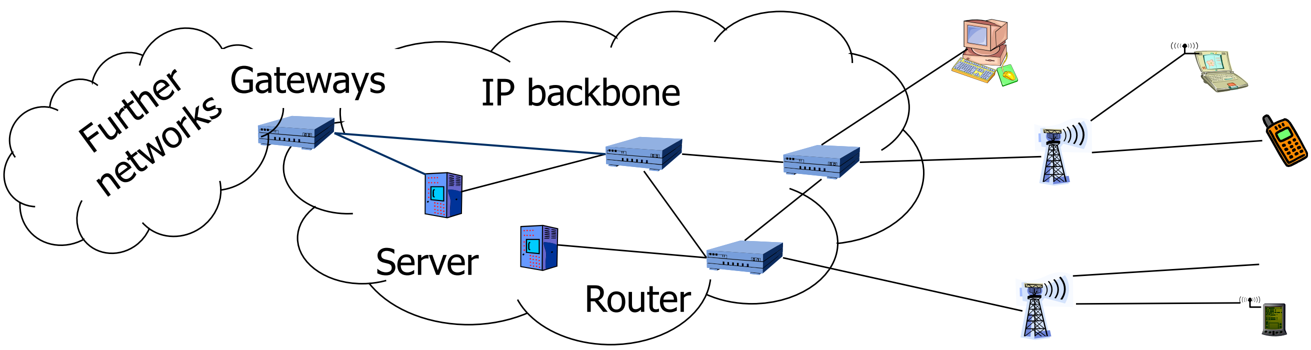
h	$L_h(2)$	Path	$L_h(3)$	Path	$L_h(4)$	Path	$L_h(5)$	Path	$L_h(6)$	Path
0	∞	-	∞	-	∞	-	∞	-	∞	-
1	2	1-2	5	1-3	1	1-4	∞	-	∞	-
2	2	1-2	4	1-4-3	1	1-4	2	1-4-5	10	1-3-6
3	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
4	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6

Comparison between B-F and Dijkstra's algorithms

- Solutions from the two algorithms agree
- **Bellman-Ford algorithm**: [distance-vector protocol]
  - Each node bases its path calculation on knowledge of the cost to all directly connected neighbors plus the advertised costs for routes heard from these neighbors.
  - Easily susceptible to loops because the routers depend on information from neighbors, which might no longer be useful after a failure
  - Has a distributed version that is fairly effective.
- **Dijkstra's algorithm**: [link-state protocol]
  - Requires each node to have complete information about the entire topology.
  - Each router requires more memory to hold the link-state database and more processing capacity to run the algorithm.
  - Used in routing algorithms such as Open Shortest Path First (OSPF) (for wired networks) and Optimized Link State Routing (OLSR) (for mobile ad hoc networks).

Infrastructure-based Wireless Networks

- Typical wireless network: Based on infrastructure
  - e.g., cellular mobile network
  - Base stations connected to a wired backbone network
  - Mobile nodes communicate wirelessly to these base stations
  - Traffic between different mobile entities is relayed by base stations and wired backbone
  - Mobility is supported by switching from one base station to another
  - Backbone infrastructure required for administrative tasks

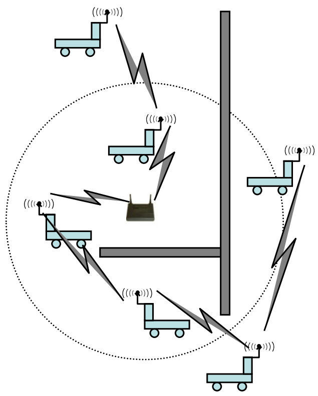


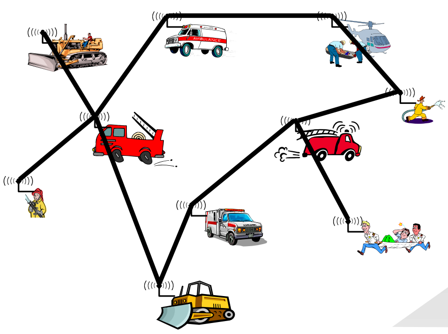
Infrastructure-based Wireless Networks – Limits?

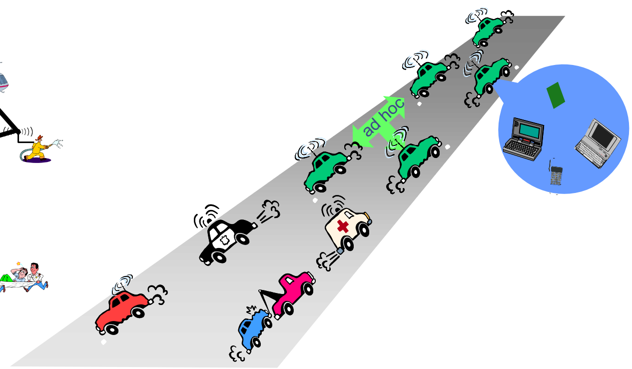
- What if …
  - No infrastructure is available? – e.g., in disaster areas
  - Too expensive/inconvenient to set up? – e.g., in remote, large construction sites
  - There is no time to set it up? – e.g., in military operations

Possible applications for Infrastructure-Free Networks

- Factory floor automation
- Disaster recovery
- Car-to-car communication



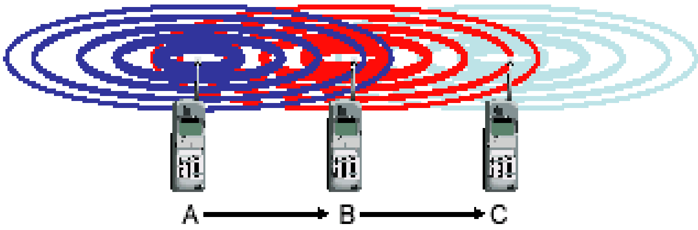




- Military networking: Tanks, soldiers, …
- Finding out empty parking lots in a city, without asking a server
- Search-and-rescue in an avalanche
- Personal area networking (watch, glasses, smartphone, medical appliance, …)
- …

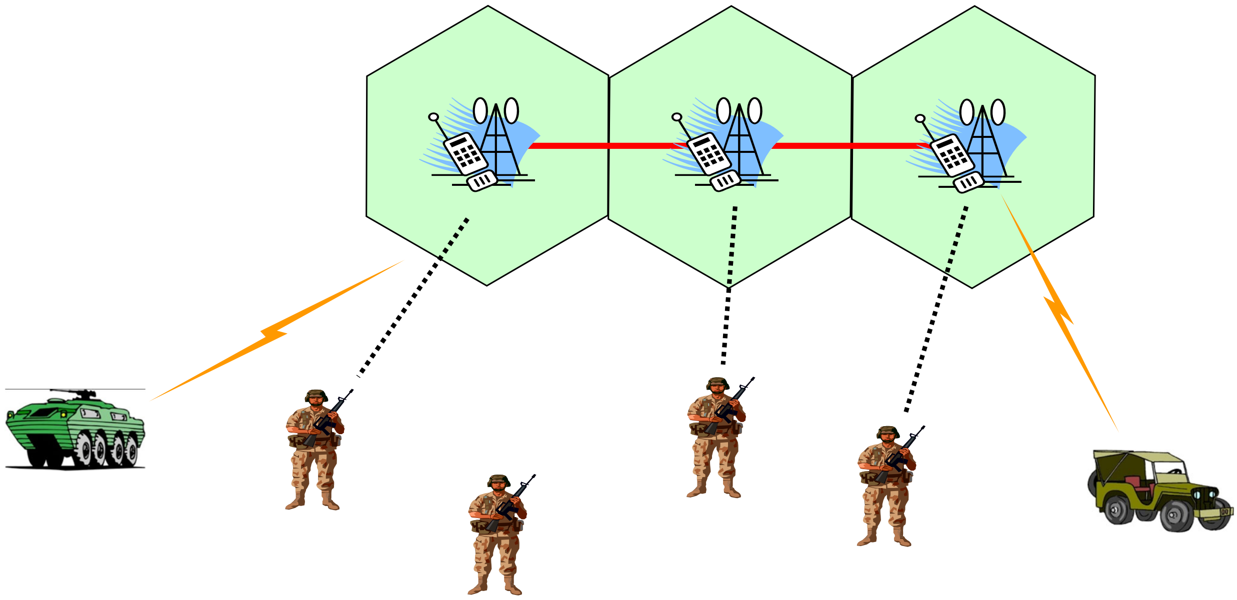
What are Ad Hoc Networks?

- Sometimes there is no infrastructure
  - Automated battlefield, disaster recovery
- Sometimes not every station can hear every other station
  - Data needs to be forwarded in a "multihop" manner



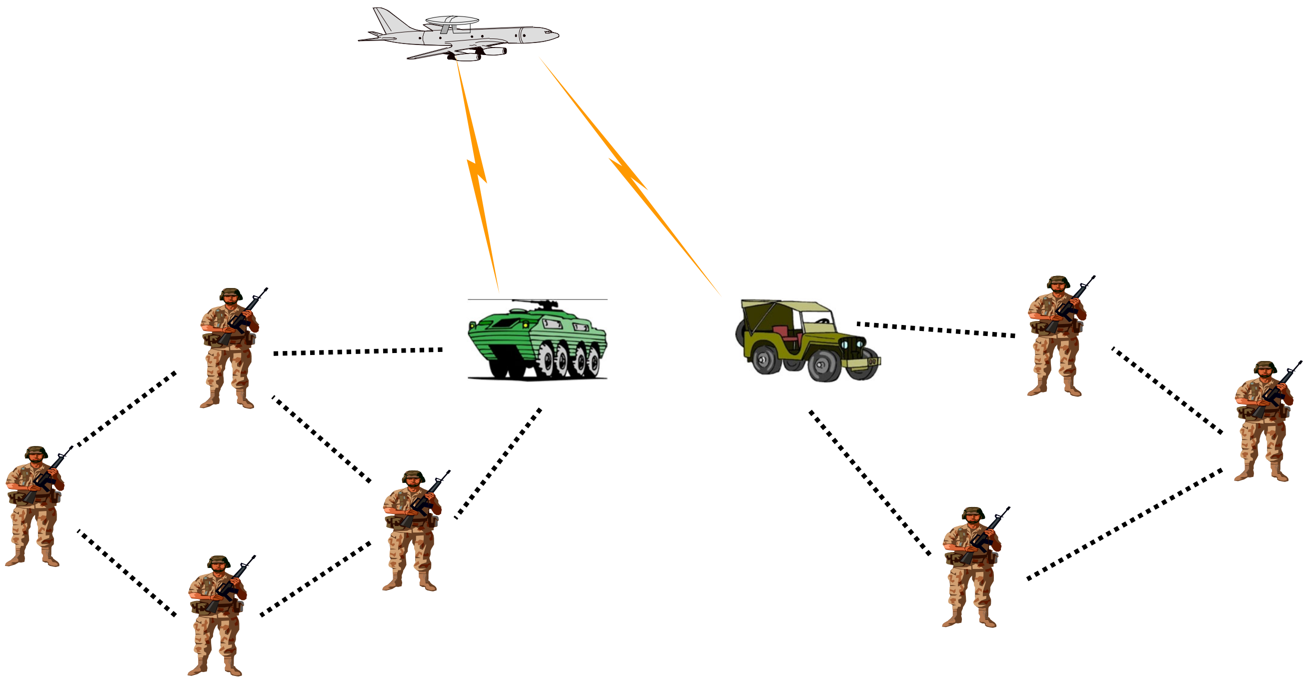
Infrastructure vs. Ad Hoc

Infrastructure Network (cellular or Hot spot)



Infrastructure vs. Ad Hoc

Ad hoc, Multihop Wireless Network



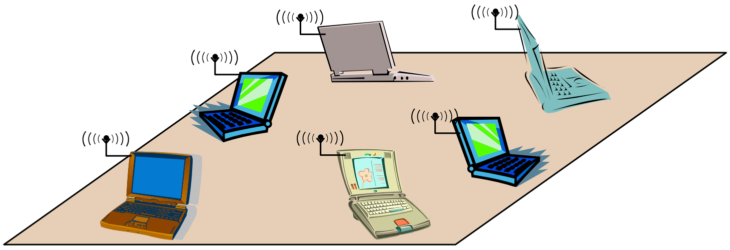
Ad Hoc Networks

- Ad hoc network:
  - A collection of wireless mobile hosts forming a temporary network without the aid of any established infrastructure or centralized administration.
- Significant differences to existing wired networks:
  - Wireless
  - Self-starting
  - No administrator
  - Cannot assume that every node is within communication range of every other node
  - Possibly quite dynamic topology of interconnections

(Wireless) Ad Hoc Networks

- Try to construct a network without infrastructure, using networking abilities of the participants
  - This is an ***ad hoc network*** – a network constructed "for a special purpose"

- Simplest example: Laptops in a conference room – a ***single-hop ad hoc network***



Challenges to Routing in MANETs (Mobile Ad Hoc NETworks)

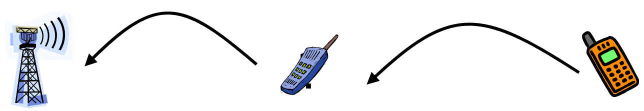
- Lack of a fixed infrastructure
  - Each node in the network must route messages towards their destination
- Nodes operate on battery power
  - Routing of messages may cause faster battery consumption, leading to node going offline
- Dynamic Topology
  - Nodes are constantly moving, leaving, or joining

MANET Protocol Considerations

- Simple, Reliable and Efficient
- Distributed but lightweight in nature
- Quickly adapt to changes in topology and traffic pattern
- Protocol reaction to topology changes should result in minimal control overhead
- Bandwidth efficient
- Mobility Management involving user location management and Hand-off management

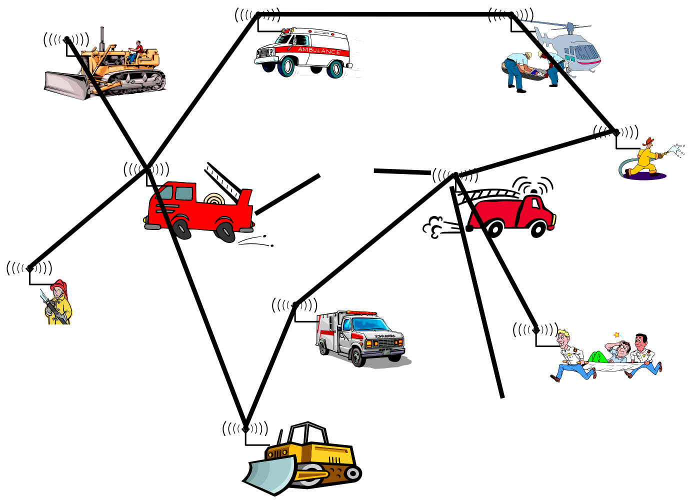
Limited range! Multi-hopping

- For many scenarios, communication with peers outside immediate communication range is required
  - Direct communication limited because of distance, obstacles, …
  - Solution: ***multi-hop network***



Mobility! Suitable, Adaptive Protocols

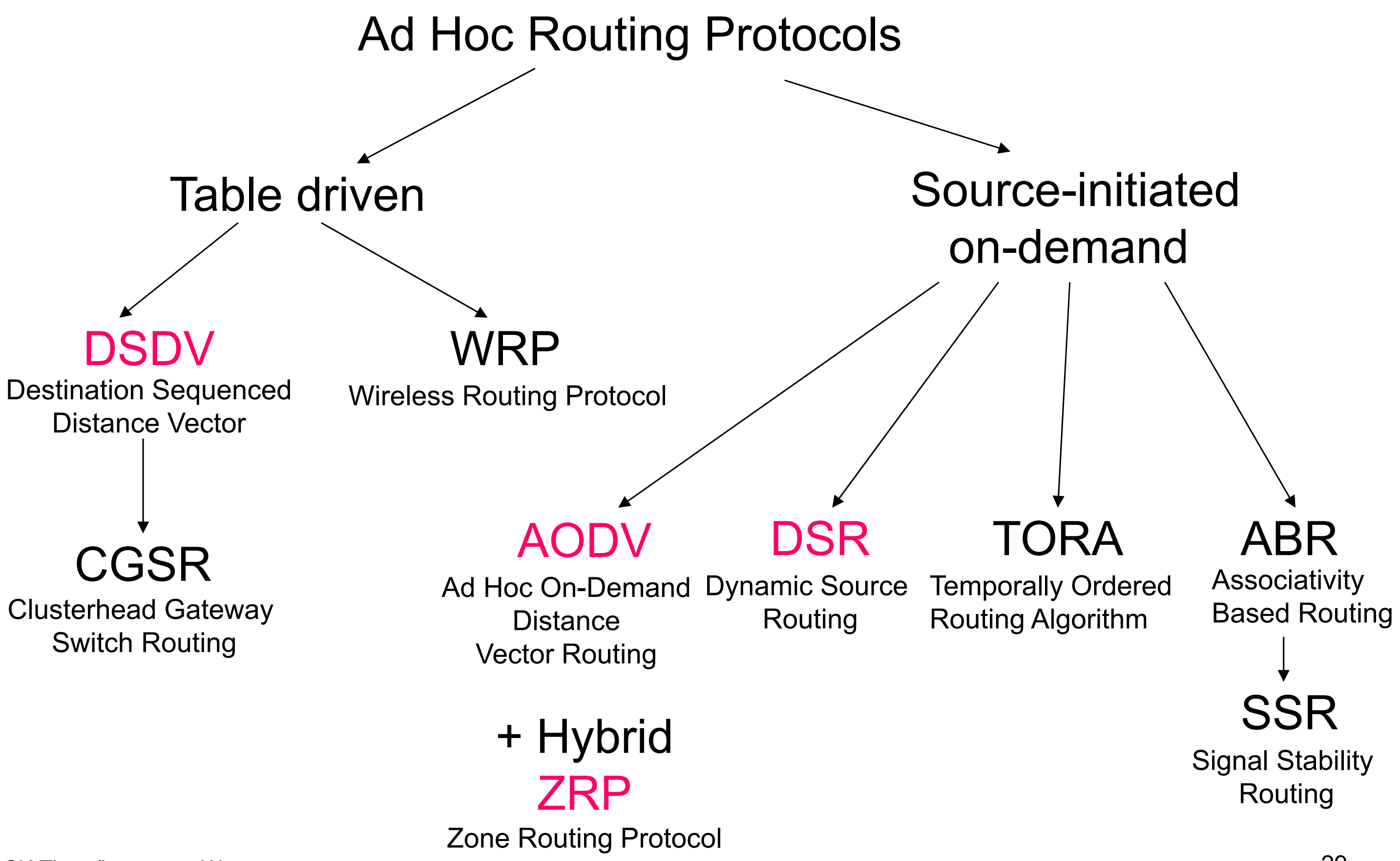
- In many (not all!) ad-hoc network applications, participants move around
  - In cellular network: simply hand over to another base station
- In ***mobile ad hoc networks*** **(*MANET*)**:
  - Mobility changes neighborhood relationships
  - Must be compensated for
  - e.g., routes in the network have to be changed
- Complicated by scale
  - Large number of such nodes difficult to support



Three Categories of Protocols

- **Source Initiated On-Demand Routing Protocols**
  - Reactive
  - Create routes only when desired by the source node
- **Table-driven Routing Protocols**
  - Pro-active
  - Maintain consistent, up-to-date routing information from each node to every other node
- **Hybrid Routing Protocols**
  - Use pro-active protocol in local zone, use reactive protocol between zones

Existing Ad Hoc Routing Protocols



Trade off

- Latency of route discovery
  - Proactive protocols may have lower latency since routes are maintained at all times
  - Reactive protocols may have higher latency because a route from *X* to *Y* will be found only when *X* attempts to send to *Y*
- Overhead of route discovery/maintenance
  - Reactive protocols may have lower overhead since routes are determined only if needed
  - Proactive protocols can result in higher overhead due to continuous route updating

Table-driven Routing Protocols

*

- **Destination Sequenced Distance-Vector (DSDV)**
  CE Perkins and P Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers", SIGCOMM '94, London, UK, 31 Aug – 2 Sep 1994, pp 234-244.
- **Wireless Routing Protocol (WRP)**
  S Murthy and JJ Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks", Journal of Mobile Networks and Applications, vol. 1, no. 2, Oct 1996, pp 183-197.
- **Clusterhead Gateway Switch Routing (CGSR)**
  CC Chiang, HK Wu, W Liu and M Gerla, "Routing in clustered multihop, mobile wireless networks with fading channel", IEEE SICON, 1997.

Distance Vector (Distributed Bellman-Ford)

- Link costs may change over time
  - changes in traffic conditions
  - link failures
  - mobility
- Each node maintains its own routing table
  - need to update table regularly to reflect changes in network
- Let D_i be the shortest distance from node *i* to a destination node
- *note*: there are usually a number of destinations, do separate computation for each destination

$$D_i = \min_j [\, d_{ij} + D_j \,] \text{ : update equation}$$

- Each node *i* regularly updates the values of D_i using the update equation
  - each node maintains the values of d_{ij} to its neighbors, as well as values of D_j received from its neighbors
  - uses those to compute D_i and send new value of D_i to its neighbors
  - if no changes occur in the network, algorithm will converge to shortest paths in no more than *N* steps, when *N* is the no. of nodes

Destination Sequenced Distance-Vector (DSDV)

- Based on the Distance Vector (Distributed Bellman-Ford) algorithm
  - modified to work better in MANET
  - produces loop-free fewest hops path
- Each node maintains a routing table containing:
  - next hop towards each destination
  - a cost metric (distance in no. of hops) for the path to each destination
  - a destination sequence number that is created by the destination (sequence numbers help to distinguish new routes from stale ones, thereby avoiding formation of loops)
- Each node periodically forwards its routing table to its neighbors
  - each node increments and appends its own sequence number when sending its local routing table
  - this sequence number will be attached to route entries created for this node

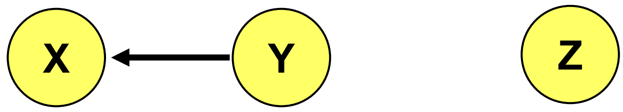
Refer to DSDV paper
(e.g. MH_1)

Destination Sequenced Distance-Vector (DSDV)

- To minimize the routing updates:
  - Either **full dump** carrying all available routing information, or
  - Smaller **incremental packets** containing the change in information since last full dump
- A route broadcast contains:
  - Destination address
  - Number of hops required to reach destination
  - Sequence number of information received about the destination
- Updates are made when the received sequence number of a destination node is larger than current table entry, or if same, the metric is better (e.g., smaller hop count)

Destination Sequenced Distance-Vector (DSDV)

- Assume that X receives routing information from Y about a route to Z



  - S(X): destination sequence # for node Z as stored at node X
  - S(Y): destination sequence # for node Z sent by Y with its routing table to node X

- Node X takes the following steps:
  - If S(X) > S(Y), then X ignores the routing information received from Y
  - If S(X) = S(Y), and cost of going through Y is smaller than the route known to X, then X sets Y as the next hop to Z
  - If S(X) < S(Y), then X sets Y as the next hop to Z, and S(X) is updated to equal S(Y)

Source Initiated On-demand Routing Protocols

- **Ad hoc On-demand Distance Vector Routing (AODV)**
  C Perkins, E Belding-Royer, and S Das, "Ad hoc On-Demand Distance Vector (AODV) Routing, RFC 3581, Jul 2003.
- **Dynamic Source Routing Protocol (DSR)**
  DB Johnson and DA Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks", Mobile Computing, 1996, pp 153-181.
- **Temporally Ordered Routing Algorithm (TORA)**
  VD Park and MS Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks", INFOCOM 1997, 7-11 Apr 1997.
- **Associativity Based Routing (ABR)**
  CK Toh, "Associativity-Based Routing for Ad Hoc Mobile Networks", Wireless Personal Communications, vol. 4, no. 2, Mar 1997, pp 103-139.
- **Signal Stability Routing (SSR)**
  R Dube, CD Rais, KY Wang and S Tripathi, "Signal stability-based adaptive routing (SSA) for ad hoc mobile networks", IEEE Personal Communications, vol. 4, no. 1, Feb 1997, pp 36-45.

What is "on-demand"

- The routes are created when required
- The source has to discover a route to the destination
- The source and intermediate nodes have to maintain a route as long as it is used
- Routes have to be repaired in case of topology changes.

Dynamic Source Routing Protocol (DSR)

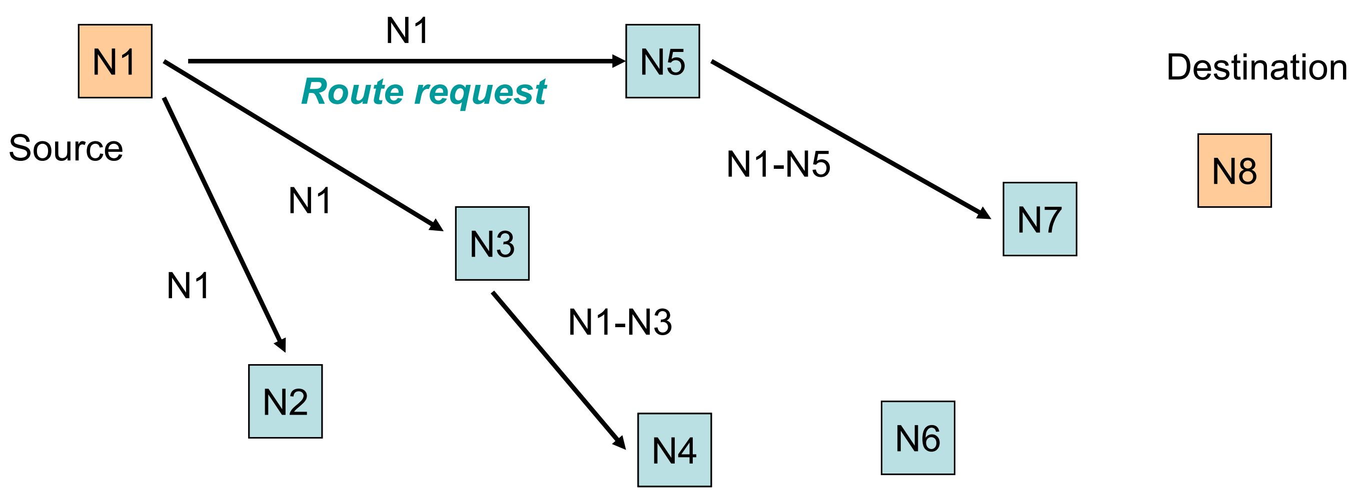
- *Note*: this method was proposed before AODV
- Mobile nodes maintain route caches that contain the source routes of which the mobile is aware
- Consists of two major phases :
  - Route discovery
  - Route maintenance

Route Discovery (DSR)

- Source broadcasts a packet containing address of source and destination
- If the packet has been received before by a mobile node, discard it
- Each node checks whether it knows of a route to the destination
- If it does not, it adds its own address to the packet and then forwards the packet

Route Discovery (DSR)

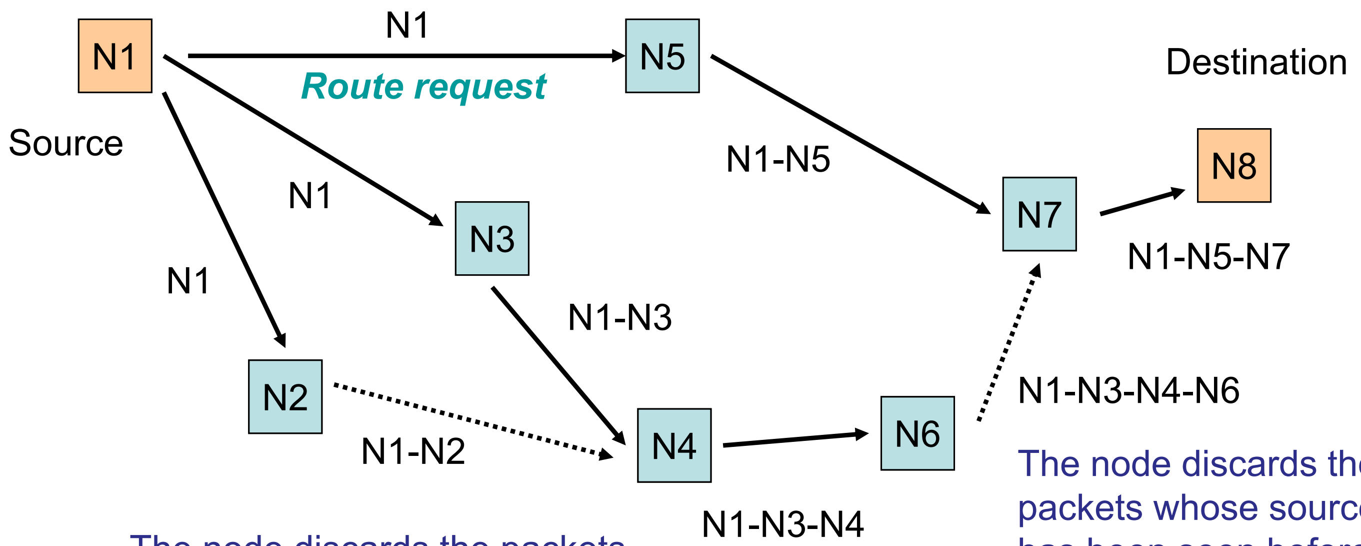
Source broadcasts a packet containing address of source and destination



Each node checks whether it knows of a route to the destination.
If not found, it adds its address into the packet, and forwards it.

Route Discovery (DSR)

Source broadcasts a packet containing address of source and destination

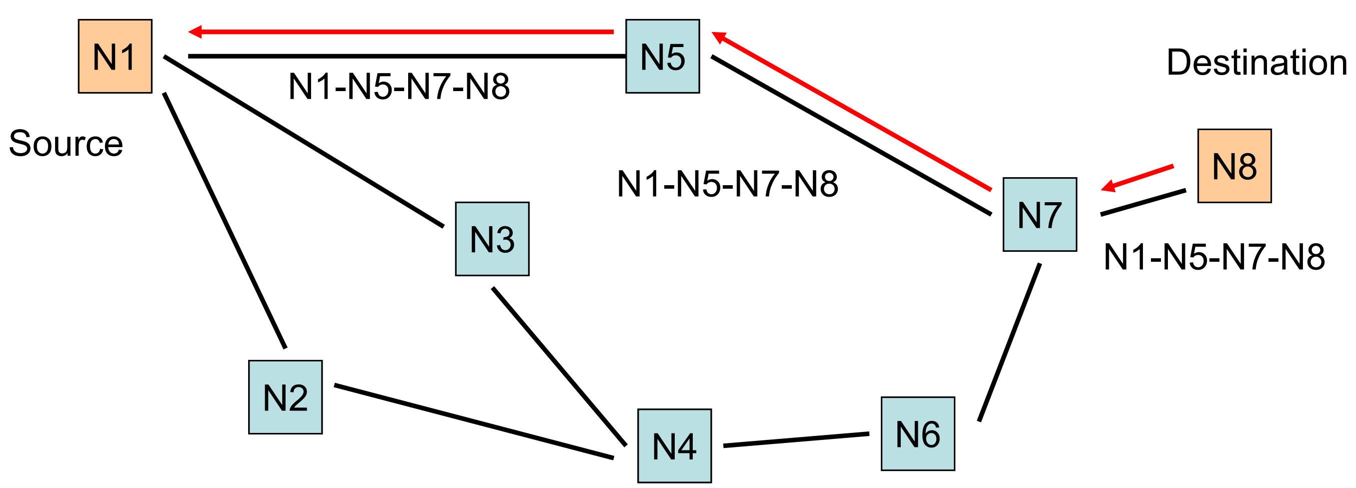


The node discards the packets whose source has been seen before

The node discards the packets whose source has been seen before

Route Reply (DSR)

The destination sends a reply packet to source.



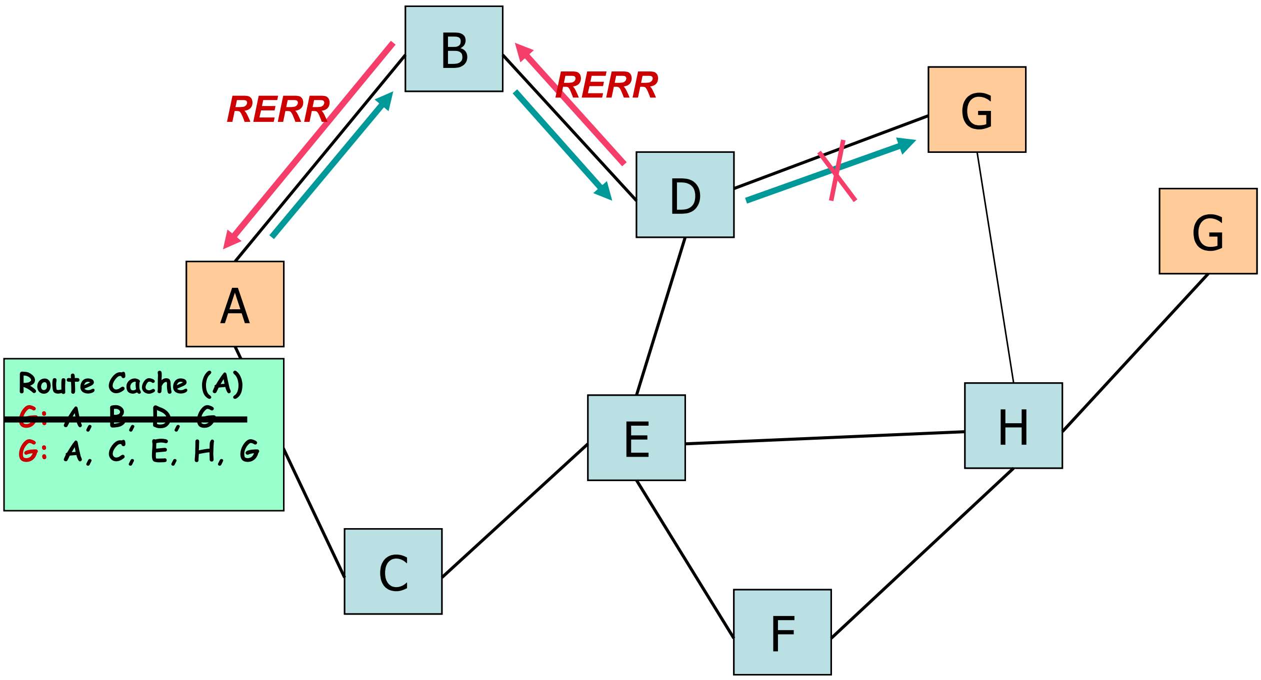
How to Return a Route Reply (DSR)

- If the destination has a route to the source in its route cache, use it
- Otherwise, if symmetric links are supported, reverse the route in the ***route record***
- If symmetric links are not supported, the destination initiates its own route discovery to source

Route Maintenance (DSR)

- 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
- Acknowledgments are used to verify the correct operation of the route links

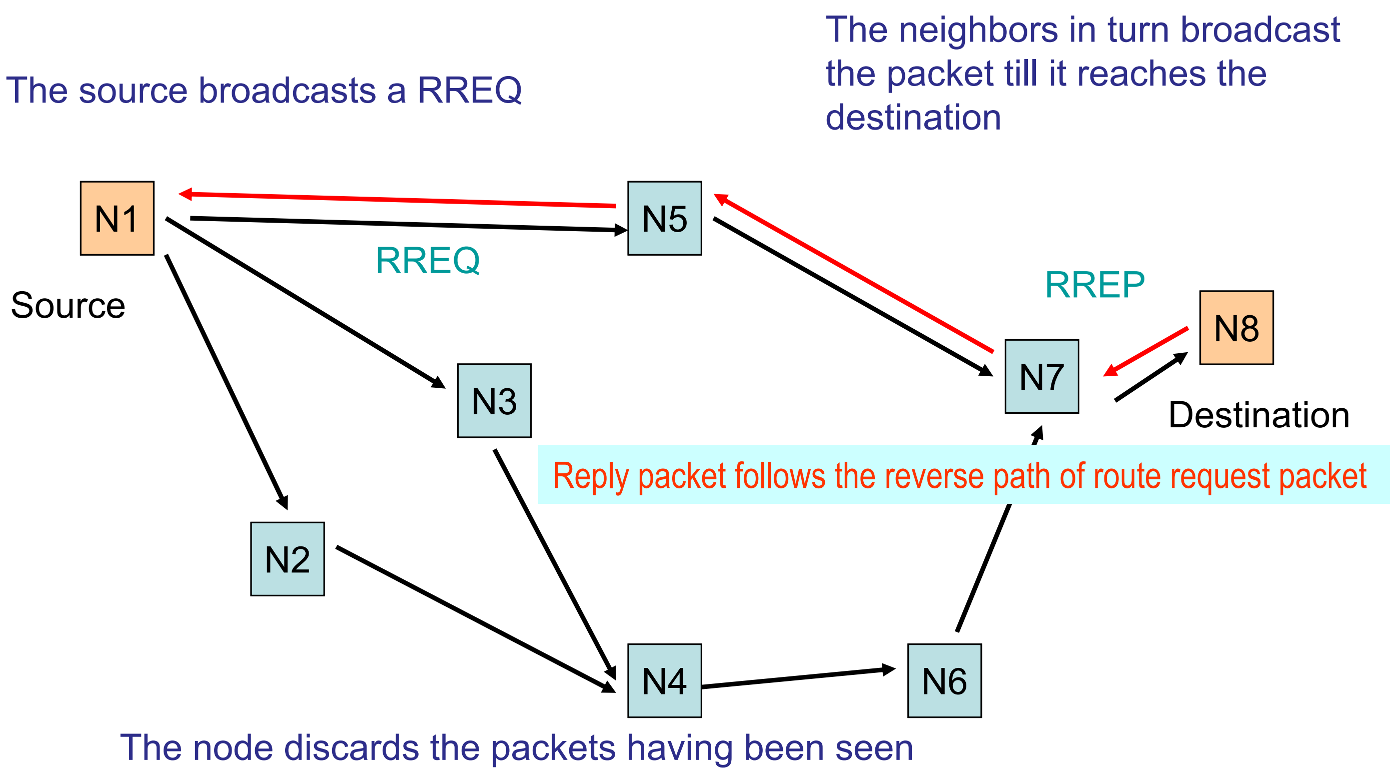
Route Maintenance (DSR)



Ad Hoc On-demand Distance Vector Routing (AODV)

- AODV includes route discovery and route maintenance
- AODV minimizes the number of broadcasts by creating routes on-demand
- AODV only supports the use of symmetric links because a RREP (Route Reply) is forwarded along the path established by a RREQ (Route Request)
- AODV uses ***Hello*** messages to maintain the local connectivity of a node

Route Discovery (AODV)



Refer to AODV paper(s)

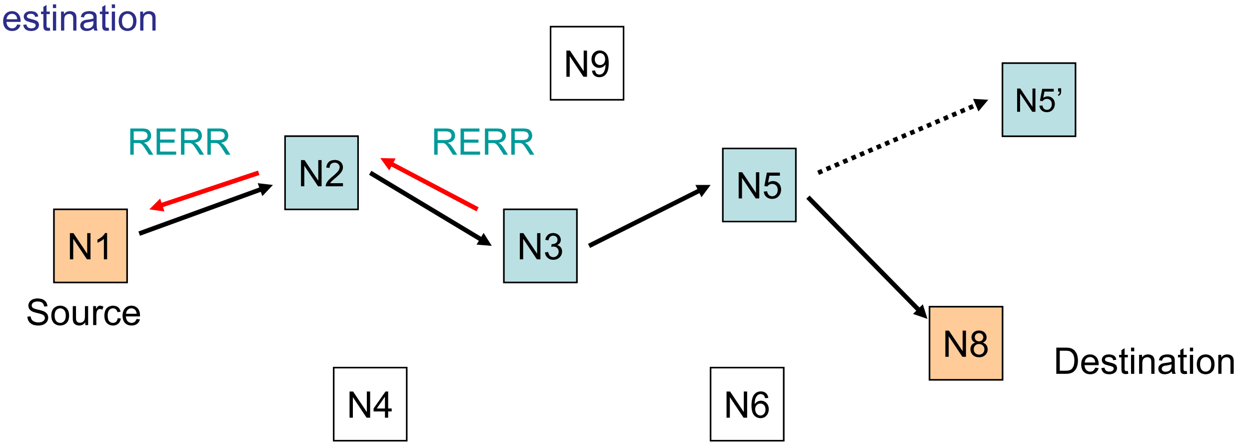
Route Maintenance (AODV)

- If the source node moves, it re-initiates route discovery to establish a new route to the destination.
- If intermediate node moves, its upstream node sends a Route Error (RERR) message to the source.
- When a source receives the RERR, it will reinitiate the route discovery if the route is still needed.

Route Maintenance (AODV)

The original path from the source to the destination

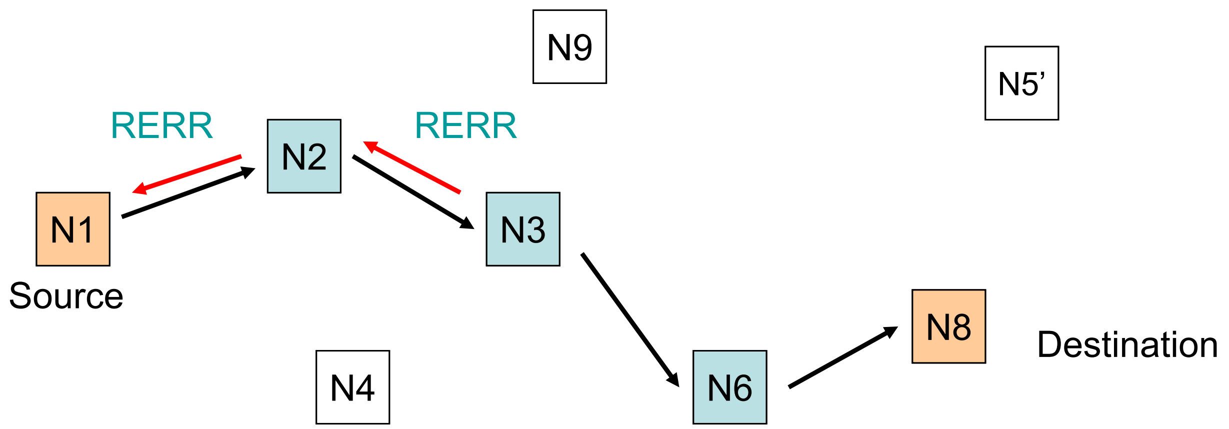
Node N5 then moves to location N5’



Node N3 notices this break and sends a RERR to the source

Route Maintenance (AODV)

On receiving the RERR, the source node re-initiates route discovery



The new route found through node N6

Comparison between DSR and AODV

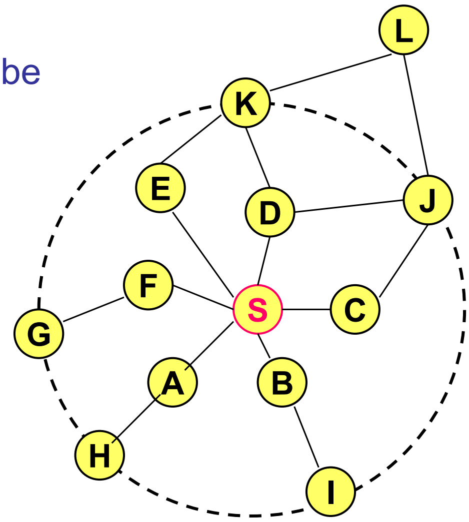
- DSR has less routing overhead than AODV.
- DSR is based on a source routing mechanism, whereas AODV uses a combination of DSR and DSDV mechanisms.
- AODV has better performance than DSR in higher-mobility scenarios.
- DSR has less frequent route discovery processes than AODV
  - after route expires, AODV needs to discover route again

Hybrid Routing Protocol

- **Zonal Routing Protocol (ZRP)**
  ZJ Haas, "A new routing protocol for the reconfigurable wireless networks", Universal Personal Communications, 12-16 Oct 1997.
- **Fisheye State Routing (FSR)**
  G Pei, M Gerla and TW Chen, "Fisheye state routing: a routing scheme for ad hoc networks", IEEE ICC 2000 18-22 Jun 2000.
- **Landmark Routing (LANMAR)**
  G Pei, M Gerla and X Hong, "LANMAR: Landmark routing for large scale wireless ad hoc networks with group mobility", 1st ACM International Symposium on Mobile ad hoc networking and computing, 2000, Boston, MA, USA, pp 11-18.
- **Location-Aided Routing (LAR)**
  YB Ko and NH Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks", Journal on Wireless Networks, vol. 6, no. 4, Jul 2000, pp 307-321.

Zone Routing Protocol (ZRP)

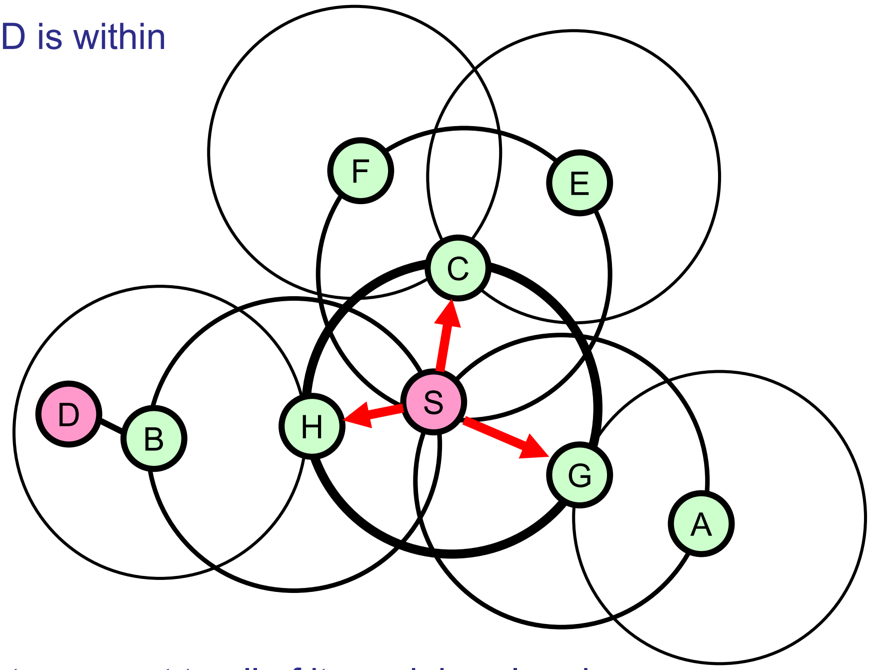
- ZRP combines
  - Proactive protocol: which pro-actively updates network state and maintains route regardless of whether any data traffic exists or not
  - Reactive protocol: which only determines route to a destination if there is some data to be sent to the destination
- All nodes within hop distance at most *d* from a node *X* are said to be in the routing zone of node *X*
- All nodes at hop distance exactly *d* are said to be peripheral nodes of node *X*'s routing zone
- Intra-zone routing: pro-actively maintain state information for links within a short distance from any given node
  - Routes to nodes within short distance are thus maintained proactively.
- Inter-zone routing: use a route discovery protocol for determining routes to far away nodes. Route discovery is similar to DSR with the exception that route requests are propagated via peripheral nodes.



Routing zone of radius 2

Route Discovery (ZRP)

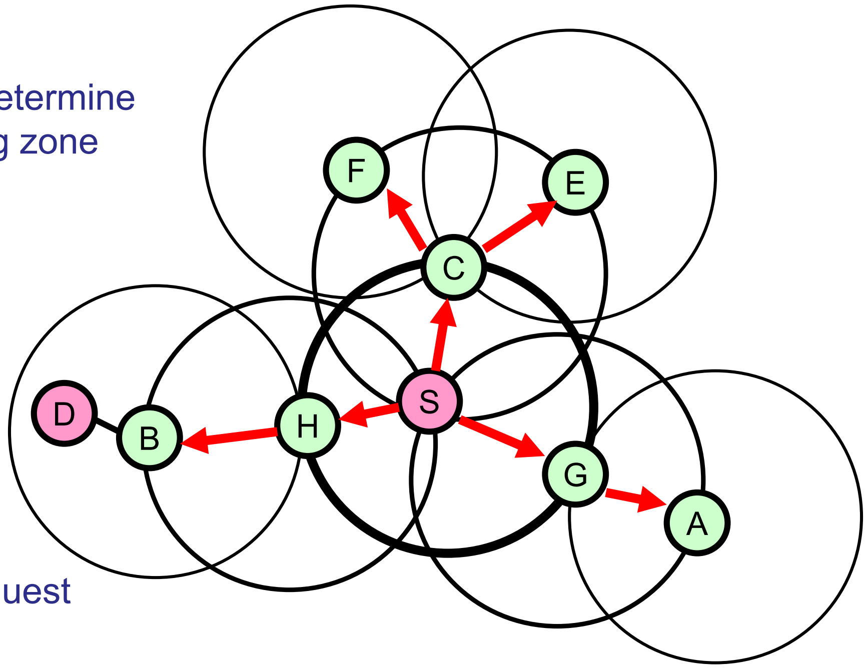
S first check whether D is within its routing zone



If not, S bordercasts a route request to all of its peripheral nodes

Route Discovery (ZRP)

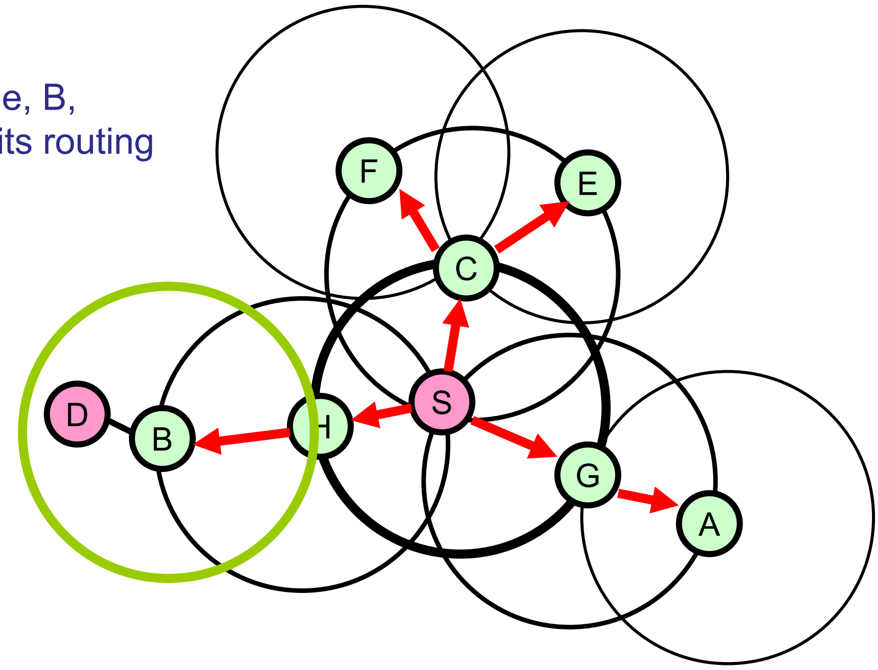
Nodes C, G, and H then determine that D is not in their routing zone



Therefore bordercast the request to ***their*** peripheral nodes

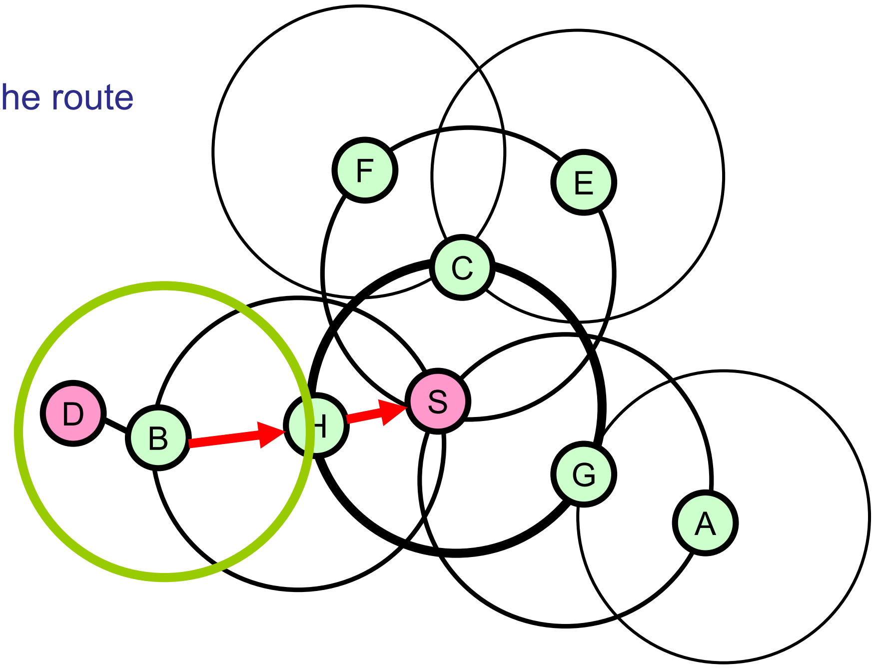
Route Discovery (ZRP)

One of H's peripheral node, B, recognizes D as being in its routing zone



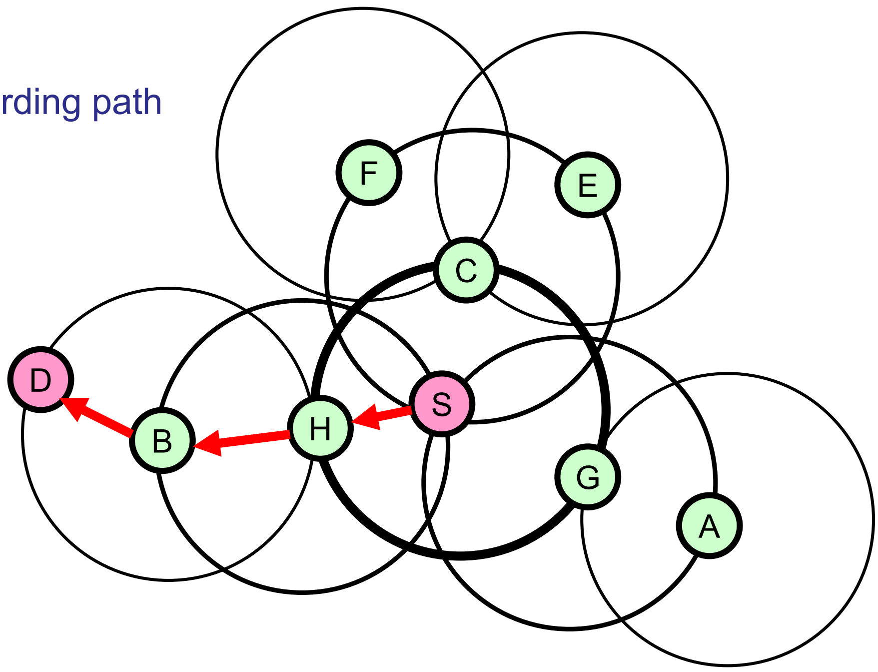
Route Discovery (ZRP)

Node B responds to the route request



Route Discovery (ZRP)

Indicating the forwarding path
S → H → B → D



Protocol Characteristics (1/2)

Routing Protocol	Route Acquisition	Flood for Route Discovery	Delay for Route Discovery	Multipath Capability	Effect of Route Failure
DSDV	Computed a priori	No	No	No	Updates the routing tables of all nodes
WRP	Computed a priori	No	No	No	Ultimately, updates the routing tables of all nodes by exchanging MRL between neighbors
DSR	On-demand, only when needed	Yes. Aggressive use of caching may reduce flood	Yes	Not explicitly. The technique of salvaging may quickly restore a route	Route error propagated up to the source to erase invalid path
AODV	On-demand, only when needed	Yes. Controlled use of cache to reduce flood	Yes	No, although recent research indicate viability	Route error propagated up to the source to erase invalid path

Protocol Characteristics (2/2)

Routing Protocol	Route Acquisition	Flood for Route Discovery	Delay for Route Discovery	Multipath Capability	Effect of Route Failure
AODV	On-demand, only when needed	Yes. Controlled use of cache to reduce flood	Yes	No, although recent research indicate viability	Route error propagated up to the source to erase invalid path
TORA	On-demand, only when needed	Basically one for initial route discovery	Yes. Once the DAG is constructed, multiple paths are found	Yes	Error is recovered locally
ZRP	Hybrid	Only outside a source's zone	Only if the destination is outside the source's zone	No	Hybrid of updating nodes' tables within a zone and propagating route error to the source
LAR	On-demand, only when needed	Reduced by using location information	Yes	No	Route error propagated up to the source

Scalability

- How large can an ad hoc network grow?
  - No one knows
  - It is safe to say that such a network cannot grow to the size of the Internet
  - Worse performance for route acquisition latency

Quality of Service

- Can delay-constrained applications operate well?
  - Vulnerable to dynamic link quality variations between neighboring nodes
  - A route between two endpoints that initially meets the application's QoS constraint may soon fail to meet them

Inter-operation with the Internet

- How can an ad hoc network take advantage of evanescent or dynamically changing points of connection to the Internet?
  - Through the Internet "gateway", a node has connectivity to the global Internet
  - Advertise itself as a default router
  - Other nodes can consider themselves connected to the default router by way of a multi-hop path

The End

Questions?