Ad Hoc Networks — Routing Protocols

- For mobile ad hoc networks, the issue of routing packets between any pair of nodes becomes challenging because the nodes can move randomly within the network.
- Nodes may also join or leave the network and networks themselves may fragment.
- Routes that were considered optimal at a given point in time might not work a few moments later.
- There are two main approaches to ad hoc routing protocol design:
  - Proactive (Table-Driven) protocols
  - Reactive (On-Demand) protocols

Ad Hoc Networks — Routing Protocols- **Proactive**

- **Proactive** schemes compute routes in the background, independent of traffic demands.
- The first type of routing scheme used in early packet radio networks such as the DARPA Packet Radio Network (PRNET) was a proactive distance-vector type protocol. As discussed earlier, this type of protocol is slow to converge and may be susceptible to routing loops.
- While these problems are somewhat overcome by the link-state approach, link-state protocols were primarily designed for wired networks.
- In the link-state approach, global network topology information is maintained in all routers by periodically flooding link-state updates. **Any** link change triggers an immediate global update.

Ad Hoc Networks — Routing Protocols- **Proactive**

- Another drawback of proactive protocols is that they may react to a change in the network topology even if no traffic is affected by the topology change.
- In order to maintain valid routes, the **periodicity** of the flooding of control messages needed to maintain accurate routes to every other node in the network must reflect the dynamics of the network. Therefore, **excessive** overhead may be generated when high mobility triggers frequent updates.
- Even in a network with little data traffic, proactive protocols will use scarce resources such as power and link bandwidth.

Ad Hoc Networks — Routing Protocols -**Reactive**

- An alternative routing approach involves establishing routes **reactively**.
- Routes between nodes are determined only when they are explicitly needed to route data packets.
- There is no updating of every possible route in the network, instead there is a focus on the routes that are being used, or routes that are being set up.
- In protocols that discover routes 'on-demand', small **query/reply** packets are used to discover routes to the final destination.
- Routes may be cached by the source nodes or by intermediate nodes that may overhear the query/reply discussion.

Ad Hoc Networks — Routing Protocols -**Reactive**

- Drawbacks of this approach may be the initial request latency which may degrade the performance of higher layers and applications.
- It is also hard to establish the quality of the paths being set up.
- In general, on-demand routing algorithms are more suited to large networks with light traffic and with low mobility
- As mobility increases, precomputed routes may breakdown, requiring new multiple route discoveries to be undertaken on the way to the destination. This introduces further delays.
- Route caching becomes ineffective in high-mobility networks since routes become stale quickly. **No** updates are propagated throughout a network that is using a reactive protocol when topology changes occur.

Ad Hoc Networks — A **PROACTIVE** Algorithm – Fisheye State Routing

- Fisheye State Routing (**FSR**) is one example of a proactive ad hoc routing protocol.
- It is an attempt to improve on concepts innovated in existing protocols, namely the Global State Routing (**GSR**) protocol and the Destination-Sequenced Distance-Vector (**DSDV**) routing protocol, which are link-state and distance-vector protocols respectively.
- Both GSR and DSDV are based upon the traditional protocols used for wired networks with some minor tweaking for a MANET environment.
- How are they different? How have they been improved?
  - The key difference is in the way routing information is disseminated.
  - In normal link-state routing packets are generated and flooded throughout the network whenever a node detects a topology change.

Ad Hoc Networks — A **PROACTIVE** Algorithm – Fisheye State Routing

  - In both DSDV (distance-vector approach) and GSR (link-state approach) the updates that are sent out by a node contain a **sequence number** or **timestamp**.
  - When updates arrive at other nodes, the receiving node is able to differentiate between the age of the updates coming from each node.
  - If a node has already heard the update from another node, or if it receives a stale update, it can ignore the update and quench it – killing off the flooding process.
- While GSR is better than normal link-state protocols, the large update messages used in GSR waste a considerable amount of network bandwidth. FSR seeks to improve upon GSR.
- Why Fisheye?
  - Seemingly, the eye of a fish captures with high detail the visual information near the centre or focal point, while the detail of the information decreases as the distance from the focal point increases.

Ad Hoc Networks — A **PROACTIVE** Algorithm – Fisheye State Routing

- In FSR, the 'fisheye' approach translates to maintaining accurate distance and path quality information about the immediate neighbourhood of a node, with progressively less detail as the distance increases.
- In effect, each update message does not contain information about all nodes in the network. Instead, it exchanges information about closer nodes more frequently than it does about farther nodes, reducing the overall update message size.
- In a wireless environment, a radio link between mobile nodes may experience frequent disconnects and reconnects:
  - A normal link-state protocol (GSR) would release a link state update for each such change, flooding and possibly congesting the network.
  - FSR avoids this by using periodic updates instead of event-driven updates.

Ad Hoc Networks — A **PROACTIVE** Algorithm – Fisheye State Routing

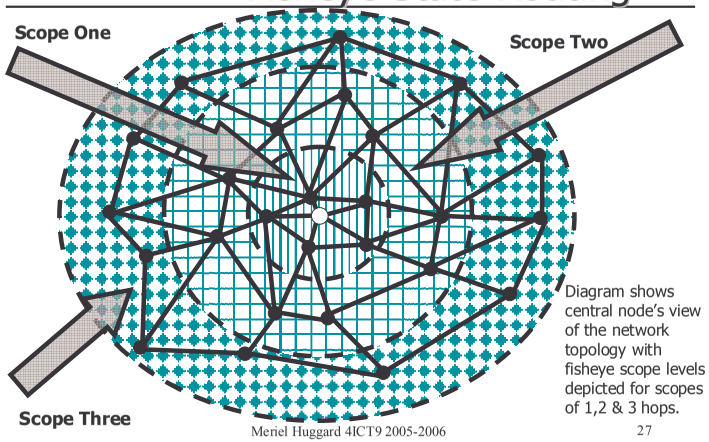


Diagram shows central node's view of the network topology with fisheye scope levels depicted for scopes of 1,2 & 3 hops.

Ad Hoc Networks — A **PROACTIVE** Algorithm – Fisheye State Routing

- FSR also copes well with large networks:
  - Typically in a link-state algorithm, as the network grows, so to do the update messages. FSR attempts to reduce this message size without affecting routing accuracy.
  - How? Firstly, FSR breaks the network down into regions:
    - Remember, link-state algorithms build complete topology maps with the imformation they gather.
    - FSR uses the fisheye technique to define the fisheye scopes with respect to the central node, each routing table entry is now associated with a scope level.
    - The scope is defined as the set of nodes that can be reached within a given number of hops. In the diagram, p.27, there are 3 scopes for 1, 2 and 3 hops respectively.
    - The number and size of scopes will depend on the size and mobility of the network.

Ad Hoc Networks — A **PROACTIVE** Algorithm – Fisheye State Routing

  - Secondly, different exchange periods are used for the different entries in the routing table. More precisely:
    - Link-state information about nodes within a smaller scope are propagated to the neighbours with the highest frequency.
    - The rest of the link-state information, corresponding to nodes in a larger scope, is sent out with decreasing frequency.
    - This has the effect of suppressing a considerable fraction of the link-state entries in a typical update.
- Any drawbacks?
  - While nodes that are nearby will have timely link-state information, the large latencies involved in the updating of 'far' nodes leads to imprecise knowledge about the best path to a distant node.
  - However imprecise information is good enough. Source nodes will have a general idea of the direction of the destination node and as the data gets nearer to the destination node, the routing information will become progressively more accurate.

Ad Hoc Networks — A **REACTIVE** Algorithm – Dynamic Source Routing

- The Dynamic Source Routing (**DSR**) protocol is a source initiated on-demand routing protocol.
- The basic DSR protocol consists of two primary concepts:
  - Route Discovery
  - Route Maintenance
- Each node that is particpating in a DSR enabled network maintains certain data structures:
  - Route Cache
  - Send & Retransmission Buffers
  - Route Request Table
- When a DSR enabled network is booted-up there is **no** routing information anywhere in the system. All information is retrieved on a need-to-know basis. If there is no request to traffic data, then there will be no control messaging unless, and until, such a request is made.

Ad Hoc Networks — A **REACTIVE** Algorithm – Dynamic Source Routing

- How does it operate?
  - When a node wants to send a data packet to a destination, it looks at the routes contained in its route cache, if any routes have been cached:
    - **IF** a route to the destination is in the cache, this route is used.
    - **OTHERWISE** the node initiates a route discovery process for a route to the requested destination.
  - The **Route Discovery** process entails:
    - Broadcasting (flooding) a **route request** packet

Source Address	Destination Address	Unique Identifier

    - Each intermediate node checks whether it knows of a route to the destination.

Source Address	Destination Address	Unique Identifier	+	Hop Address	+	Hop Address

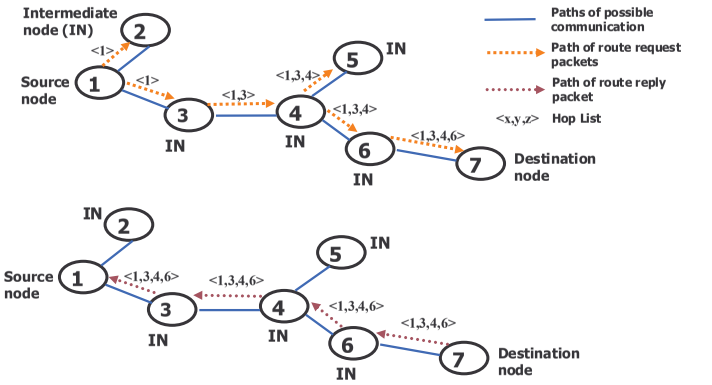
Ad Hoc Networks — A **REACTIVE** Algorithm – Dynamic Source Routing

- It is important to stem the flood of requests and to this end:
  - Nodes do not process route requests that they have recently heard.
  - They also do not process route requests that contain their own address in the existing hop-list.
- Limits are also placed on how far a route request may propagate. This may take the form of:
  - Hop limits
  - Time limits
- A **route reply** packet is generated when either the destination node or an intermediate node with a route to the destination receives a route request packet. A route request arriving at either of these nodes will already contain a record of the sequence of hops taken from the source to this node – called a **hop list.**

Ad Hoc Networks — A **REACTIVE** Algorithm – Dynamic Source Routing

- How is the newly discovered route hop list sent back to the source node?
  - If the destination node or intermediate node contains a route to the source node in its cache, it may send the route reply back to the source node using this cached route **or**
  - If symmetric links exist, then the route reply packet containing the route hop list may be sent back to the source node along the route listed in the route reply packet **or**
  - If symmetric links do not exist then the intermediate/destination node may initiate a route discovery process to the source and the route reply may piggyback this new route request.
- The duration of the validity of discovered routes is dependent on the characteristics of the network. Route caches will flush out stale routes more often in networks of higher mobility, necessitating new costly attempts at route discovery .

Ad Hoc Networks — A **REACTIVE** Algorithm – Dynamic Source Routing



Ad Hoc Networks — A **REACTIVE** Algorithm – Dynamic Source Routing

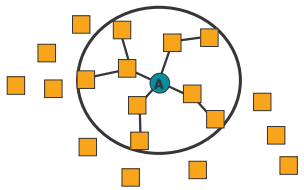
- Since DSR is a protocol designed for use in mobile ad hoc networks, it must be able to deal with the breakdown of routes. Routes may falter for any of the following reasons:
  - Change in link status: *symmetric / asymmetric / no link*
  - Increased node mobility
  - Congestion
- DSR uses the process of **Route Maintenance** to overcome problems with established routes by means of two packet types:
  - Route Error packets
    - Generated when a node encounters a fatal error in transmission. This packet will detail the failed link and may be flooded to other immediate nodes. All nodes with route caches containing routes dependent on the failed node will truncate these routes from the failed node onwards.
  - Acknowledgement packets
    - These packets are used to verify the transmission of packets from node to node on a hop-by-hop basis.

Ad Hoc Networks — Hybrid Routing Algorithms

- As there are pros and cons associated with both reactive and proactive routing protocols, another type of algorithm attempts to leverage the positive aspects of both types.
- The Zone Routing Protocol (**ZRP**) is a hyrbrid protocol which combines on-demand routing with table-based routing.
- For route discovery inside a zone, an arbitrary proactive routing scheme (e.g. distance vector, link-state) can be applied. Then for interzone routing an on-demand routing protocol (DSR, TORA) can be used.

Ad Hoc Networks — Hybrid Routing Algorithms - **ZRP**

- ZRP in a nutshell:
  - Uses a proactive scheme to build routes up to **x** hops
  - Uses a reactive scheme to find routes more than **x** hops away
- Each node in the network has its own routing zone:
  - The size of the zone is defined by a zone radius (**x**) which is defined by a metric such as number of hops
  - Each node must maintain routing information for its own zone

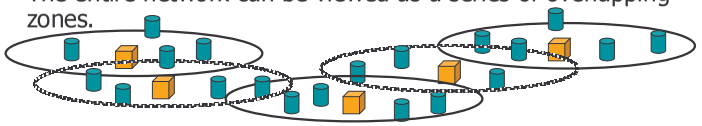


A ZRP enabled network, illustrating the routing zone of Node A.

The zone radius is 2 hops.

Ad Hoc Networks — Hybrid Routing Algorithms - **ZRP**

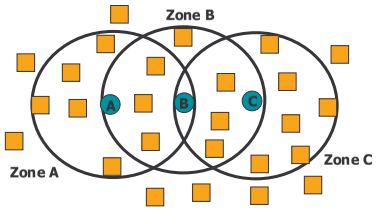
- The entire network can be viewed as a series of overlapping zones.



- Within each zone routing information is gathered by means of an optimal proactive **Intra Zone Routing Protocol (IARP)**, whether it be a link-state or distance-vector protocol.
  - e.g. FSR, DSDV
- The size of the zone depends on the characteristics of the network in that area:
  - The zone radius of each zone may change dynamically to reflect the local characteristics of the network, i.e. low-mobility → larger zone radius, as proactive updating is less expensive and vice versa for high-mobility.

Ad Hoc Networks — Hybrid Routing Algorithms - **ZRP**

- If a destination node is within the zone of the source node, then the source node will have immediate routing information, due to the employment of a proactive routng protocol.
- If the destination node is outside the zone then the node employs its reactive **Inter Zone Routing Protocol (IERP)** to retrieve a route.
  - e.g. DSR



- Node A wants a route to node C.
- C is not within the routing zone of A, so A sends a query packet to the nodes at the edge of its zone, i.e. B.
- B checks its proactively maintained routing tables to see if it knows of a route to C.
- Since C is within B's routing zone, B is able to send a route back to A.