Humboldt University

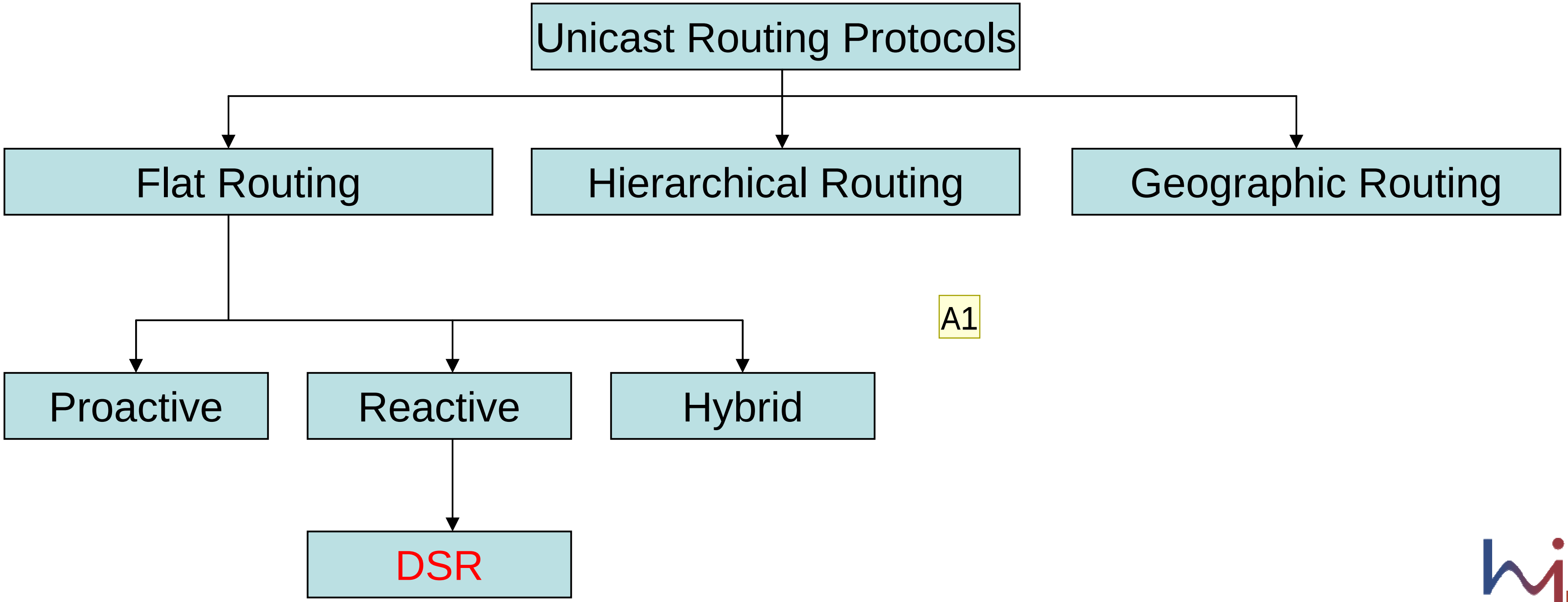
Computer Science Department
Systems Architecture Group
http://sar.informatik.hu-berlin.de

Dynamic Source Routing

In Ad Hoc Wireless Mesh Networks

Background

- Routing
  - Difficult problem
    - Wireless communication
    - Dynamic topology
    - Frequent link broken (network partition)
  - No single protocol works well in all cases



A1

Flat Routing
Every node plays an equal role and takes the same resposibility

Hierarchical Routing
Nodes are grouped, some nodes are selected as group leaders
Leaders take more responsibilities than others

Geographic Routing
Use geographic location information to make routing and forwarding decisions

AZu; 22.04.2005

Trade-Off Between Proactive and Reactive

- Latency of route discovery
  - Proactive protocols have lower latency since routes are maintained all times
  - Reactive protocols have higher latency because a node needs to find a route when it has data to send

- Overhead of route maintenance
  - Reactive protocols have lower overhead since routes are maintained only if they are needed
  - Proactive protocols have higher overhead due to continuous route updating

- Which approach achieves a better trade-off depends on the traffic and mobility patterns
  - If most nodes always have data to send and nodes' mobility is high, proactive protocols may produce higher routing efficiency
  - If only a few nodes have data to send and nodes' mobility is low, reactive protocols may produce higher routing efficiency

Basic Idea behind DSR

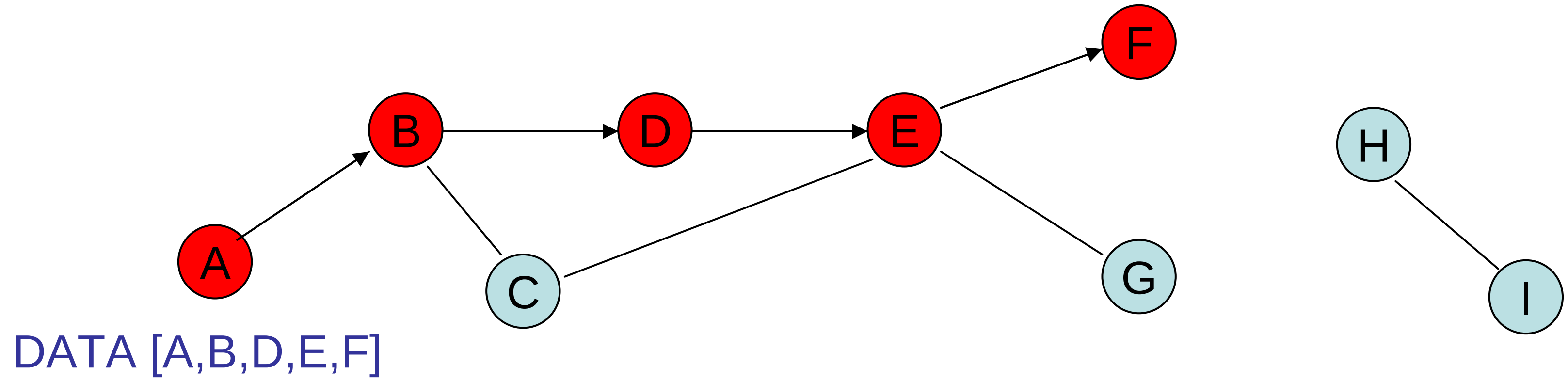
- All nodes are willing to forward packets for other nodes in the network (each node is a relay point)

- The diameter of a network will not be too large
  - Packet header will be bigger than payload

- The node’s speed is moderate
  - Local route cache will become stale soon if node’s speed is high

- All nodes are overhearing (promiscuous)
  - No energy saving

Dynamic Source Routing



- When A sends a data packet to F, the entire route is included in the packet header
- Intermediate nodes use the source route embedded in the packet's header to determine to whom the packet should be forwarded
- Different packets may have different routes, even they have the same source and destination

→ Hence called as dynamic source routing

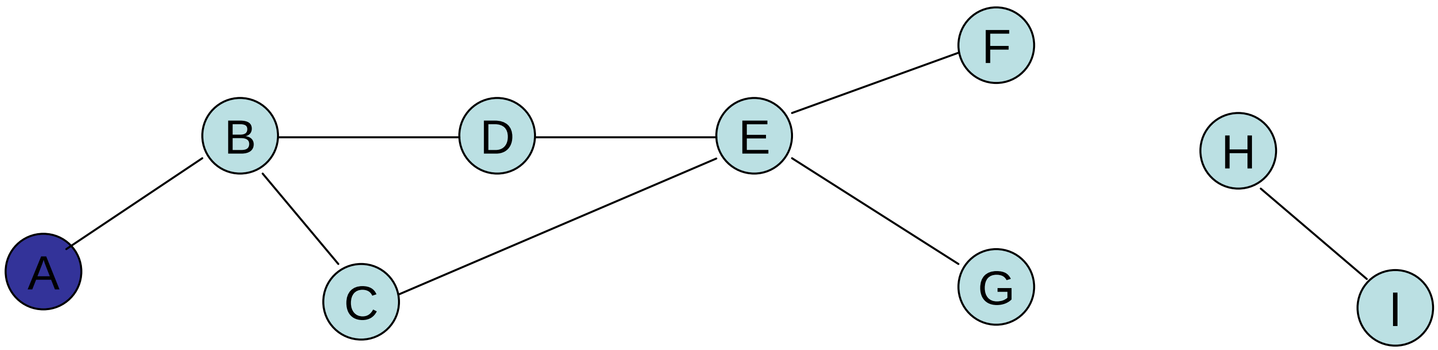
DSR Basics

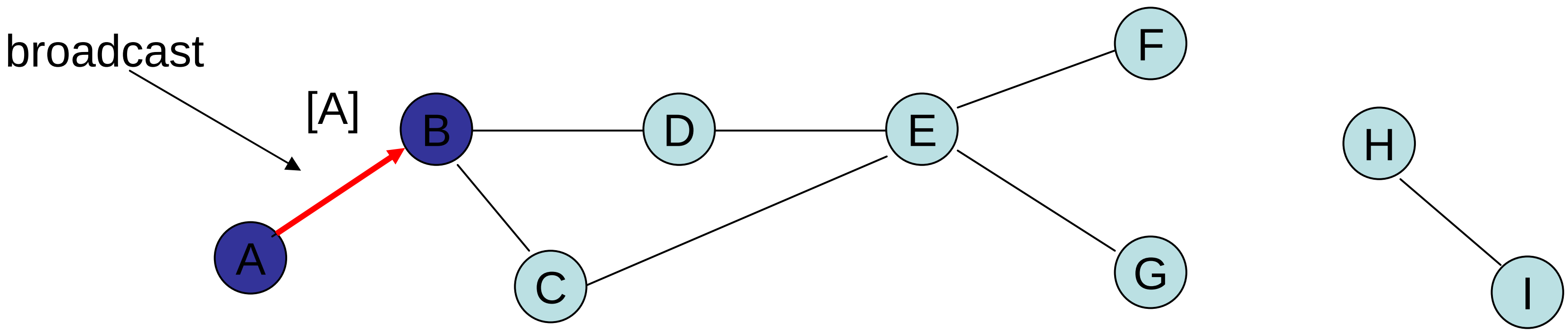
- Two basic mechanisms
  - Route Discovery
    - Route Request (RREQ)
    - Route Reply (RREP)
  - Route Maintenance
    - Route Error (RERR)
- Key optimization
  - Each node maintains a route cache
    - Overhears data, RREQ, RREP, and RERR packets
    - Passively collects new routes as many as possible
    - Reduces the cost of Route Discovery and Route Maintenance

Route Discovery

- When to perform a Route Discovery ?
- Every route request packet (RREQ) contains
  <*target address, initiator address, route record, request ID*>
- Each node maintains a list of the
  < *initiator address, request ID*>
- When a node Y receives a RREQ
  - Discards the route request packet
    - if < *initiator address, request ID*> is in its list
  - Returns a route reply packet which contains a route from initiator to target
    - If Y is target
    - If Y has an entry in its route cache for a route to target
  - Appends itself address to the route record in RREQ and re-broadcast RREQ

Route Discovery in DSR

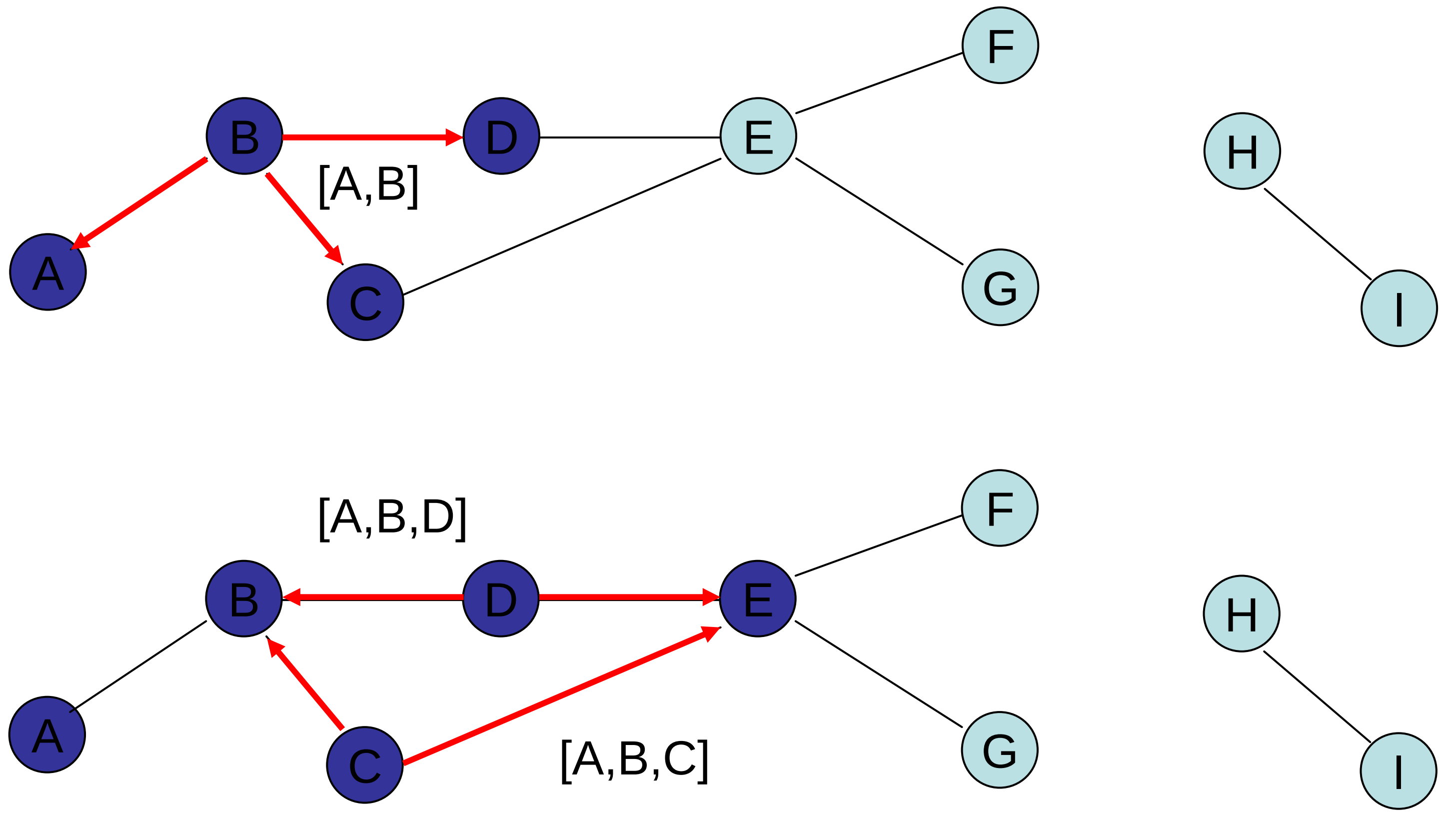




Represents a node that has received RREQ for F from A

Represents transmission of RREQ

Route Discovery in DSR

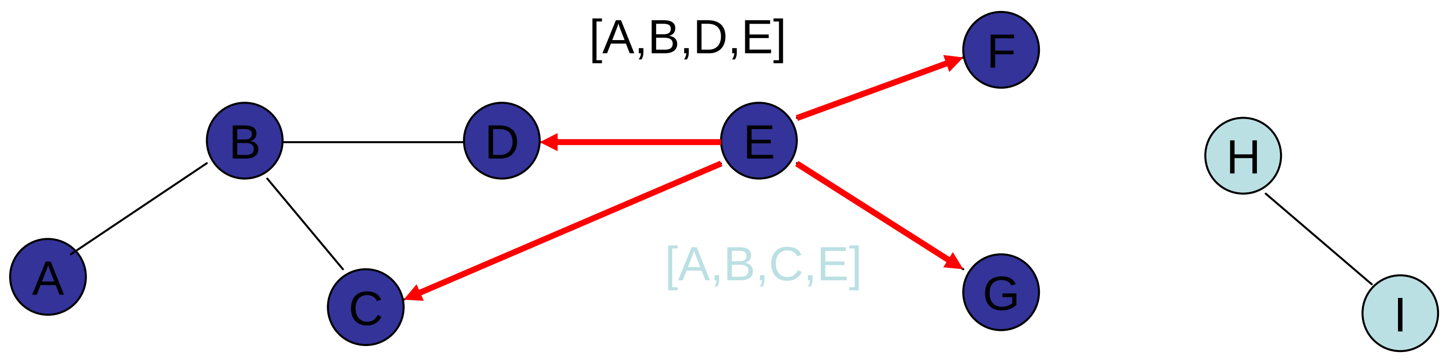


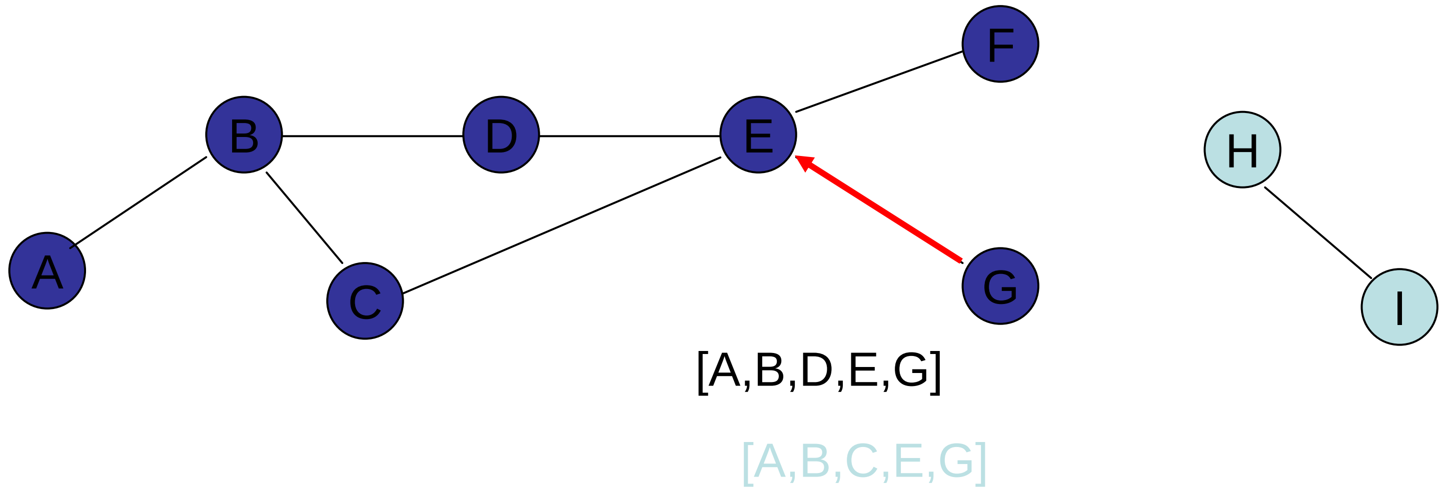
Node E receives packet RREQ from two neighbors

→ potential for collision

B receives RREQ from C and D, but does not forward it again, because B has already forwarded RREQ once

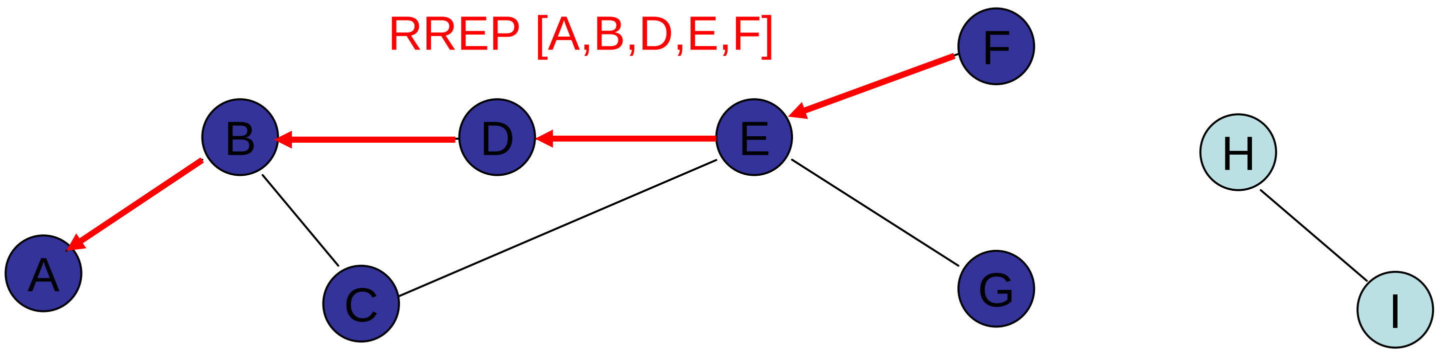
Route Discovery in DSR





F does not forward RREQ, because F is the intended target

Route Reply in DSR

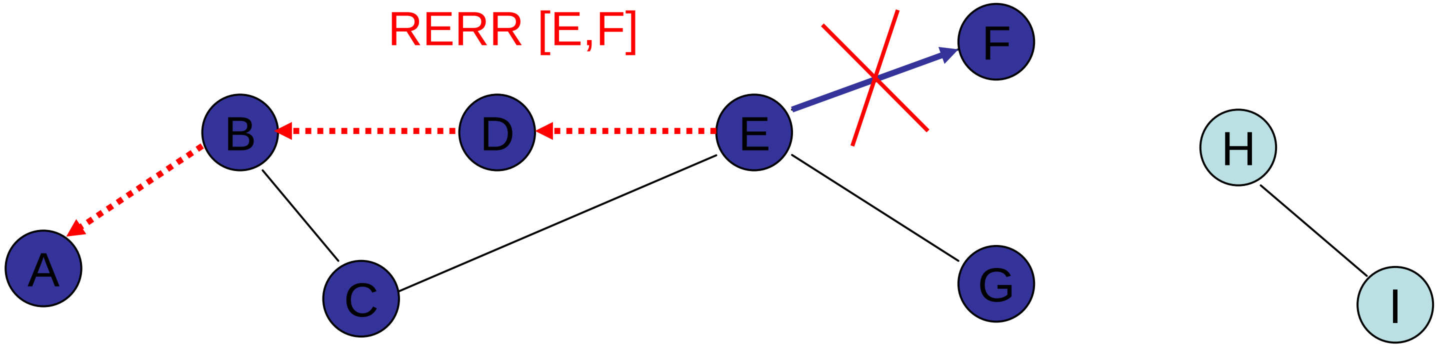


Represents RREP control message

More Details For Route Reply in DSR

- Destination F on receiving the first RREQ, sends a Route Reply (RREP)
- RREP includes the route from A to F
- How Route Reply packet is sent to A?
  - Route Reply can be sent by reversing the route in Route Request (RREQ)
    - If links are bi-directional
  - If unidirectional (asymmetric) links are allowed, then a route to A is needed
    - Local route cache has a route to A
    - Piggybacking Route Reply in Route Request packet for A
- NOTE: If IEEE 802.11 MAC is used, then links have to be bi-directional

Route Maintenance



E sends a route error to A along route E-D-B-A when it finds link [E-F] broken

Nodes hearing RERR update their route cache to remove all invalid routes related with link E-F

More Details For Route Reply in DSR

Route [A, node-1,node-2,……,node-k, F]

- Hop-by-hop maintenance (MAC or network layer)
  - How to find link [node-i,node(i+1)] is down ?
    - Utilize MAC level acknowledgement
    - Passive acknowledge (overhearing node(i+1) re-transmission)
    - Insert a bit in packet header to ask an explicit acknowledgement from node(i+1)
  - How to send route error packet to A?
    - Use the reverse route [node-i,node(i-1), ……,node-1, A]
    - Use node-i route cache to get a route to A
    - Piggybacking route error packet in route discovery packet A
- End-to-end maintenance (transport or application layer)
  - F sends ACK to A to indicate the route status
  - But A does not know which link is broken

DSR Optimizations (1)

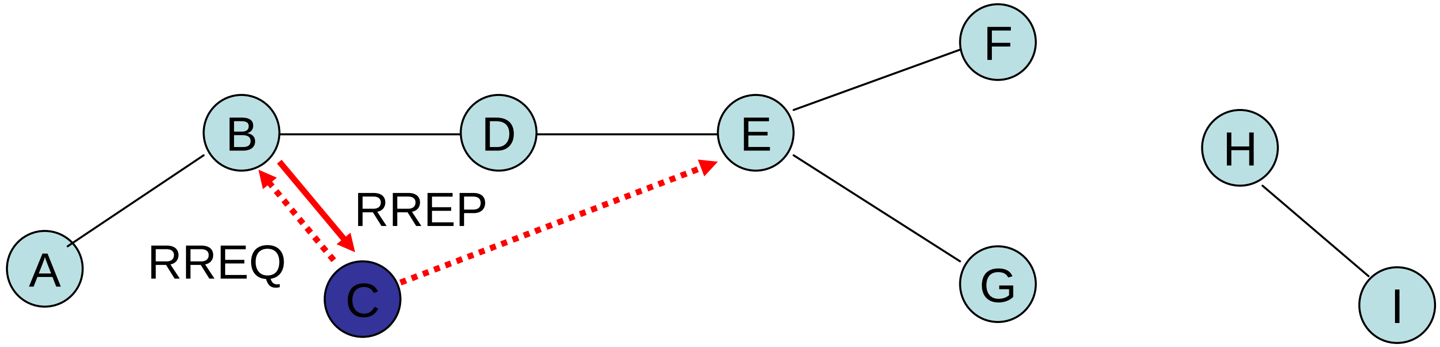
- Route Caching
  - Each node caches a new route it learns by any means
  - When A finds route [A,B,D,E,F] to F, A also learns route [A,B,D,E] to E
  - When G receives Route Request [A,B,D,E] destined for some node F, G learns route [G,E,D,B,A] to A if links are bi-directional
  - D forwards Route Reply [A,B,D,E,F], D learns route [D,E,F] to F
  - When B forwards Data [A,B,D,E,F] it learns route [B,D,E,F] to F
- Advantages
  - Can Speed up Route Discovery
  - Can Reduce Propagation of Route Requests
- Problems
  - With time passing and node moving, cached routes may become invalid (Stale caches can adversely affect on network performance)

DSR Optimizations (2)

- Avoid Route Reply Storm problem:
  - When intermediate nodes reply Route Request by using local route cache
    - Pick a random number d=H*(h-1+r) , H= average per hop delay; h= length of the route; r = random num in [0,1]
    - Delay transmitting the route reply for a period of d
    - During this period, cancel the route reply if overhearing a packet contains a route from the same initiator to the same target with length no longer than h

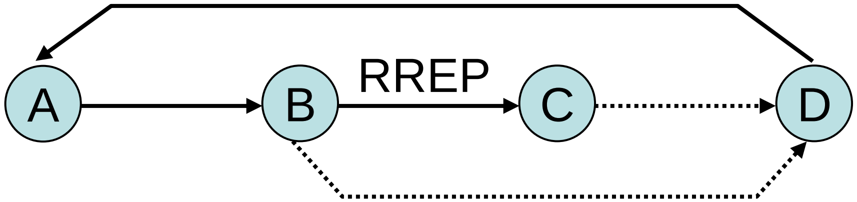
DSR Optimizations (3)

- Expanding Ring (limits the propagation of RREQ packets)
  - First, set TTL = 1 for first route request packet
  - If no route reply is received after some time period, set TTL = maximum for next route request packet



DSR Optimizations (4)

- Reflecting shorter route
  - D may find that route may be shortened by excluding the intermediate hop C, D then sends an unsolicited route reply packet to A to inform the shorter route



Expected Transmission Count Metric (ETX)

- Minimum hop-count metric:
  - Chooses arbitrarily among the different paths of the same minimum length.
  - Problems:
    - large differences in throughput among paths of the same length
    - ignoring the possibility that a longer path might offer higher throughput
- ETX:
  - Finds high-throughput paths on multi-hop wireless networks.
  - Minimizes the expected total number of packet transmissions (including retransmissions) required to successfully deliver a packet to the ultimate destination.

$$ETX = \frac{1}{d_f \times d_r}$$

d_f - forward delivery ratio is the measured probability that a data packet successfully arrives at the recipient

d_r - the reverse delivery ratio is the probability that the ACK packet is successfully received.

A2

A2 The expected probability that a
transmission is successfully received and acknowledged is df × dr
AZu; 29.04.2005

Expected Transmission Count Metric (ETX)

- DSR-Implementation:
  - Link probes are used to measure delivery ratios.
  - When a node forwards a request, it appends not only its own address, but also the metric for the link over which it received the request.
  - These metrics are included in the route replies sent back to the sender.
  - When a node receives a request which it has already forwarded, it forwards it again if the accumulated route metric is better than the best which it has already forwarded with this request ID (increases the chances that the originator will hear about the route with the best metric).
  - Entries in the link cache are weighted by the metrics which were included in the route replies.
  - No changes made in Dijkstra algorithm

Further Information

- http://www-2.cs.cmu.edu/~dmaltz/dsr.html