Distributed Systems

Location based protocols

Rik Sarkar

University of Edinburgh
Fall 2014

Announcements

- Assignment deadline extended to **Nov. 10**
- No class/office hour on Nov 10

Routing in ad hoc wireless networks

- Find route between pairs of nodes wishing to communicate.
- Proactive protocols: maintain routing tables at each node that is updated as changes in the network topology are detected.
  - Heavy overhead with high network dynamics (caused by link/node failures or node movement).
  - Not practical for networks that change frequently

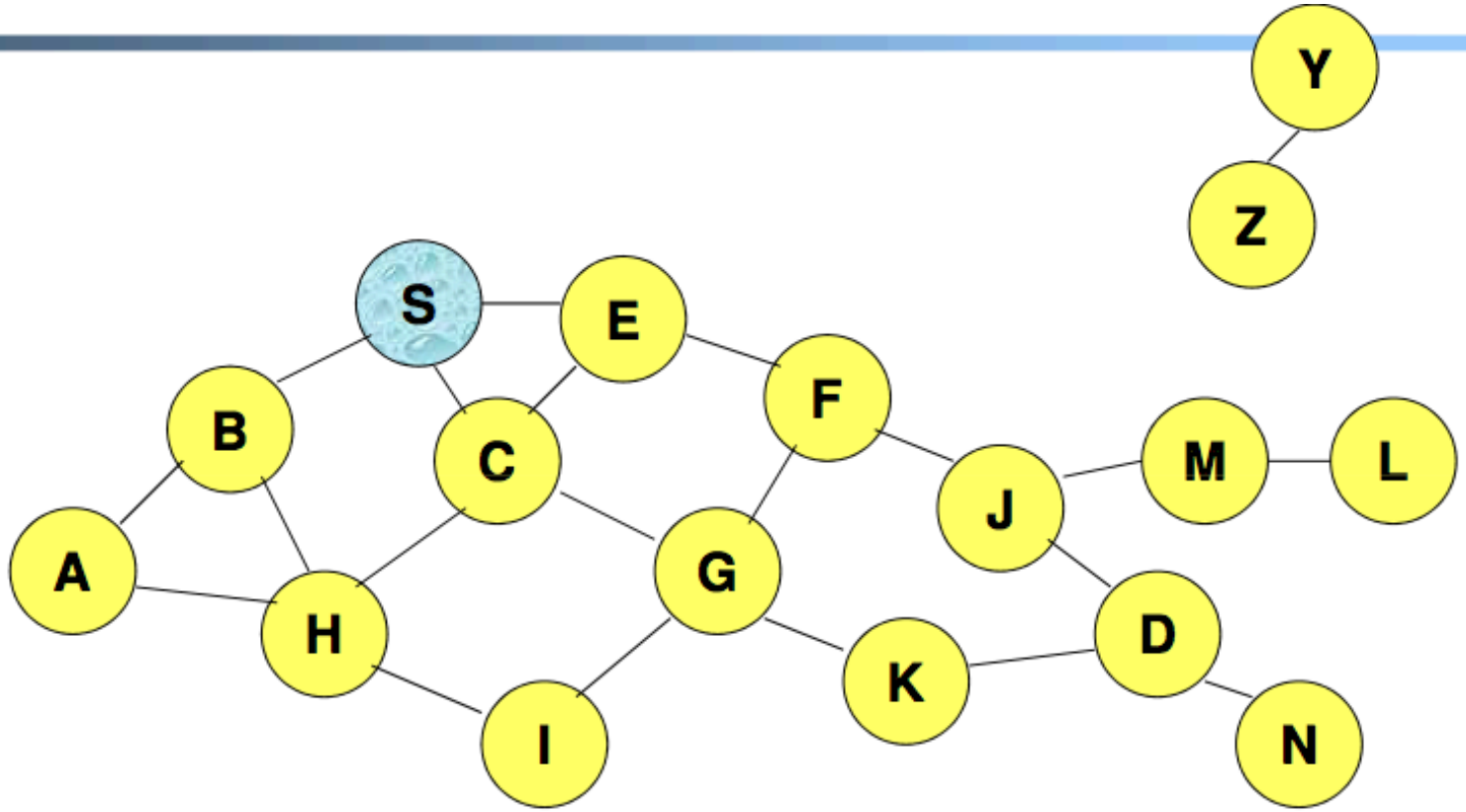
Routing in ad hoc wireless networks

- Reactive protocols: routes are constructed on demand. No global routing table is maintained.
- More appropriate for networks with high rate of changes
  - Ad hoc on demand distance vector routing (AODV)
  - Dynamic source routing (DSR)

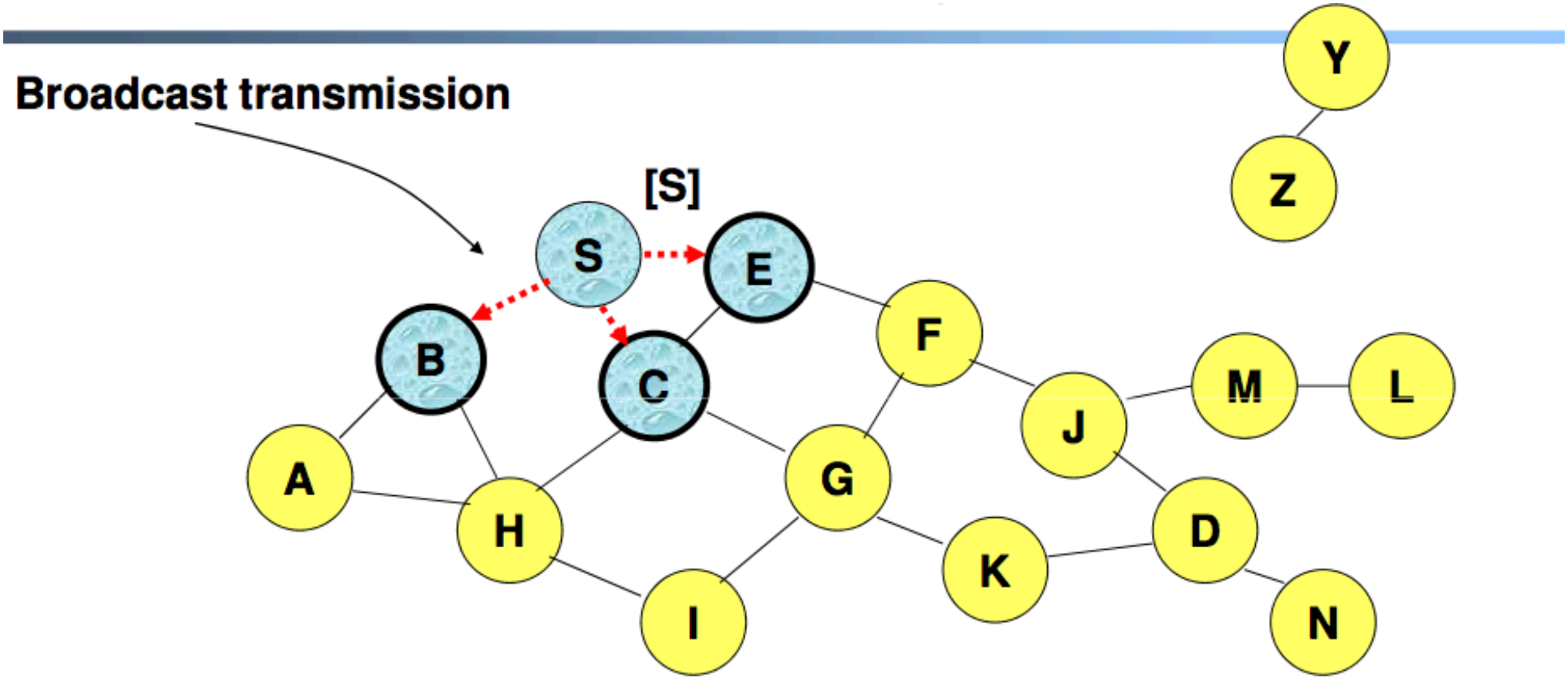
Dynamic Source Routing (DSR)

- Node S wants to send a message to node D
- S initiates a a route discovery
- S floods the network with route request (RREQ) message
- Each node appends its own id to the message

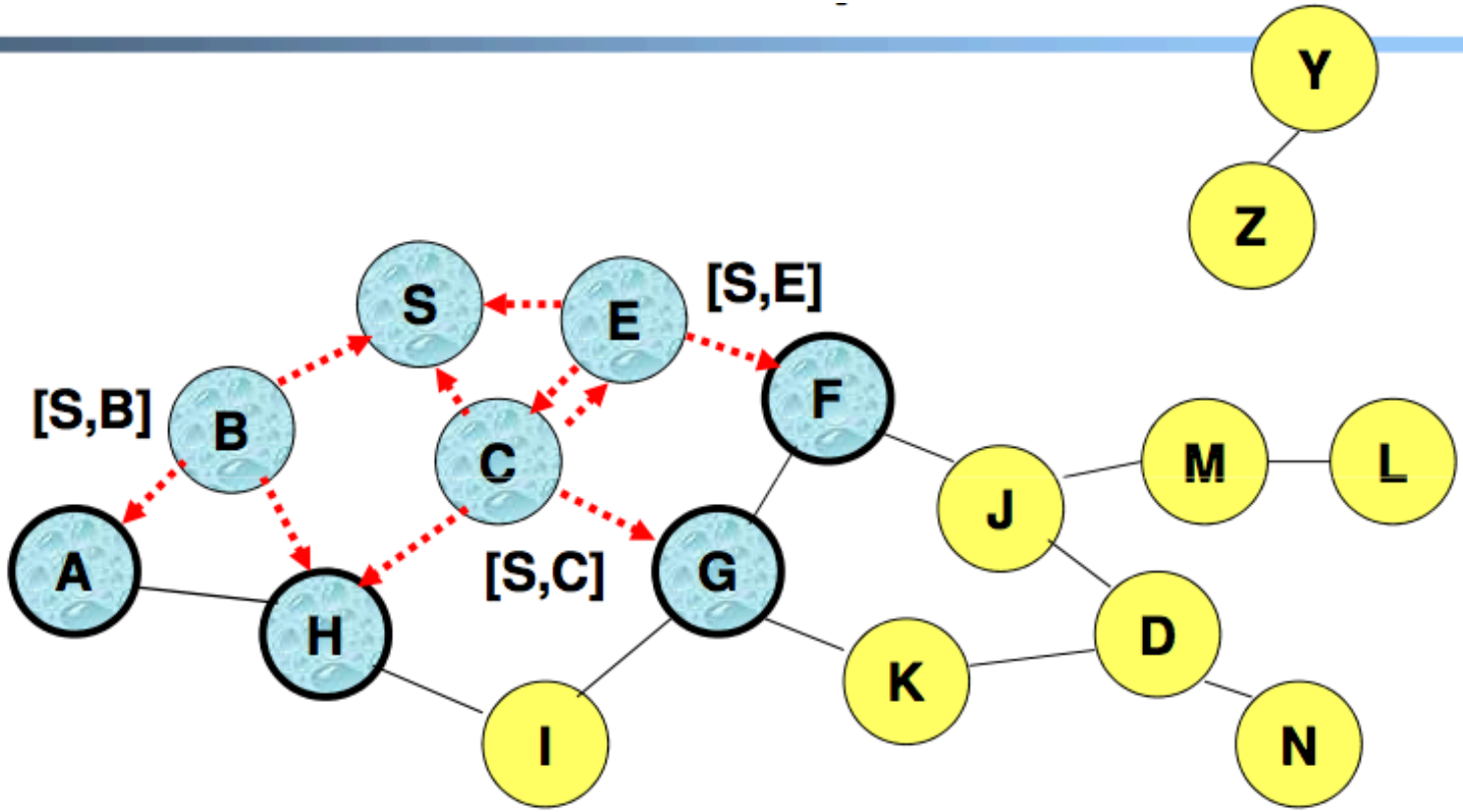
Route Discovery: RREQ



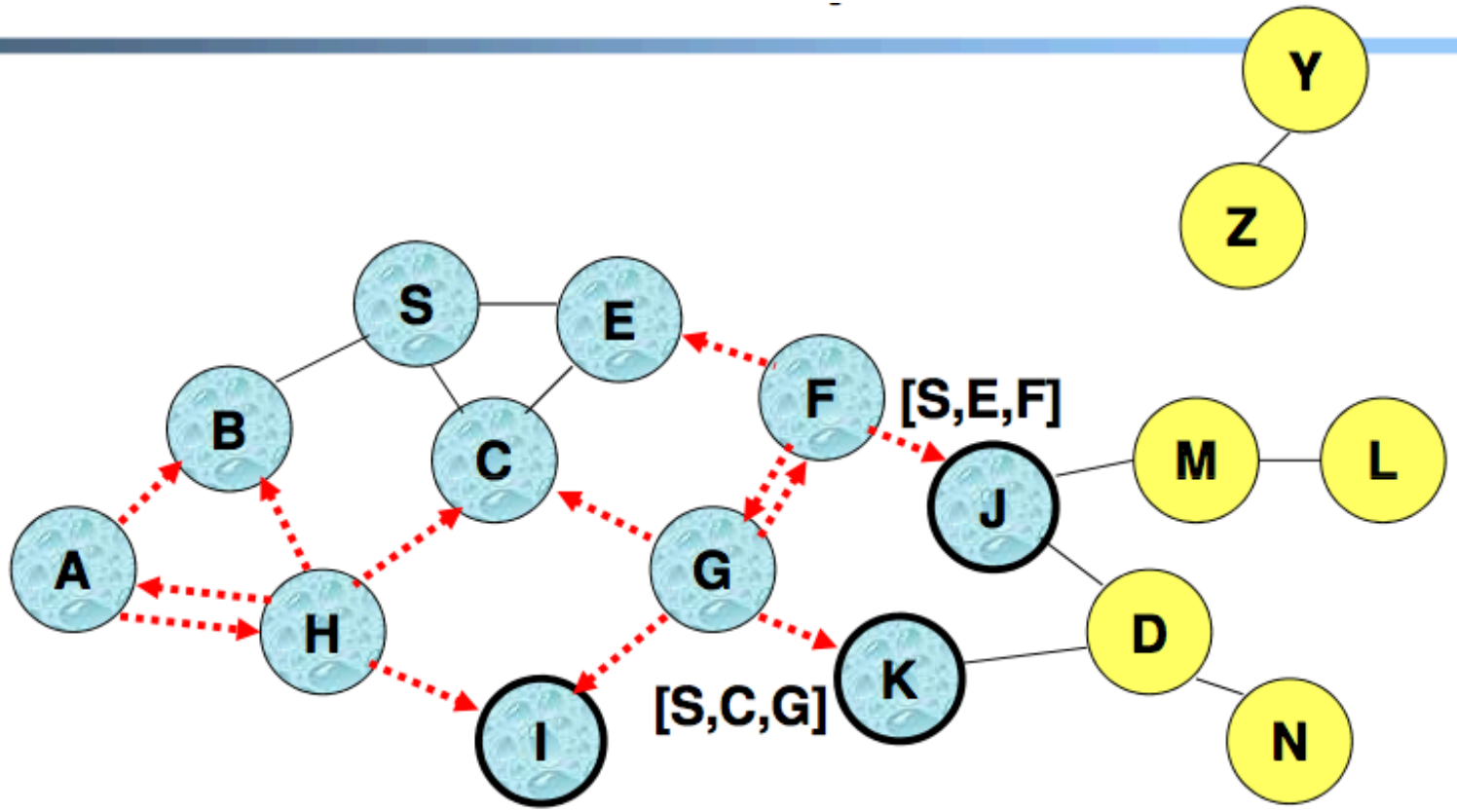
Route Discovery: RREQ



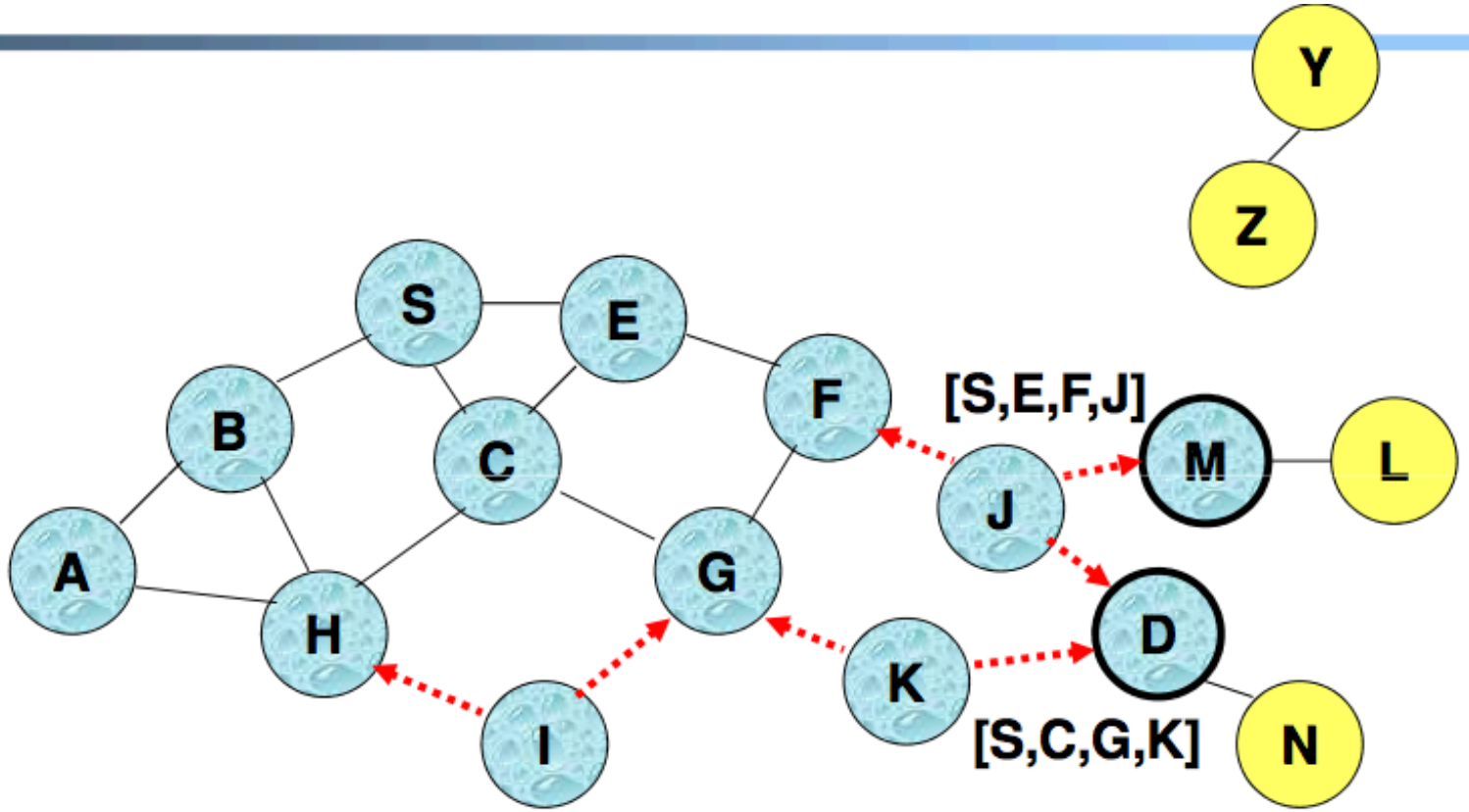
Route Discovery: RREQ



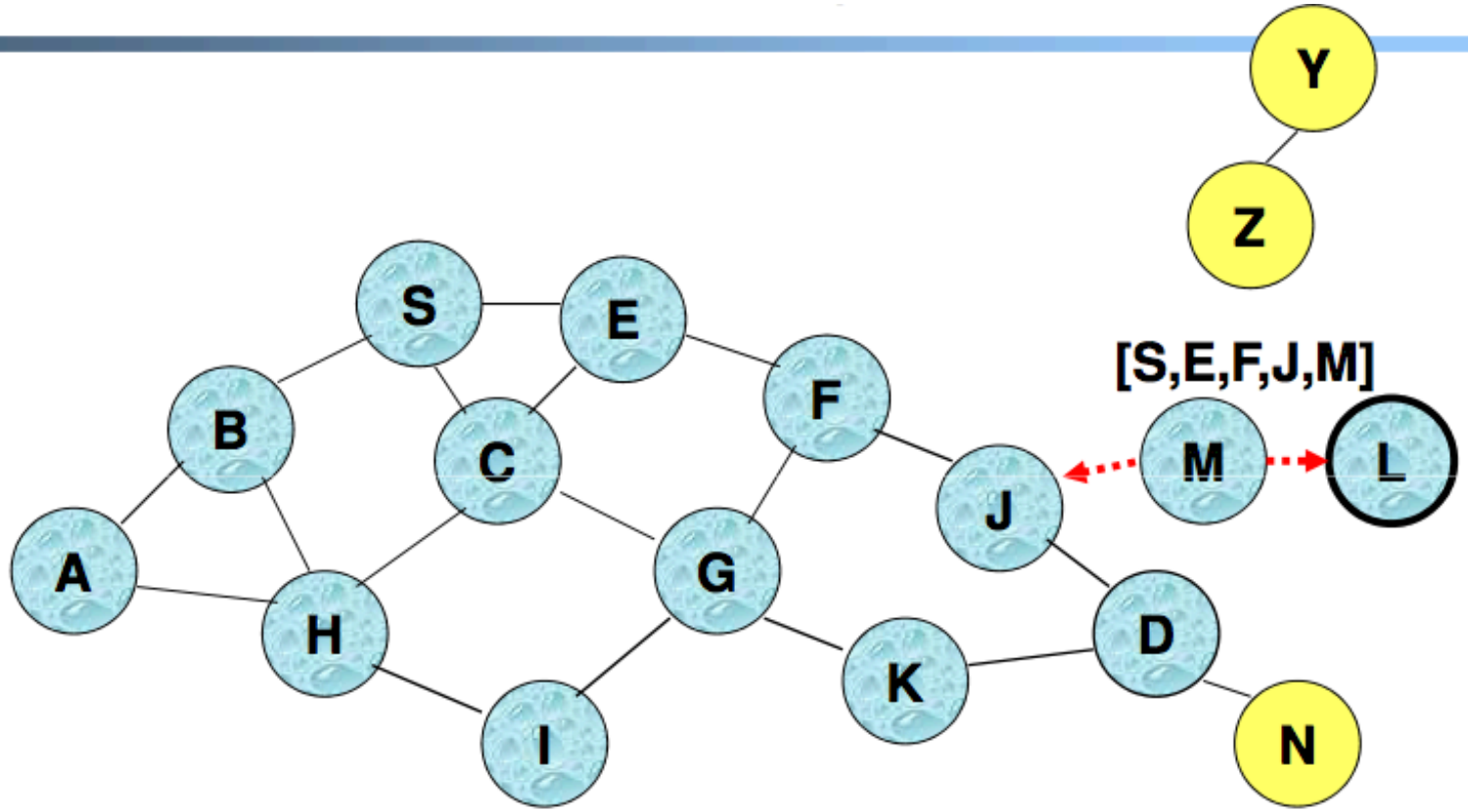
Route Discovery: RREQ



Route Discovery: RREQ



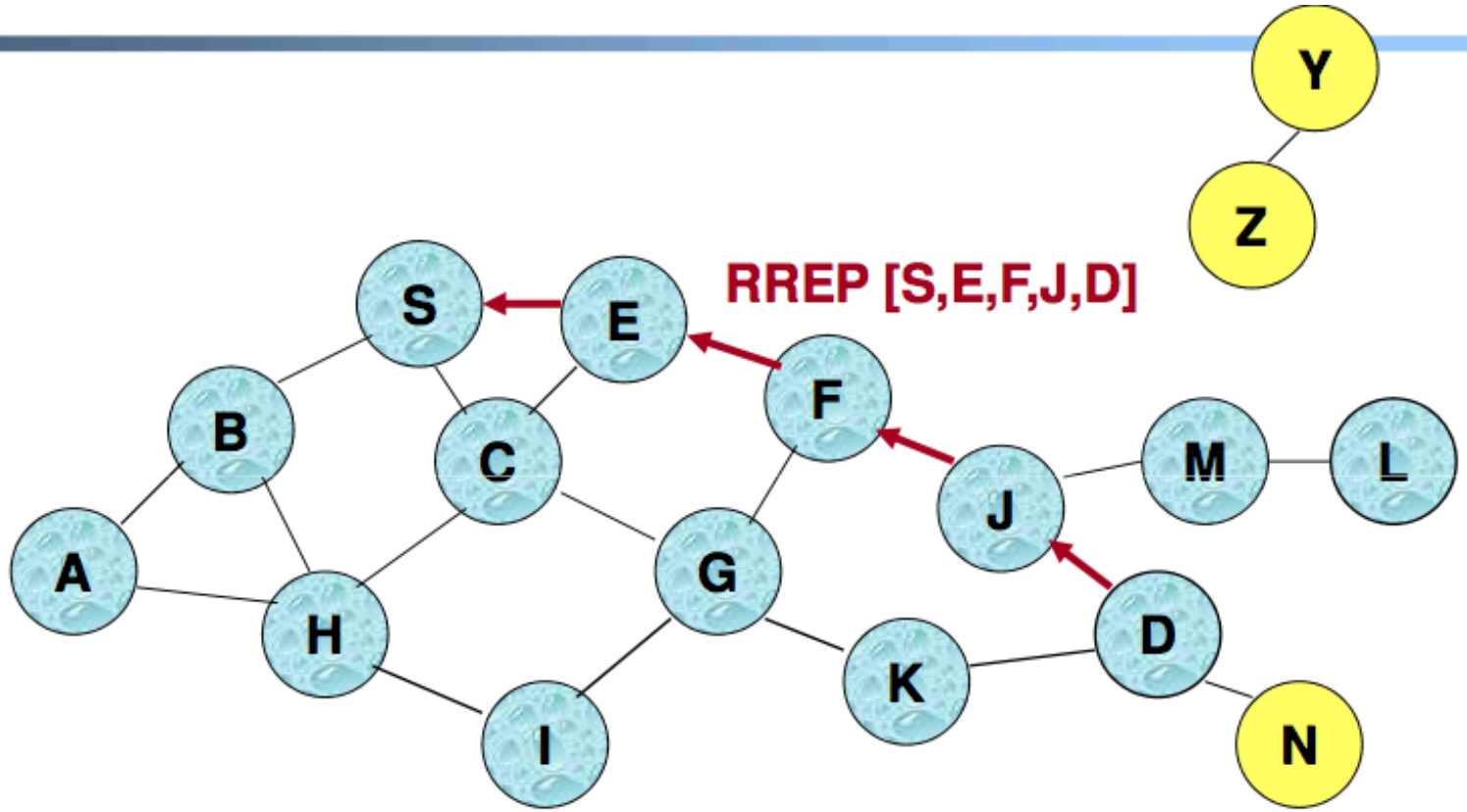
Route Discovery: RREQ



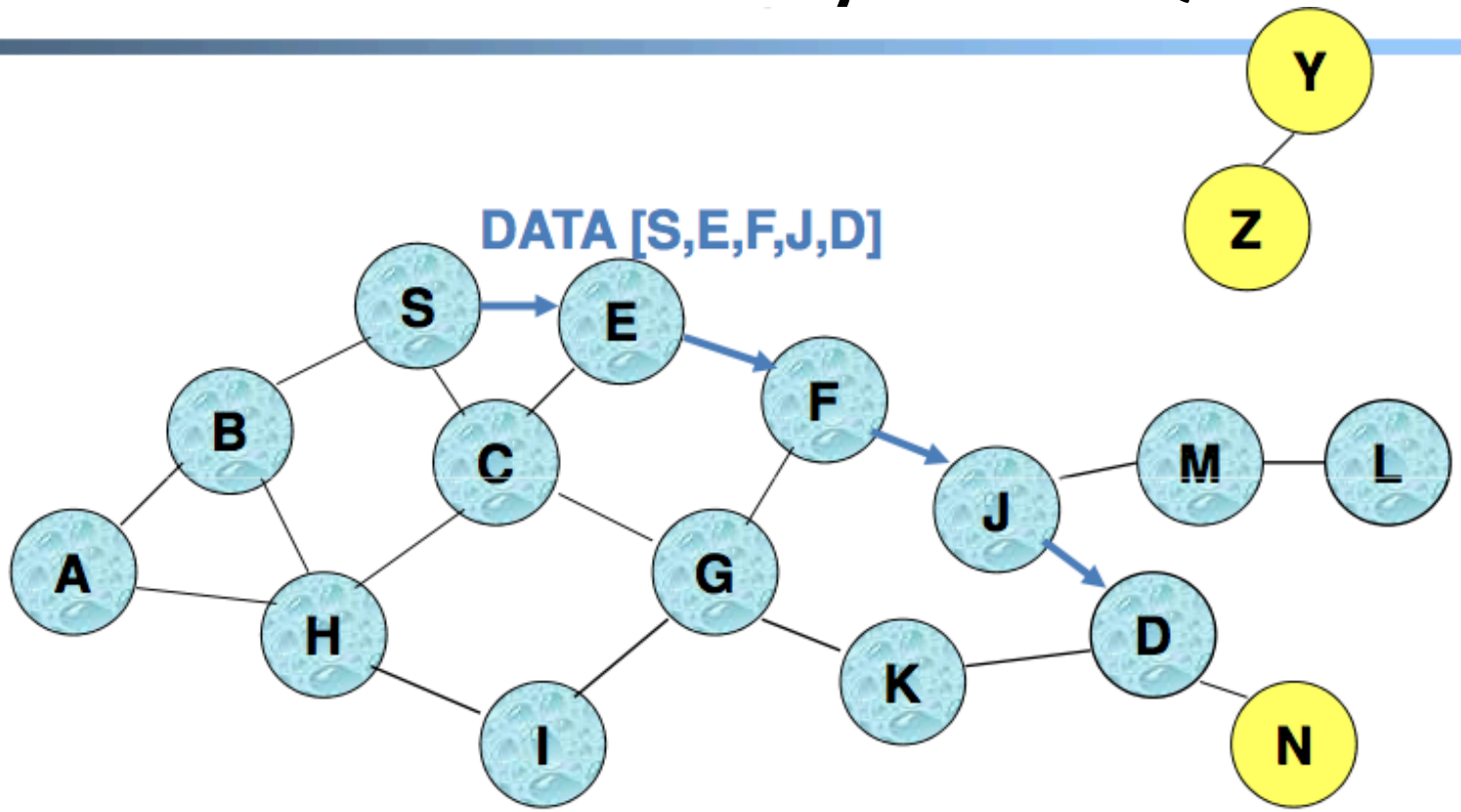
Route Discovery in DSR

- Destination D on receiving the first RREQ sends a *route reply* (RREP)
- RREP is sent on a route obtained by reversing the route in received RREQ

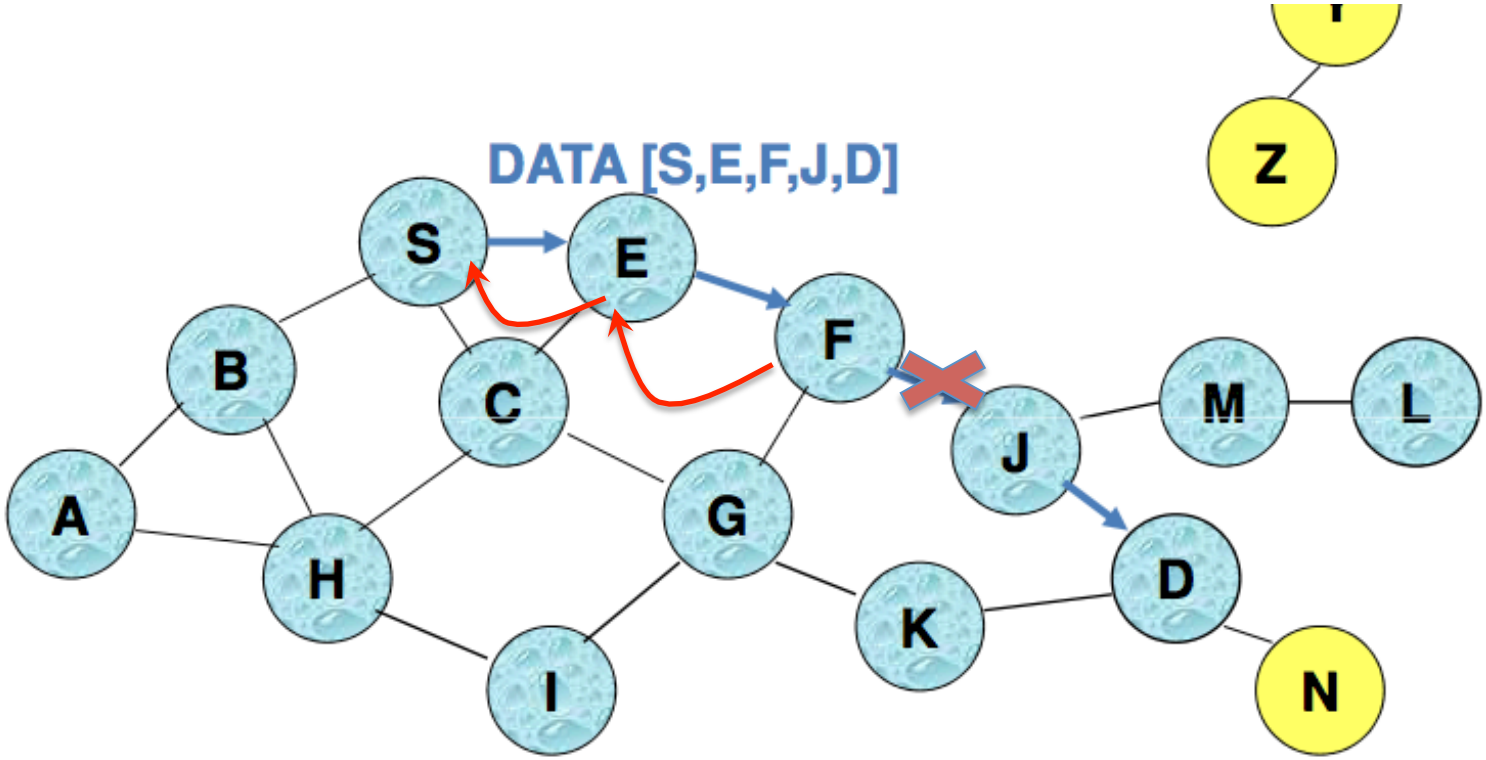
Route Discovery: RREQ



Route Discovery: RREQ



When node S sends a data packet to D, the entire route is included in the packet header, hence the name source routing



- When a link fails, an error message with the link name is sent back to S.
- S deletes any route using that link and starts discovery.

Route caching

- When a node receives or forwards a message, it learns routes to all nodes on the path
- Advantage:
  - S may not need to send RREQ
  - Intermediate node on receiving RREQ, can respond with complete route
- Disadvantage:
  - Caches may be stale: S tries many cached routes before starting a discovery. Or, intermediate nodes return outdated information.

DSR: Summary

Advantages:

- Routes computed only when needed
- Caching can make things efficient
- Does not create loops

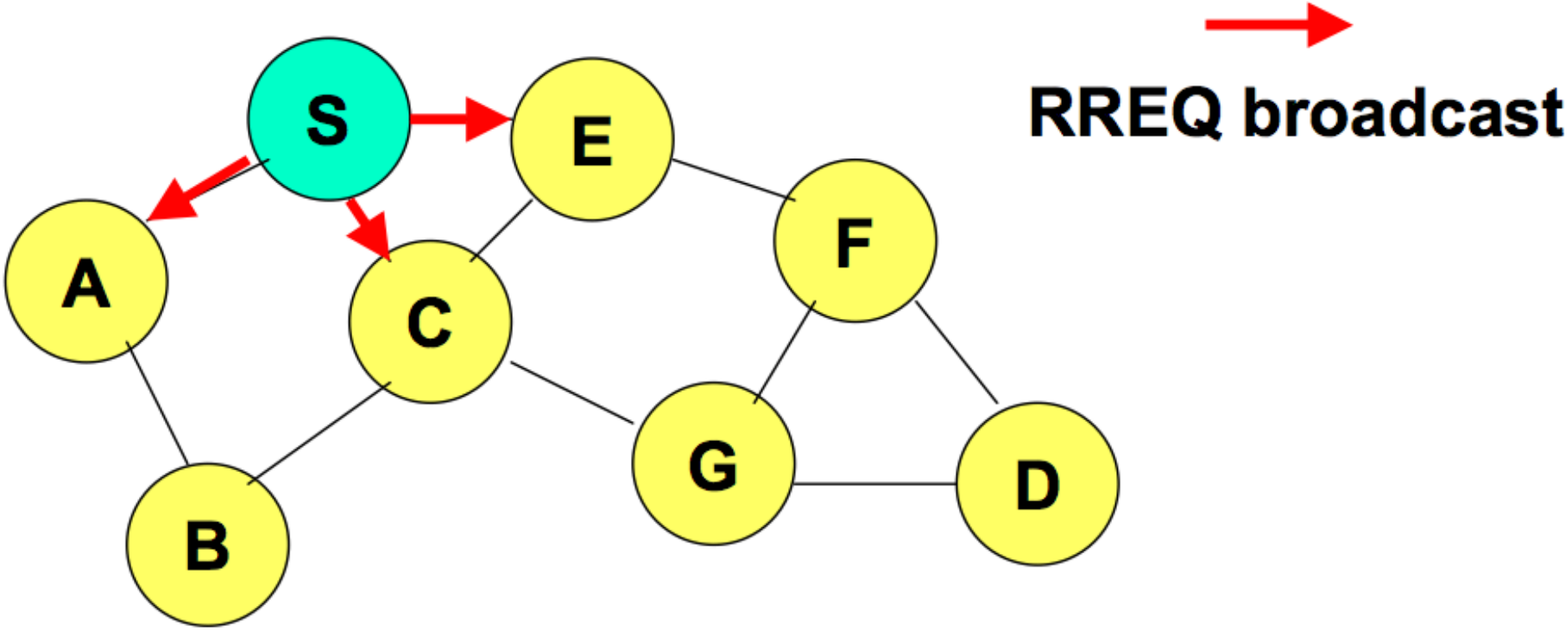
Disadvantages

- Entire route must be contained in message: can be long for large networks
- Flooding causes communication to many nodes
- Stale caches can be a problem
- Not suitable for networks with frequent changes

Ad hoc On-Demand Distance Vector Routing (AODV)

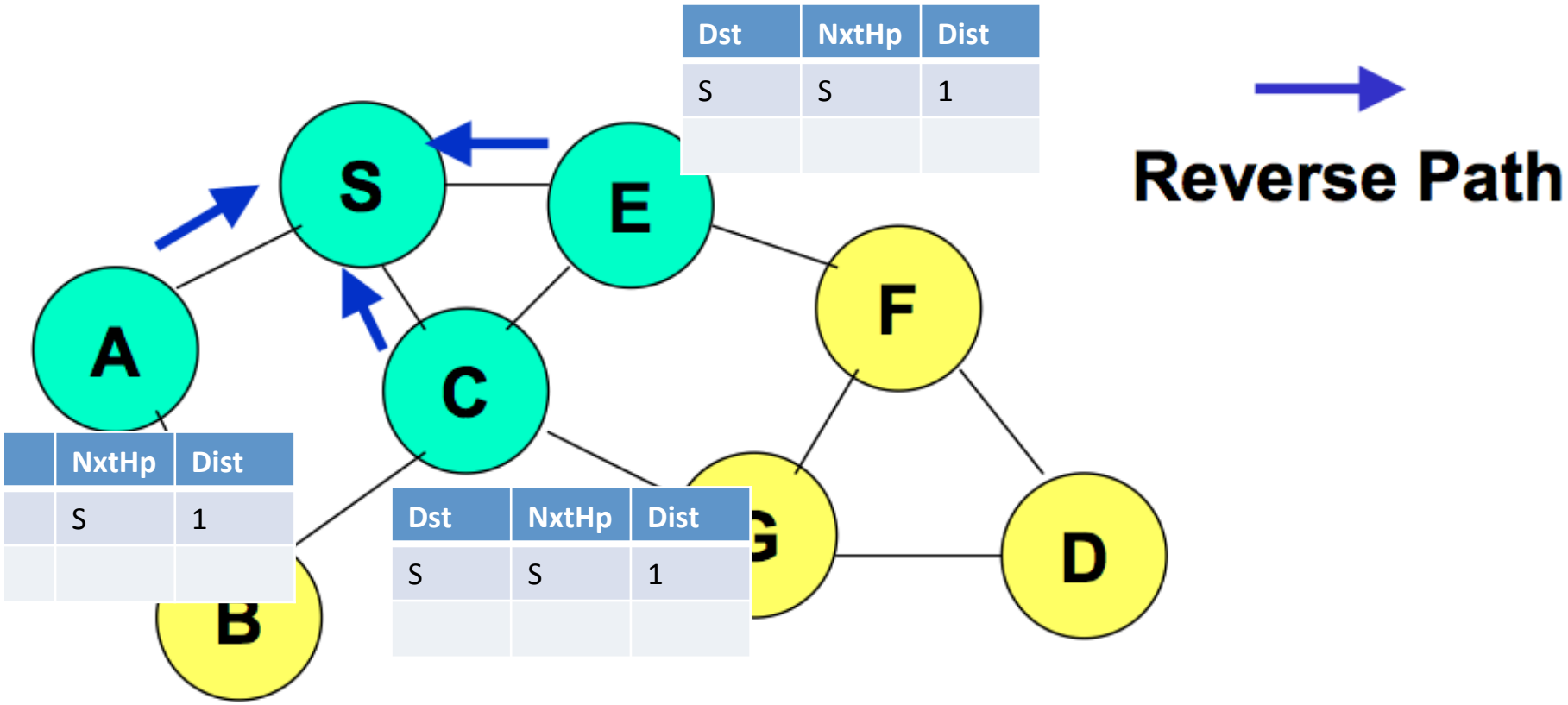
- Maintains routing tables at nodes so that the route need not be stored in the message
- No Caches: Only one route per destination

AODV Route Discovery



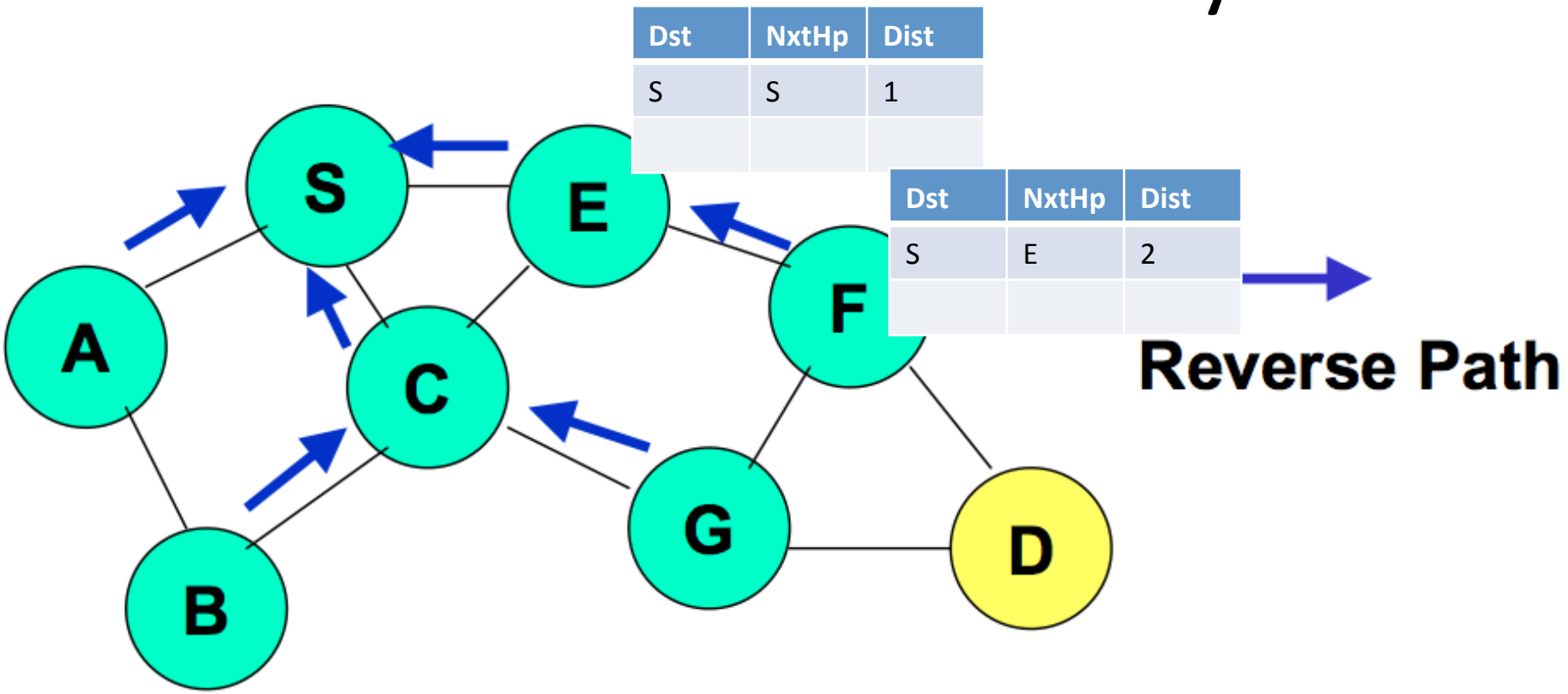
- Source floods the network

AODV Route Discovery



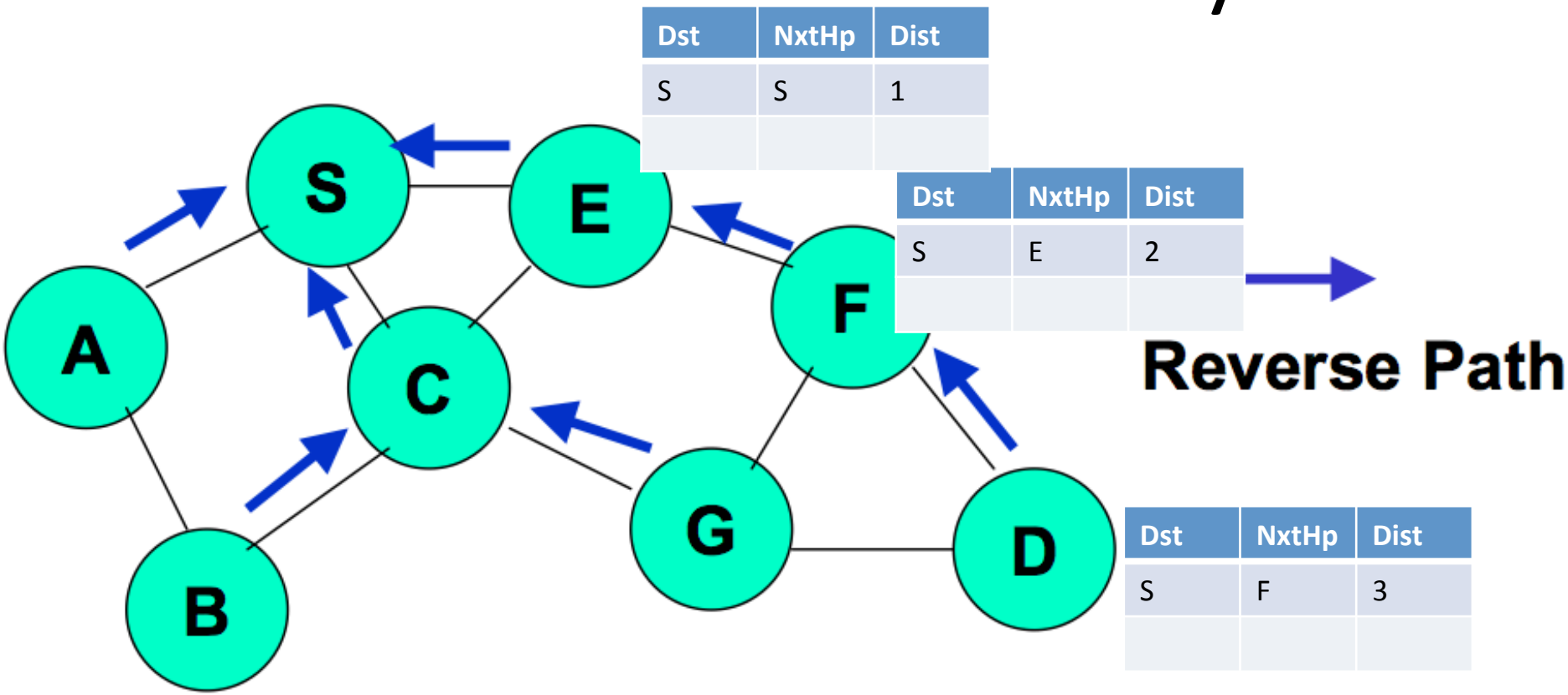
- Other nodes create parent pointer
- A node forwards a RREQ only once

AODV Route Discovery



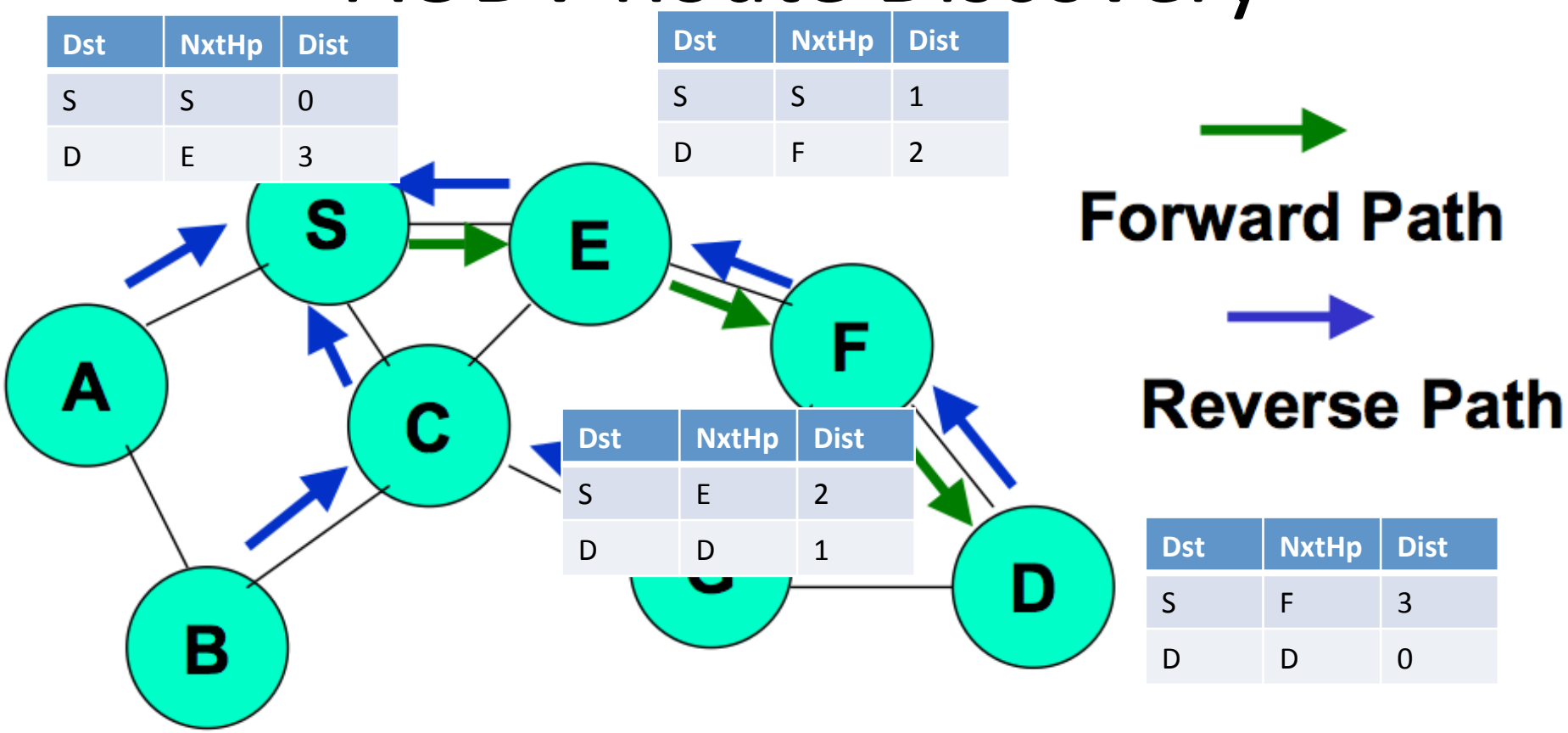
- Other nodes create parent pointer
- A node forwards a RREQ only once

AODV Route Discovery



- RREP is forwarded via reverse path

AODV Route Discovery

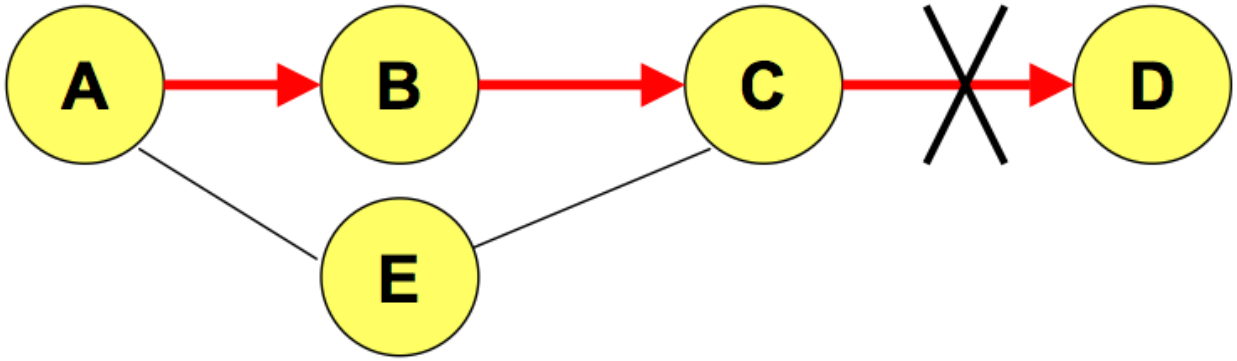


- RREP is forwarded via reverse path
- Creates a forward path

Route expiry

- A path expires if not used for a certain time.
- If a node sees that a routing table entry has not been used by this time, it removes this entry
- Even if the path itself is valid
- Good for networks with frequent changes
- Bad for static and stable networks

Can create loops

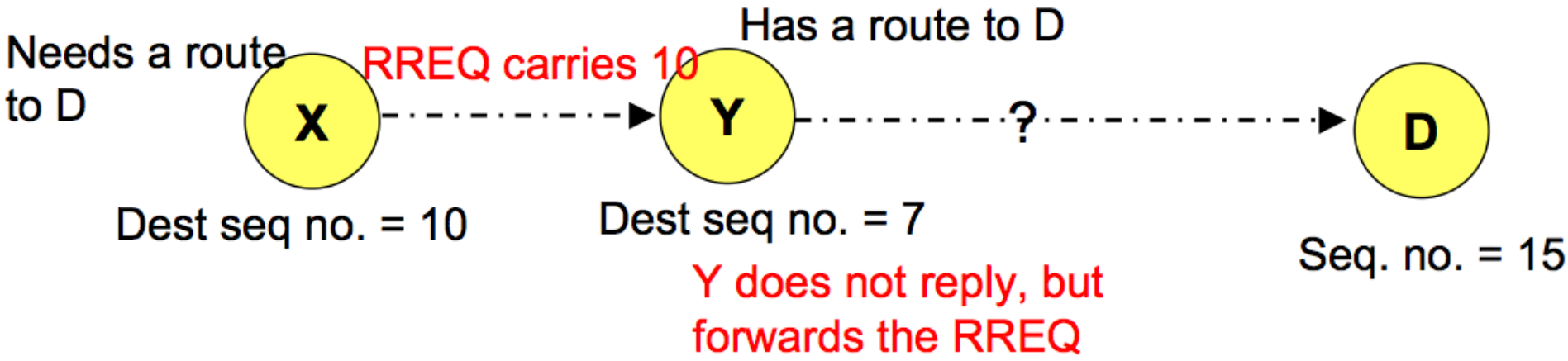


- Assume C->D link has failed, but A does not know because the ERR message was lost
- C is now trying to find path to D
- A responds since A thinks it has a path
- Creates loop: C-E-A-B-C

Sequence numbers in AODV

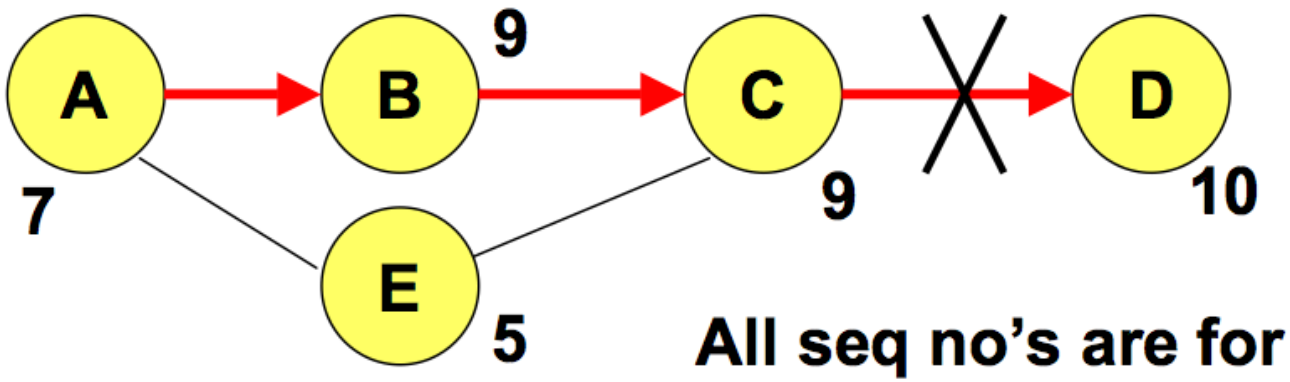
- If A has a route to D, A keeps a sequence number.
- A increments this number periodically: tells how old the information is

Using sequence numbers



- Rule : sequence number must increase along any route

Sequence number rule avoids loop



- A does not reply, since its sequence no. is less than that of C

AODV

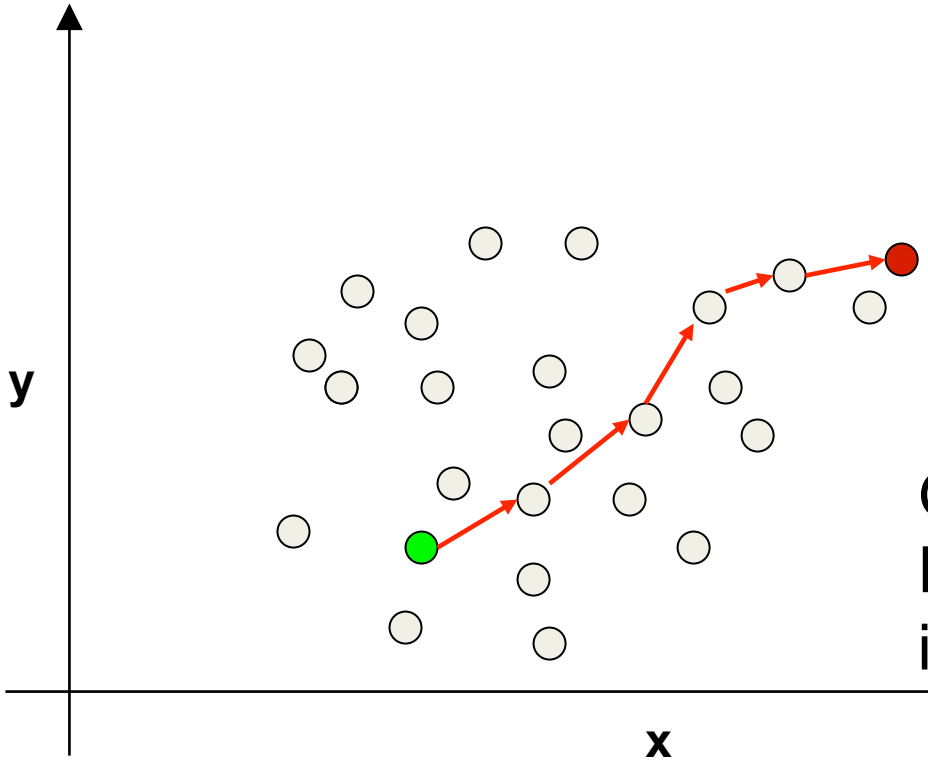
- Routing tables, message does not contain route
- Fresh routes preferred
- Old unused routes expire
- Stale routes less problematic
- Needs sequence numbers to prevent loops
- Better for more dynamic, changing environments

Routing in ad hoc networks

- Reactive protocols: routes are constructed on demand. No global routing table is maintained.
- More appropriate for networks with high rate of changes
  - Ad hoc on demand distance vector routing (AODV)
  - Dynamic source routing (DSR)
- Need flooding
  - Inefficient in large networks

Geographical routing: Using location

- Geographical routing uses a node's location to discover path to that node.

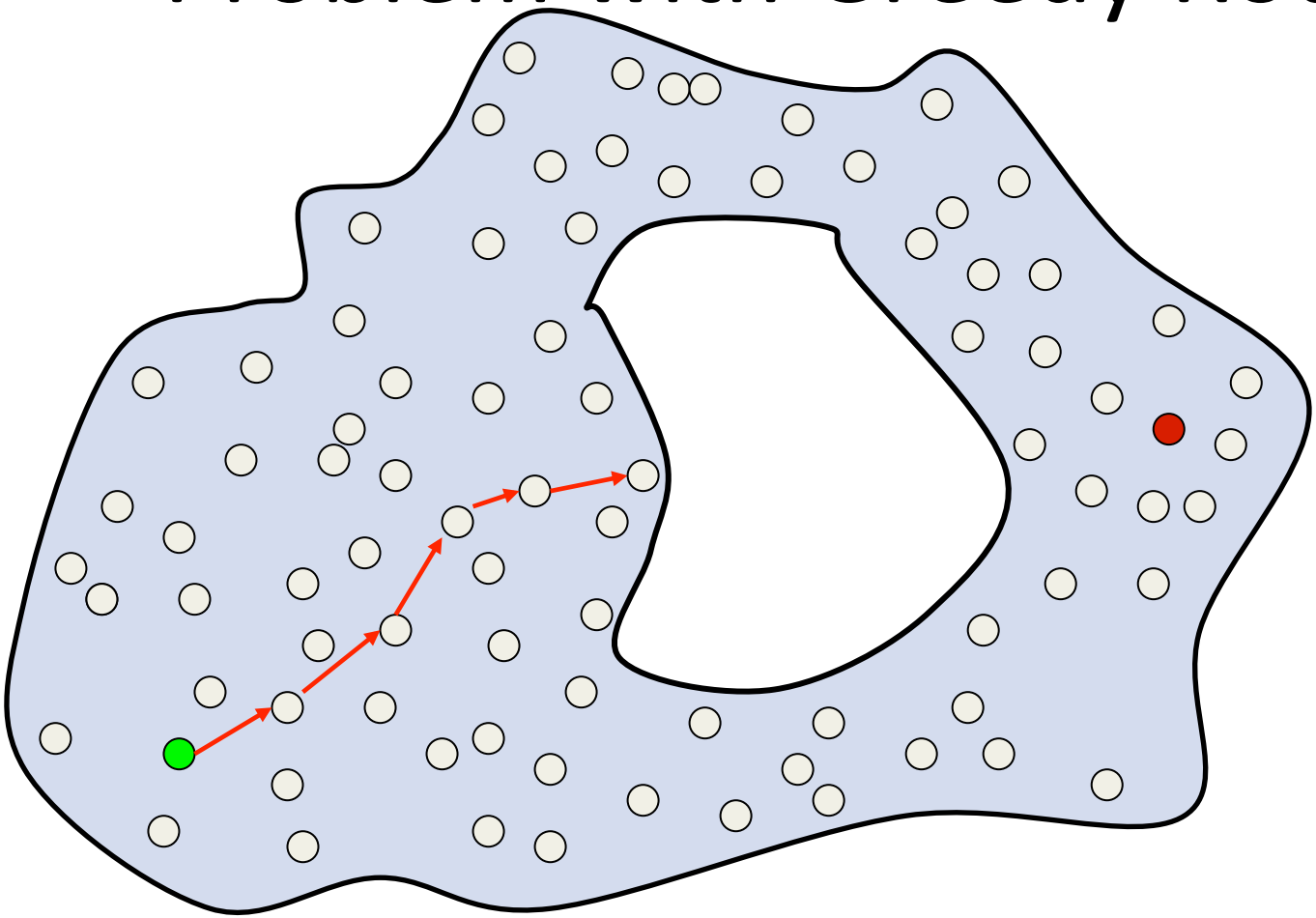


Greedy Routing:
Forward to the neighbor that is nearest to the destination

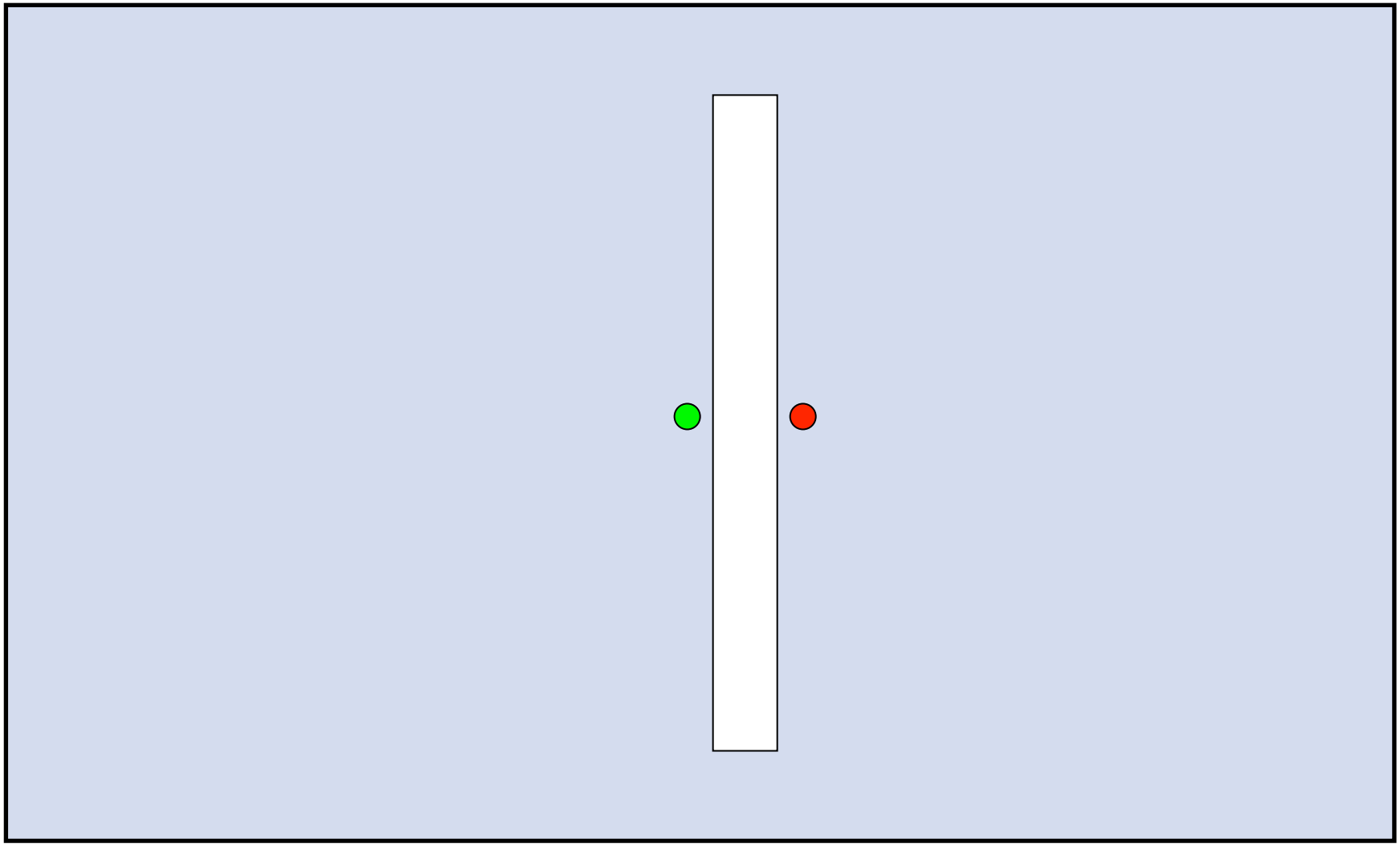
Geographical routing

- Assumptions:
  - Nodes know their own geographical location
  - Nodes know their 1-hop neighbors
  - Routing destinations are specified geographically (a location, or a geographical region)
  - Each packet can hold a small amount of routing information.

Problem with Greedy Routing



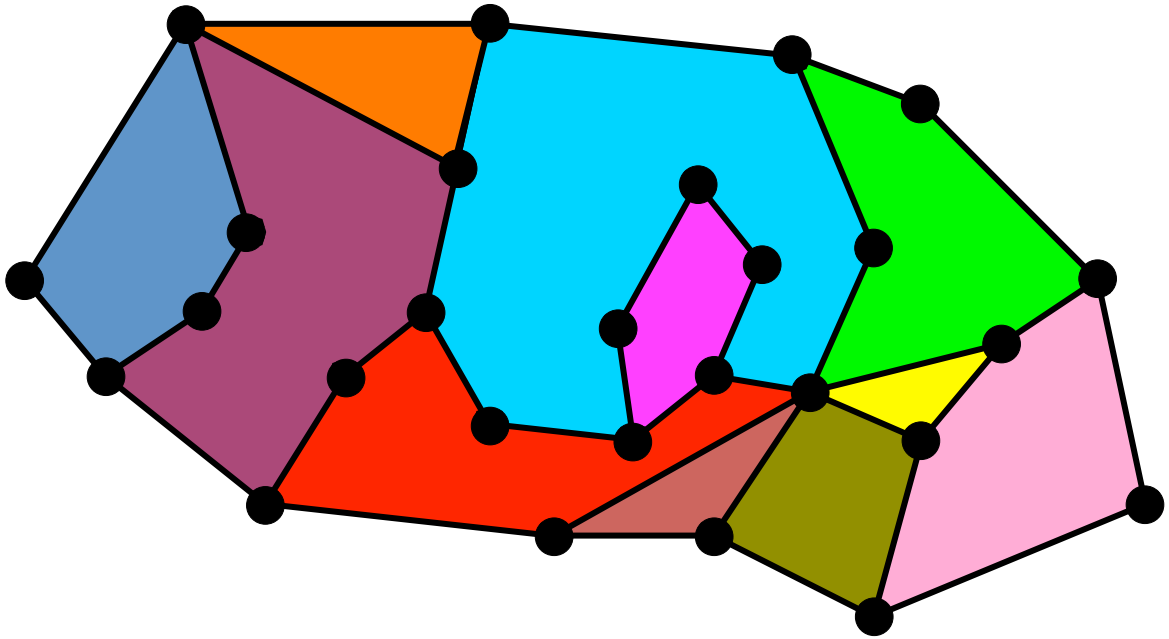
It can get stuck at a local minimum



How will you get around an obstacle?

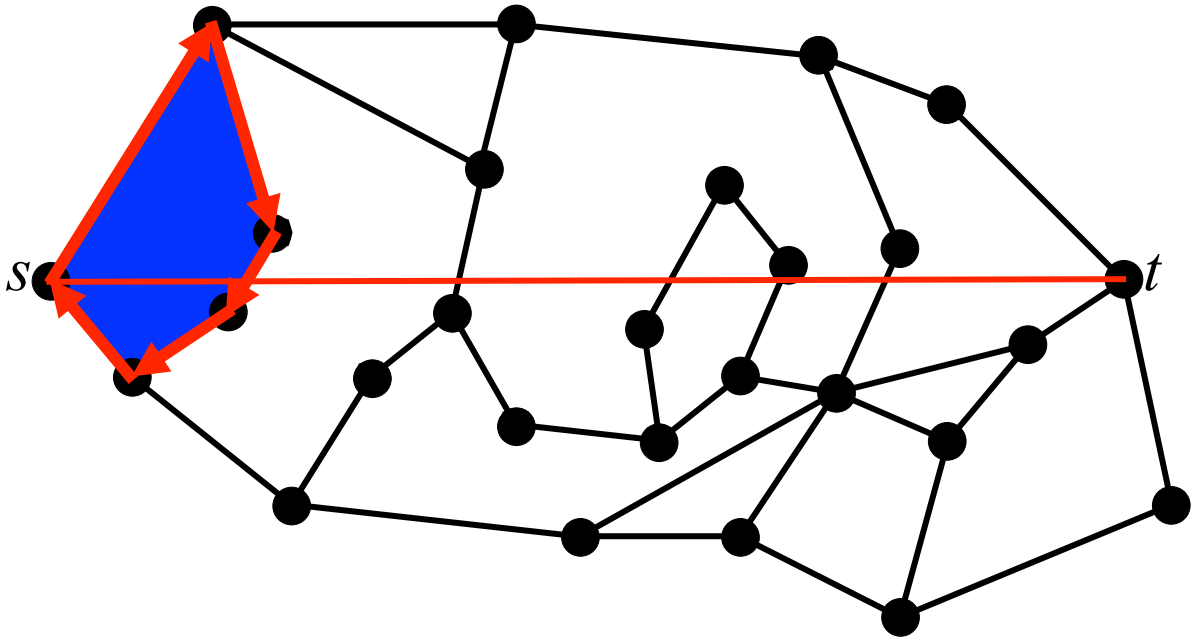
Face Routing

- Use a planar graph
- Keep left hand on the wall, walk until hit the straight line connecting source to destination.
- Then switch to the next face.

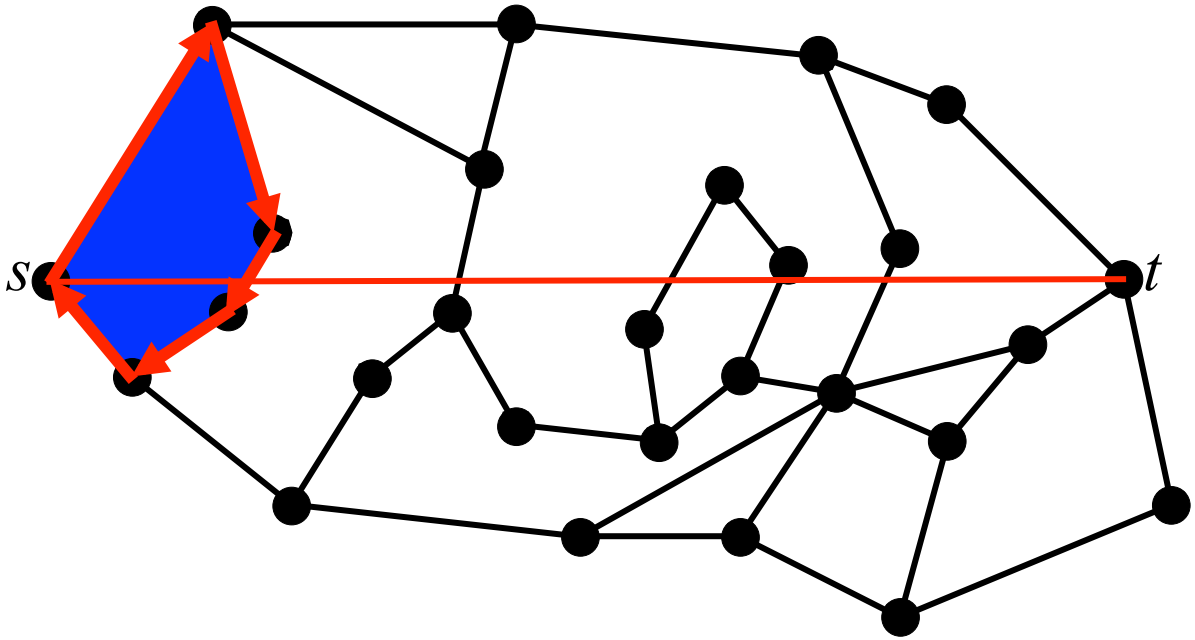


Face Routing

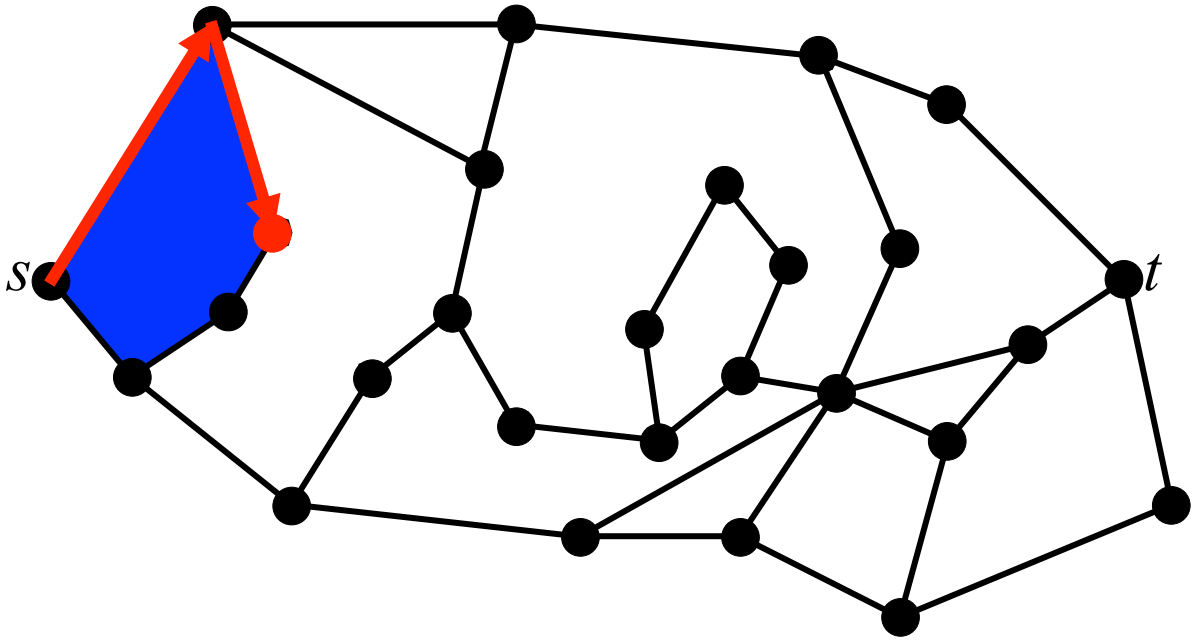
- Keep left hand on the wall, walk until hit the straight line connecting source to destination.
- Then switch to the next face.



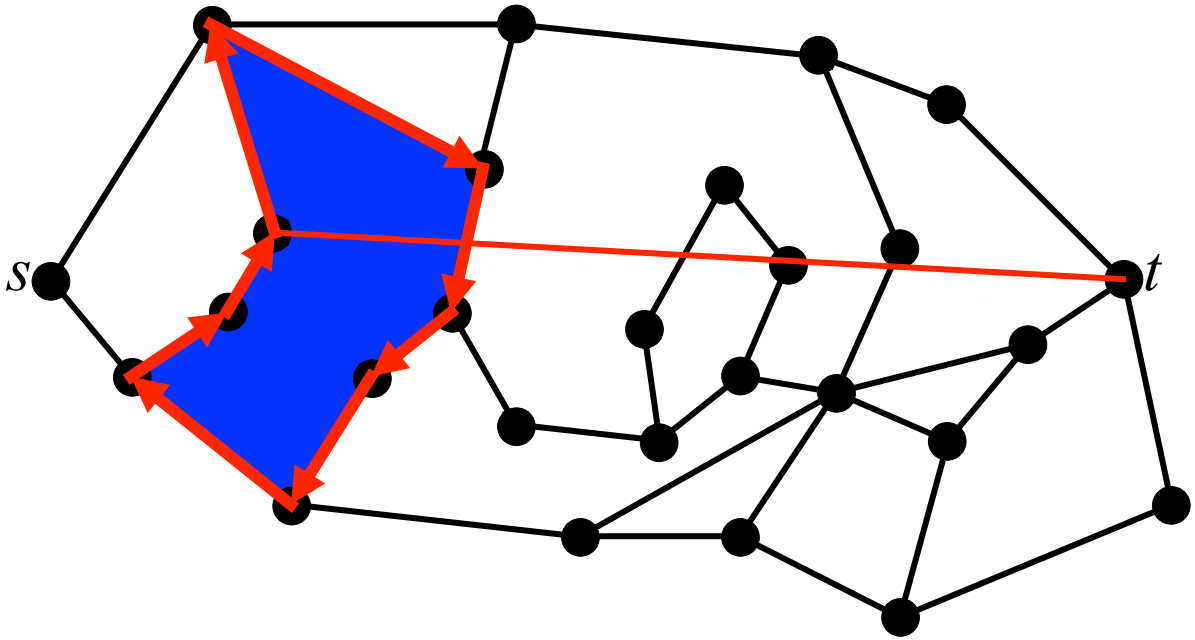
Face Routing



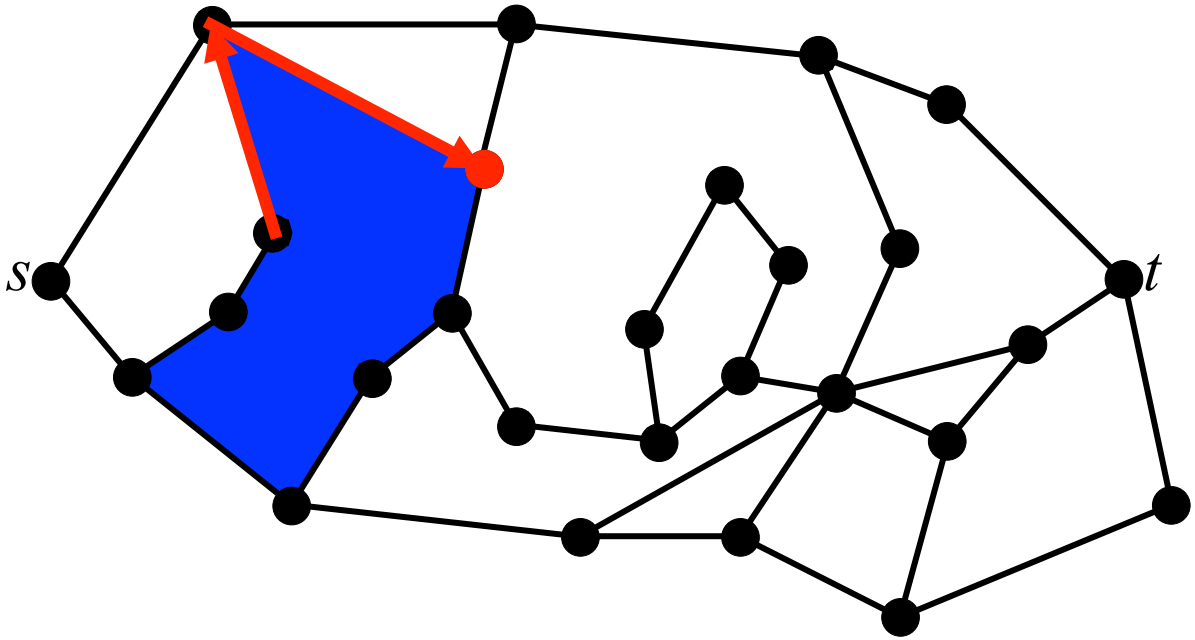
Face Routing



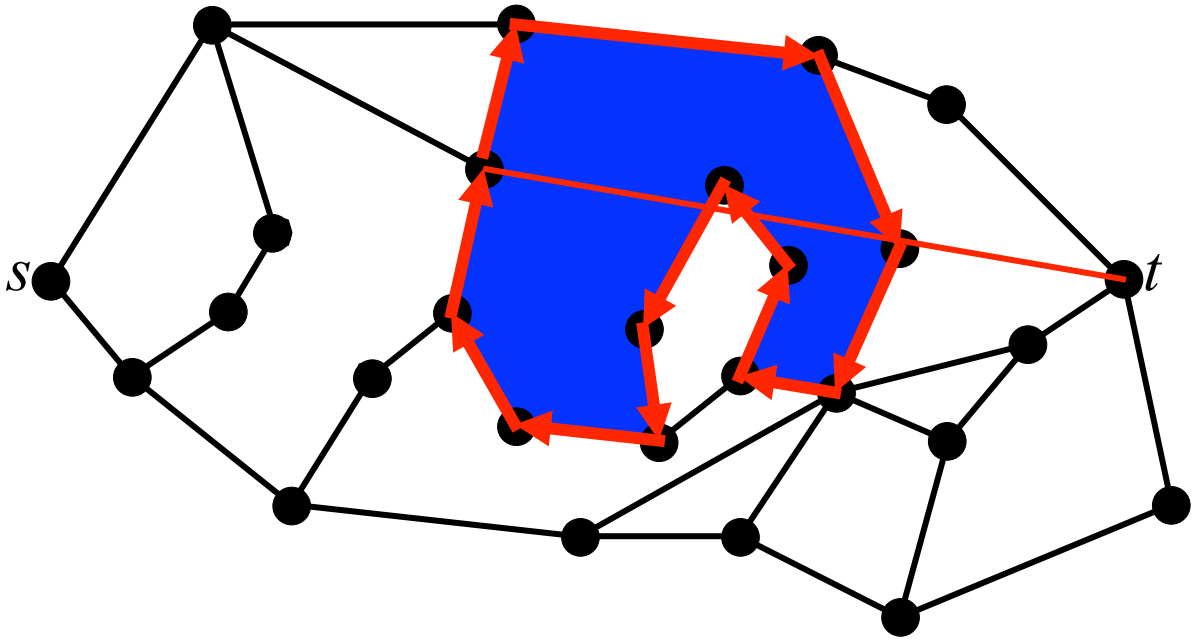
Face Routing



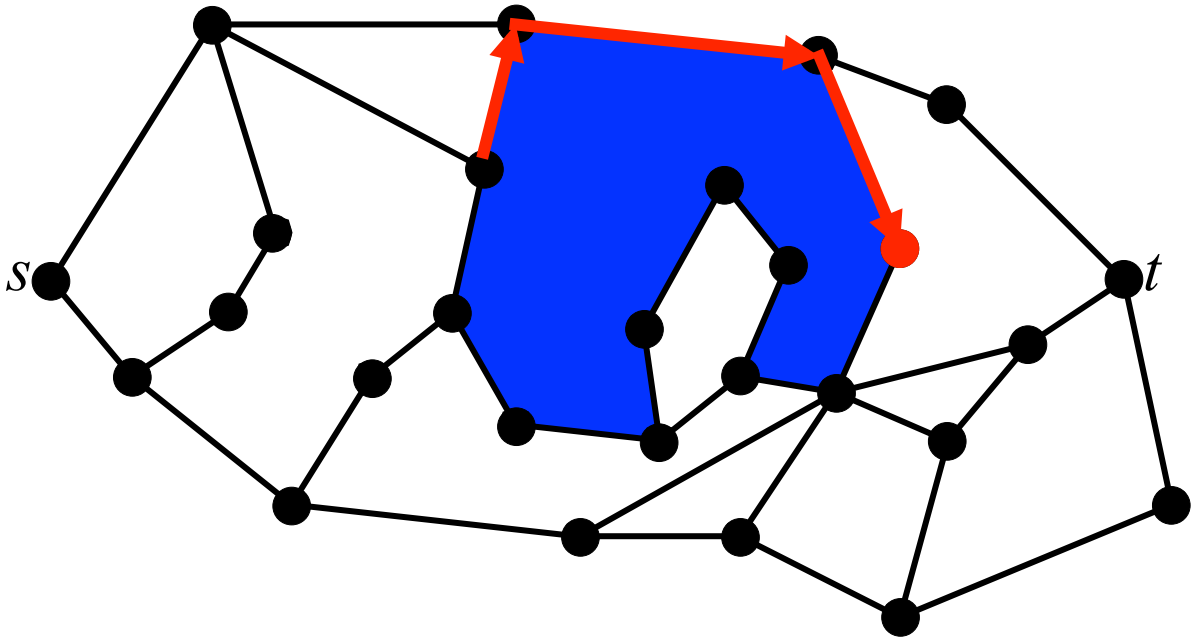
Face Routing



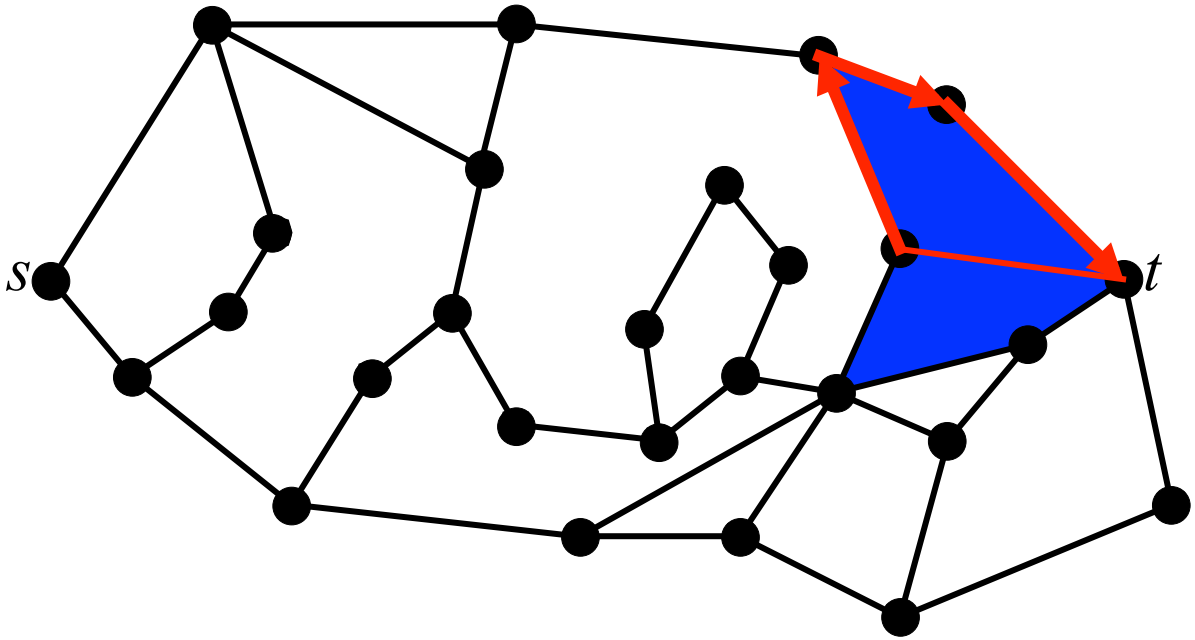
Face Routing



Face Routing

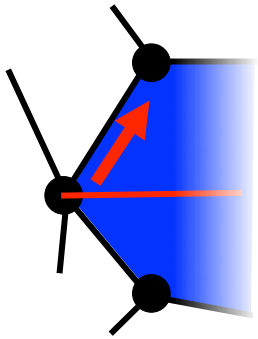


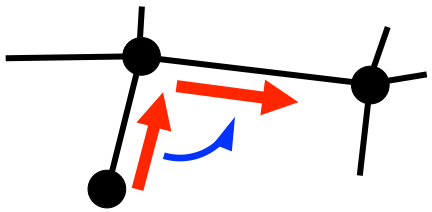
Face Routing



Face Routing Properties

- All necessary information is stored in the message
  - Source and destination positions
  - The node when it enters face routing mode.
  - The first edge on the current face.
- Completely local:
  - Knowledge about direct neighbors' positions is sufficient
  - Faces are implicit. Only local neighbor ordering around each node is needed.

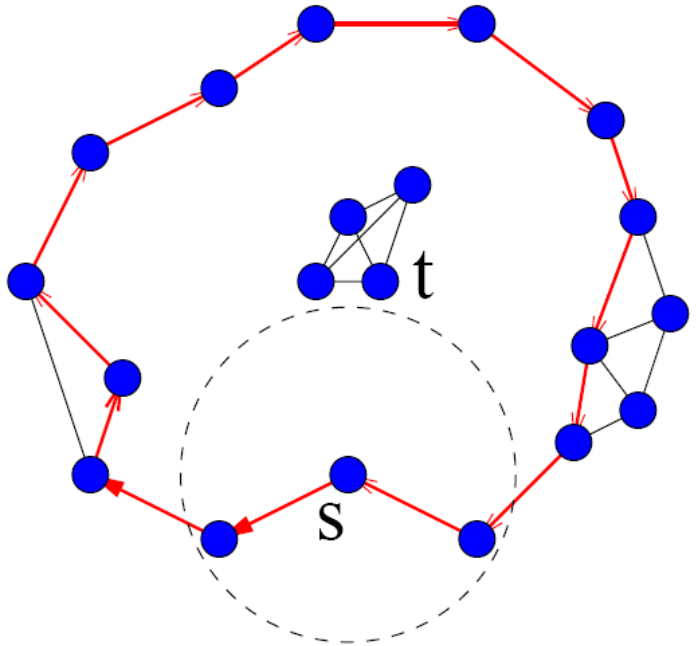




"Right Hand Rule"

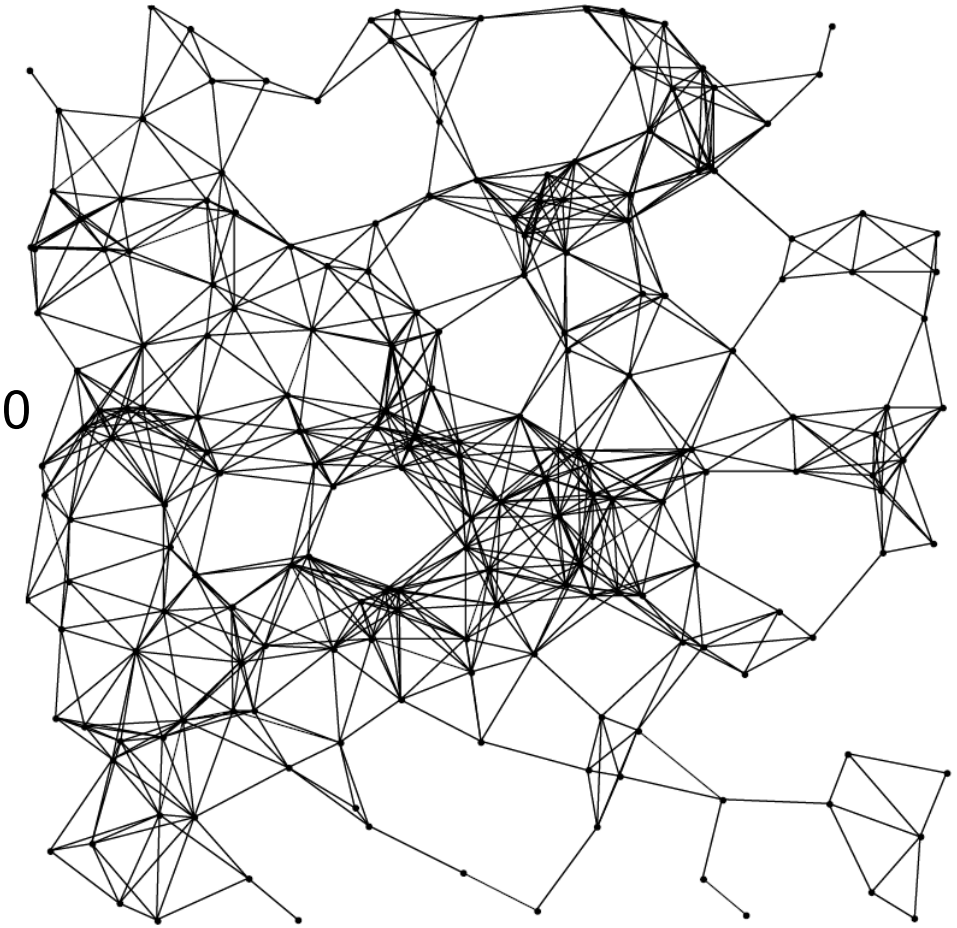
What if the destination is disconnected?

- Face routing will get back to where it enters the perimeter mode.
- Failed – no way to the destination.
- Guaranteed delivery of a message if there is a path.



An example of Unit Disk Graph

200 nodes randomly deployed in a 2000×2000 meters region. Radio range =250meters



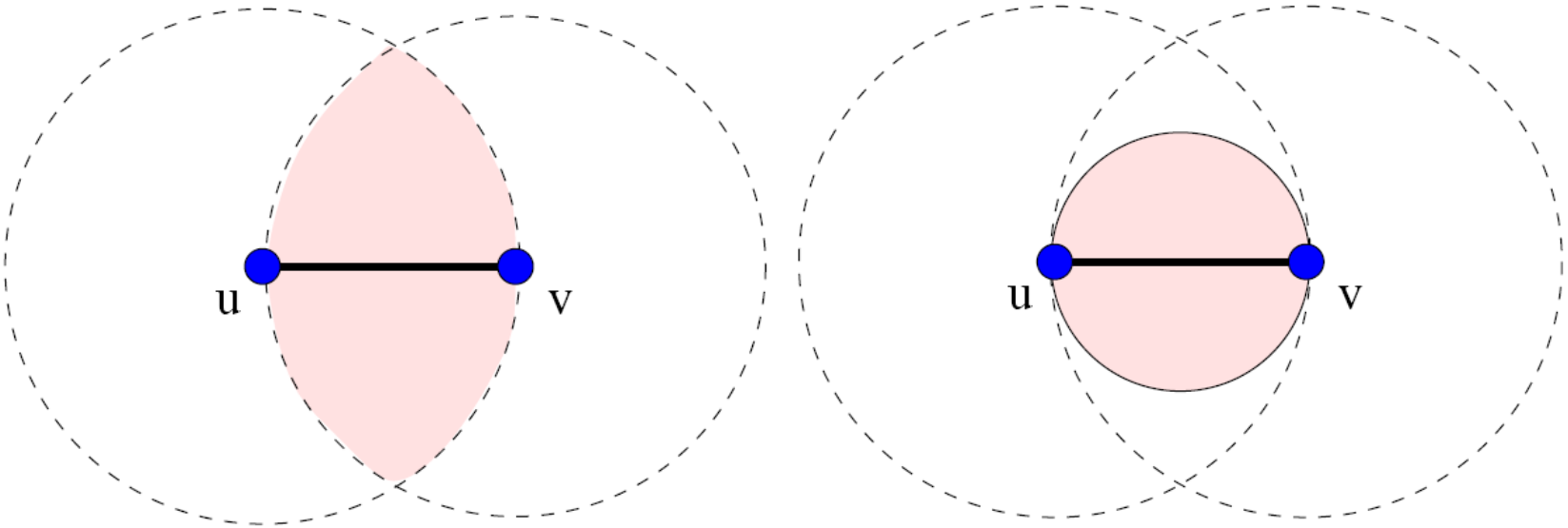
Face routing needs a planar graph…

Compute a planar subgraph of the unit disk graph.

- Preserves connectivity.
- Distributed computation.

Relative Neighborhood Graph and Gabriel Graph

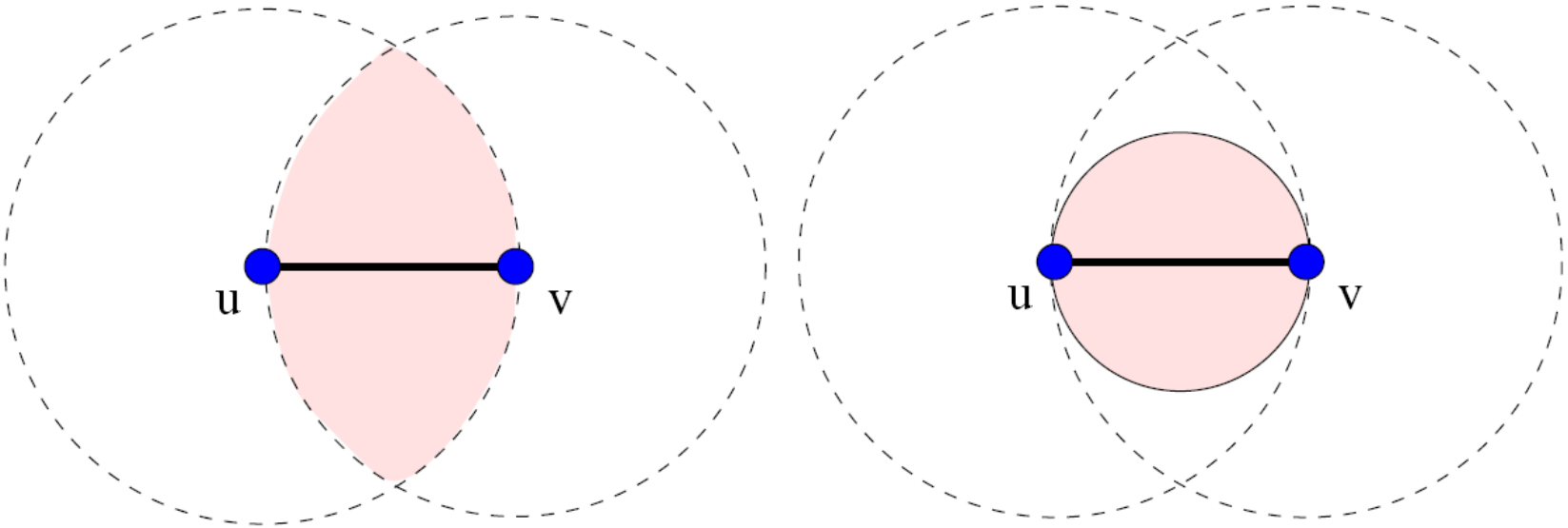
- Relative Neighborhood Graph (RNG) contains an edge uv if the lune is empty of other nodes.
- Gabriel Graph (GG) contains an edge uv if the disk with uv as diameter is empty of other nodes.
- Both can be constructed in a distributed way.



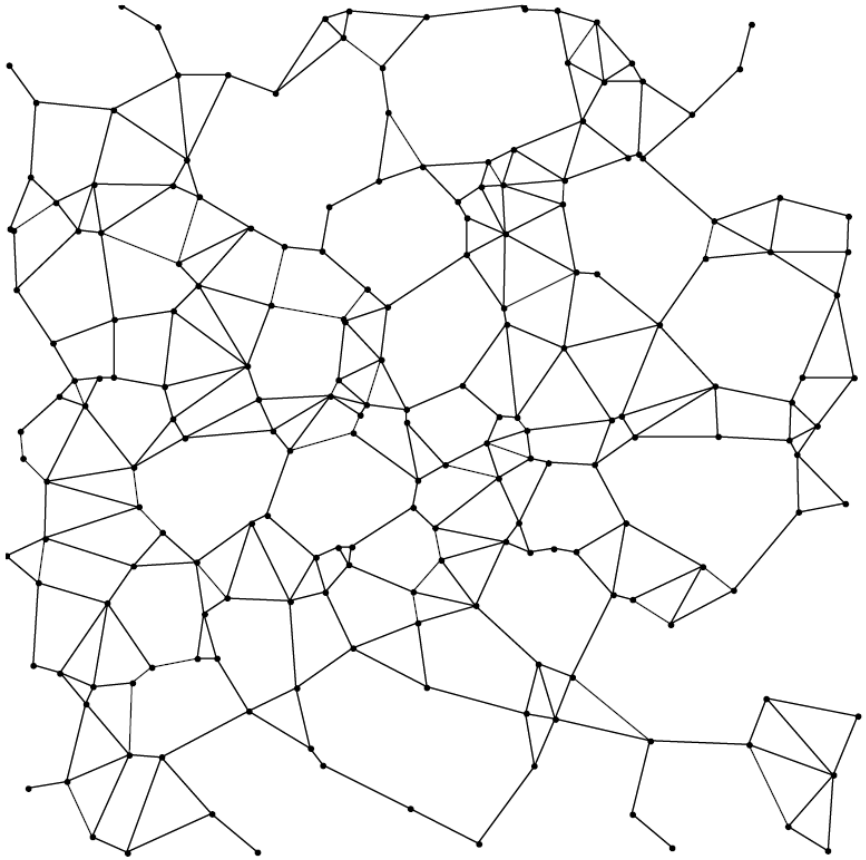
Relative Neighborhood Graph and Gabriel Graph

- Claim: MST $\subseteq$ RNG $\subseteq$ GG

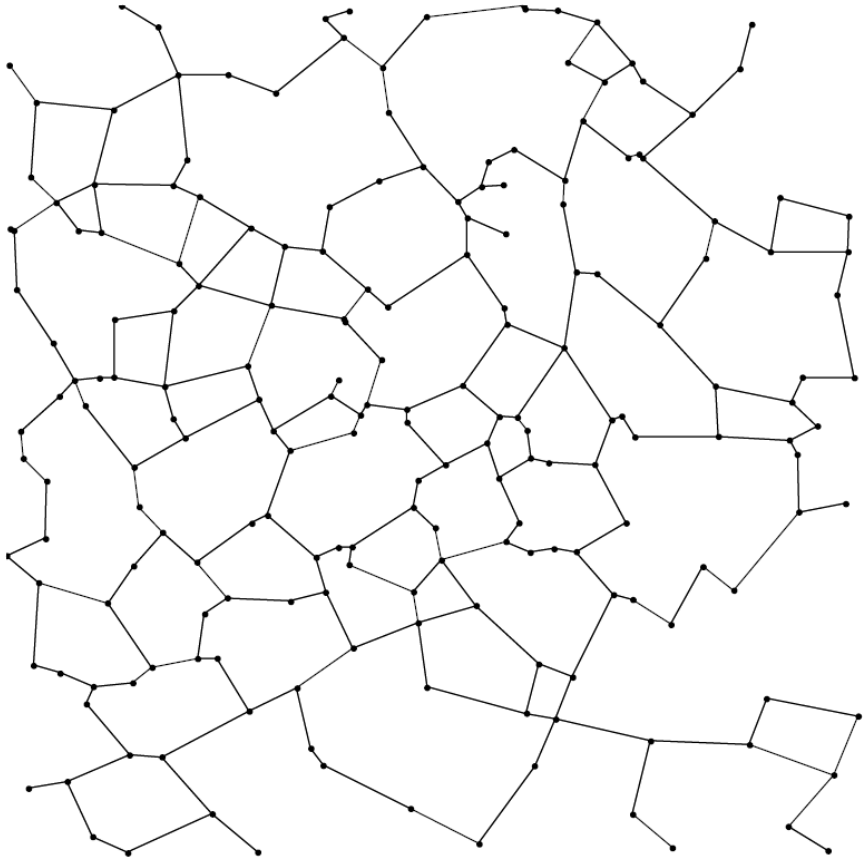
- Thus, RNG and GG are planar and keep the connectivity (MST connects all nodes in UDG).



An example of GG and RNG



GG



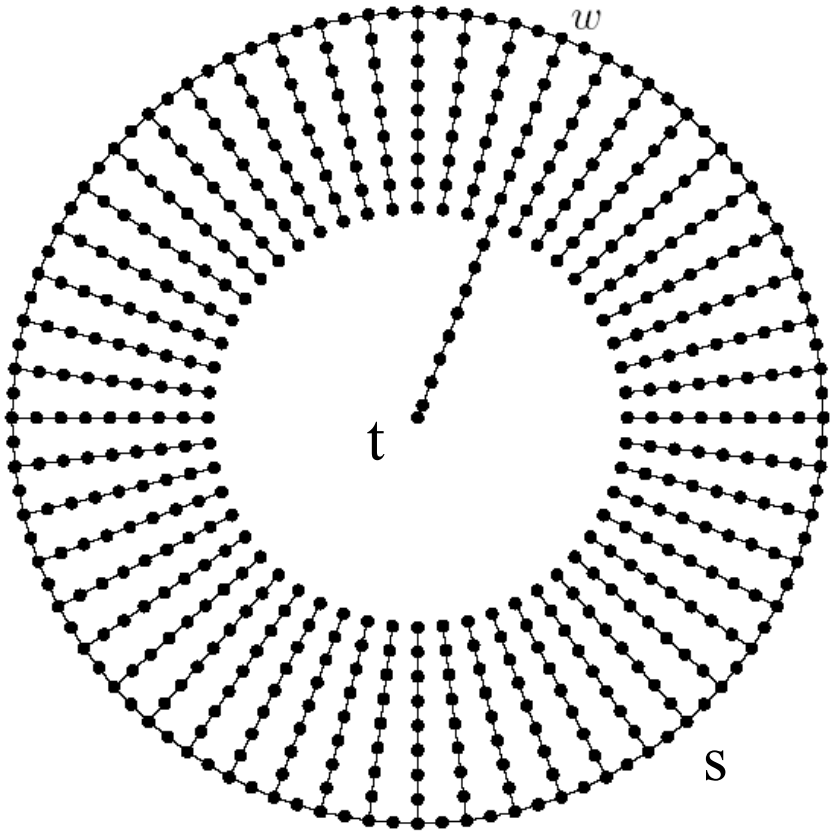
RNG

Two problems remain

- Both RNG and GG remove some edges ➔ a short path may not exist!
- The shortest path on RNG or GG might be much longer than the shortest path on the original network.
- Even if the planar subgraph contains a short path, can greedy routing and face routing find a short one?

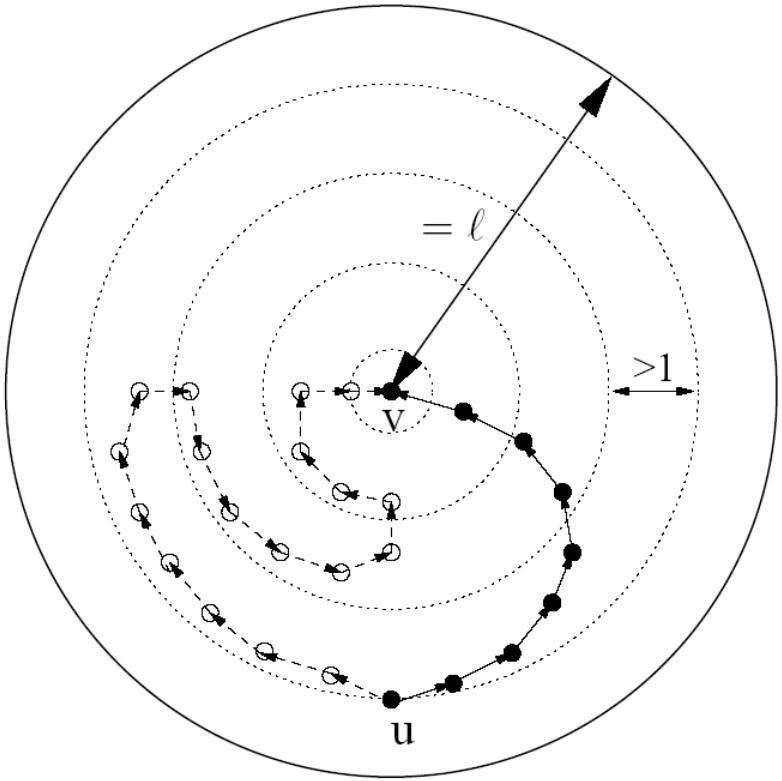
Bad news: Lower bound of localized routing

- Any deterministic or randomized localized routing algorithm can take a path of length $\Omega(k^2)$, if the optimal path has length k.
- The adversary decides where the chain wt is. Since we store no information on nodes, in the worst case we have to visit about $\Omega(k)$ chains and pay a cost of $\Omega(k^2)$.



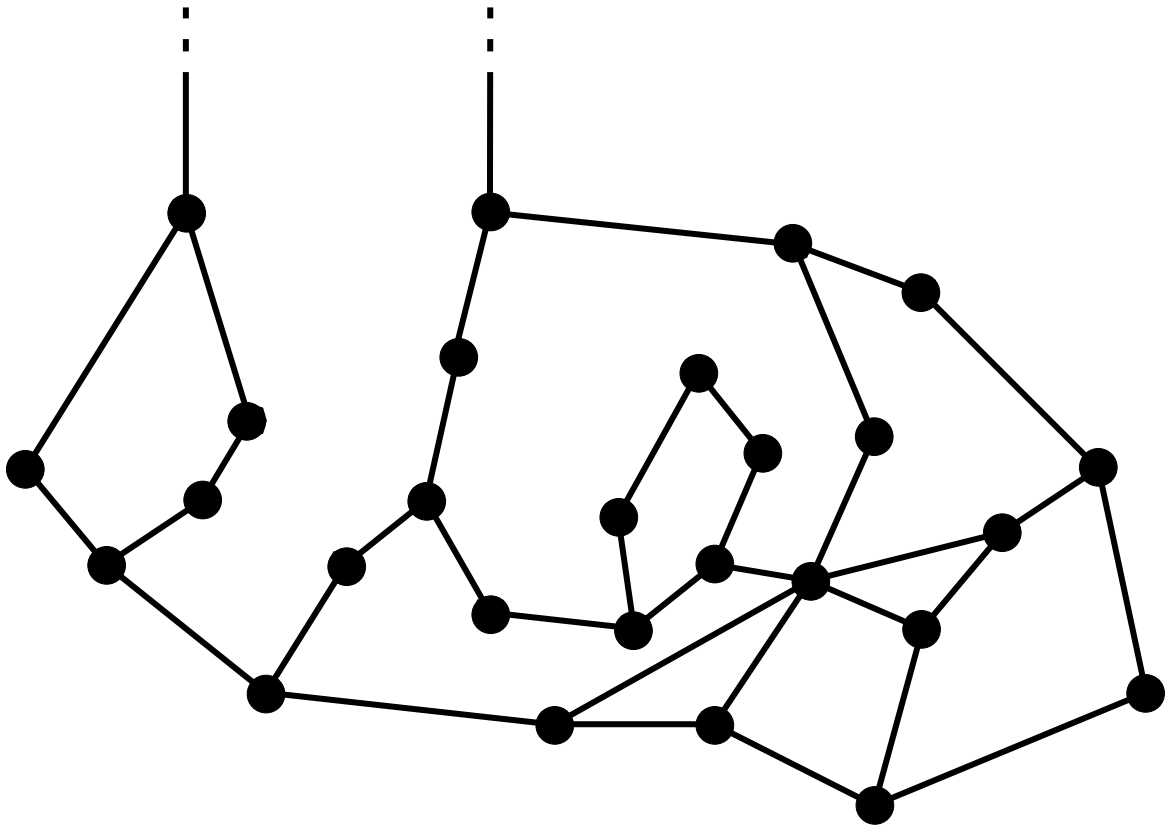
Good news: greedy forwarding is optimal

- If greedy routing gets to the destination, then the path length is at most $O(k^2)$, if the optimal path has length k.
- |uv| is at most k. On the greedy path, every other node is not visible, so they are of distance at least 1 away. By this packing property, there are at most $O(k^2)$ nodes inside a disk of radius k.



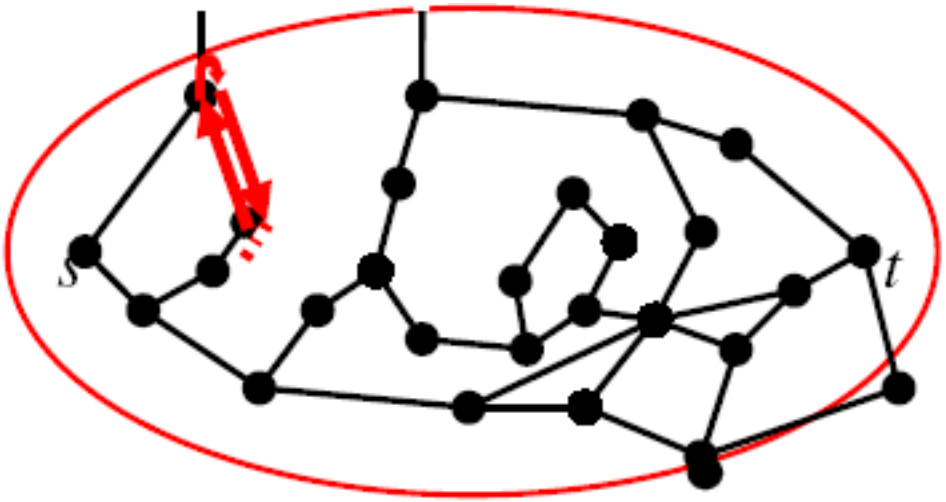
How is face routing?

Face routing can be bad : O(n)



Adaptive Face Routing

- Suppose the shortest path on the planar graph is bounded by L hops.
- Bound the search area by an ellipsoid {x : |xs|+|xt| <= L} never walk outside the ellipsoid.
- Follow one direction, if we hit the ellipsoid; turn back.
- In the worst case, visit every node inside the ellipsoid
- About $O(L^2)$ by the bounded density property.



Adaptive Face Routing

- How to guess the upper bound L?
- Start from a small value say |st|; if we fail to find a path, then we double L and rerun adaptive face routing.
- By the time we succeed, L is at most twice the shortest path length k. The number of phases is O(log k).
- Total cost = $O(\Sigma_i (k/2^i)^2) = O(k^2)$. | asymptotically

Combine Greedy and Face Routing

- Route greedy till it gets stuck at some node p
- Switch to face routing
- When at some node q which is nearer to destination than p, switch back to greedy
- Called Greedy-Face-Greedy strategy

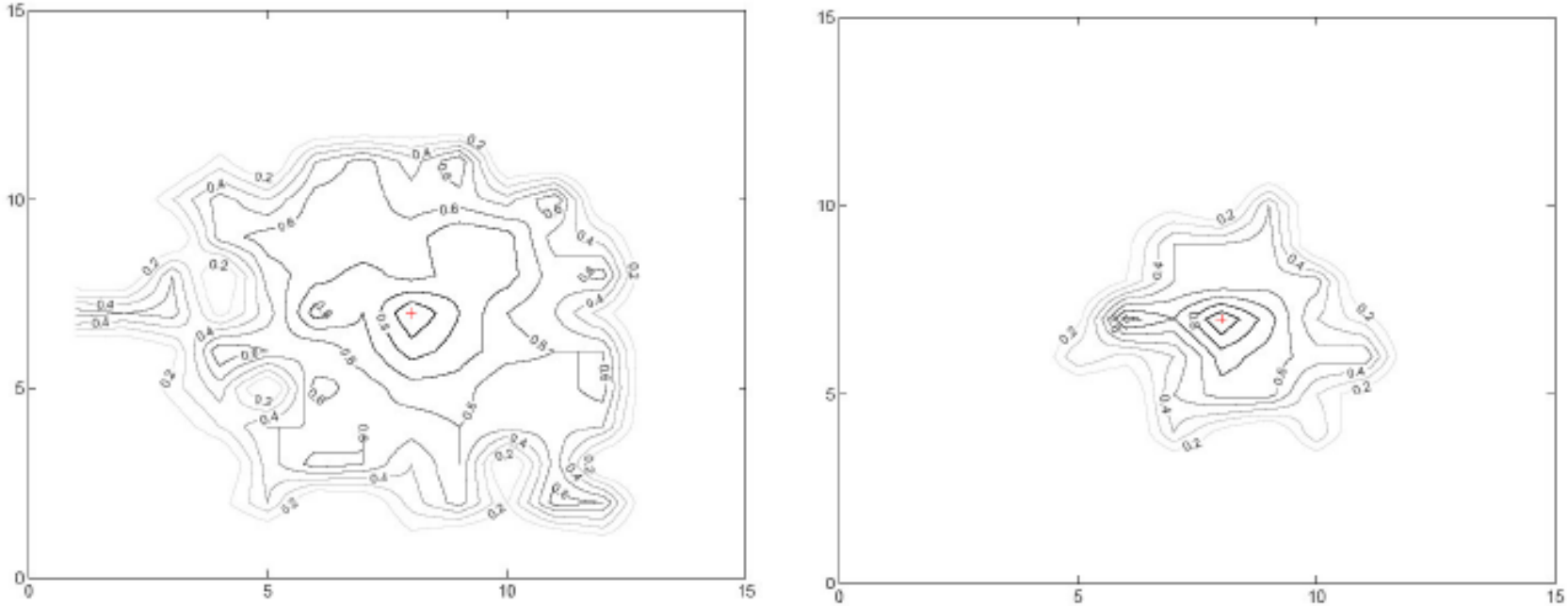
Many Variations on these strategies

- [Bose, et.al 99] Routing with guaranteed delivery in ad hoc wireless networks.
- [Karp and Kung 00] GPSR: Greedy Perimeter Stateless Routing for Wireless Networks.
- [Kuhn, et.al 02] Asymptotically optimal geometric mobile ad hoc routing.
- [Kuhn, et.al 03a] Worst-case optimal and average-case efficient geometric ad hoc routing.
- [Kuhn, et.al 03b] Geometric ad hoc routing: of theory and practice.
- [Kim, et.al 05b] Geographic Routing Made Practical.
- [Kim, et.al 05a] On the Pitfalls of Geographic Face Routing.
- [Frey, et.al 06] On Delivery Guarantees of Face and Combined Greedy-Face Routing in Ad Hoc and Sensor Networks.

Lesson: Do it carefully!!

The protocols in practice

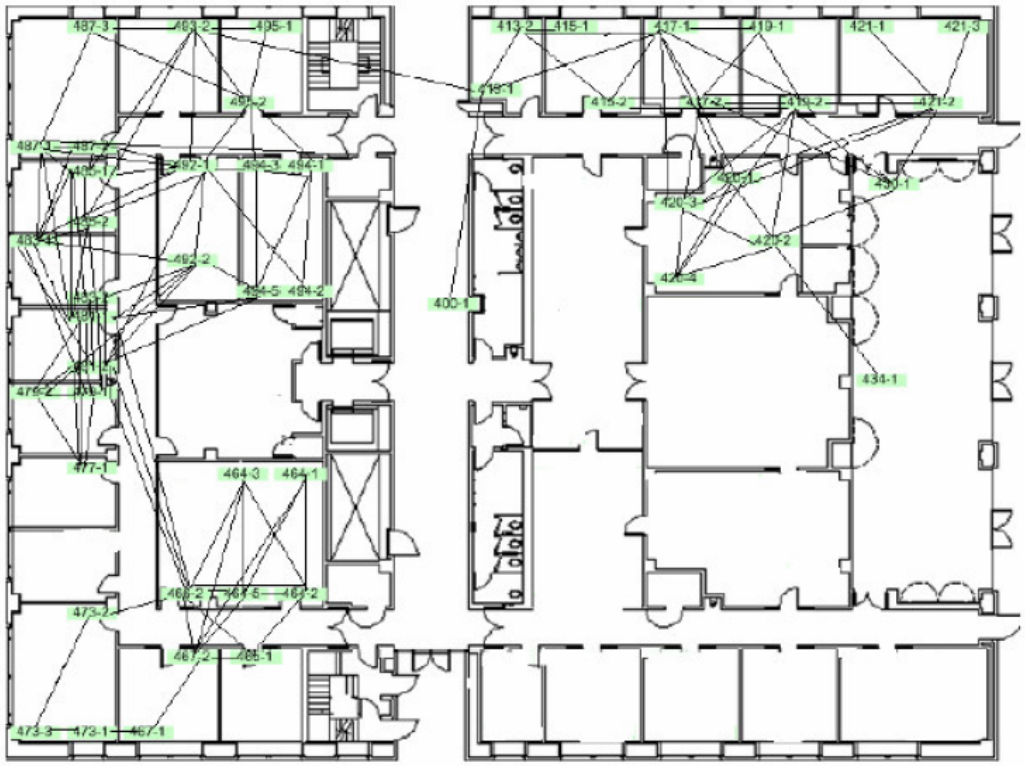
- Locations are not always known
- Communication ranges are not disks
  - Planar graph construction fails



Picture : Govindan et al.

Experiment

- GPSR succeeds in 68.2% directed pairs



A 50-node testbed at Intel Berkeley Lab

Alternative:

- Cross link detection protocol
  - Detects crossing links and appropriately does routing on non-planar graphs.

- Kim, Y.-J., Govindan, R., Karp, B., and Shenker, S., **On the Pitfalls of Geographic Face Routing**, DIAL-M-POMC'05.
- Kim, Y.-J., Govindan, R., Karp, B., and Shenker, S., **Geographic Routing Made Practical**, NSDI 2005.

Problem: How to find the data?

- A tourist in a park asks
- “Where is the elephant?”
- Out of all the sensors/cameras which one is close to an elephant?

Data centric routing

- Traditional networks try to route to an IP address
- Find path to the node with a particular ID
- But what if we try to find data, not specific nodes?
- After all, delivering data is the ultimate goal of routing and networks
- Data centric storage
  - Storage depends on the data (elephant, giraffe,song…)
- Data centric routing (search)
  - Route to the data

Distributed Database

- Information Producer
  - Can be anywhere in the network
  - May be mobile
  - Many producers may generate data of the same type
- User or Information Consumer
  - Can be anywhere
  - May be many

Distributed Database: Challenges

- Consumer does not know where the producer is, and vice versa
- Need to search : Must be fast, efficient

Basic methods:

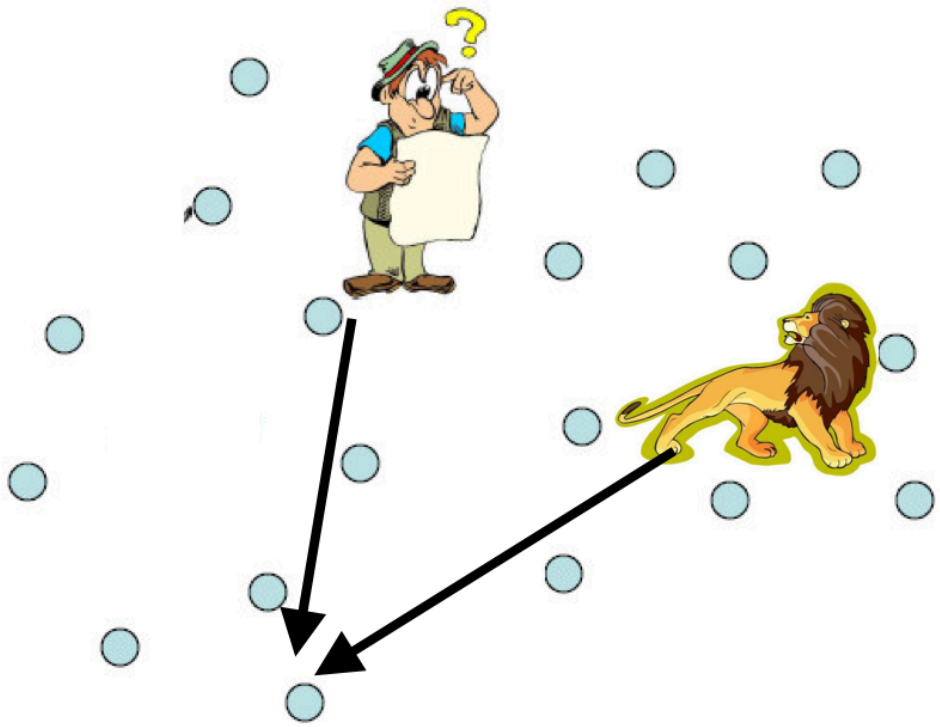
- Push: Producer disseminates data
- Pull: Consumer looks for the data
- Push-pull: Both producer, consumer search for each-other

Distributed hash tables

- Use a hash on the data: **h**(song1.mp3) = node#26
- Anyone that has song1.mp3 informs node#26
- Anyone that needs Song1.mp3 checks with node#26
- Used in peer to peer systems like Chord, pastry etc

Geographic Hash Tables

- Content based hash gives coordinates:
  - **h**(lion) = (12, 07)
- Producer sends msg to (12, 07) by geographic routing and stores data
- Consumer sends msg to (12, 07) by geographic routing and gets data

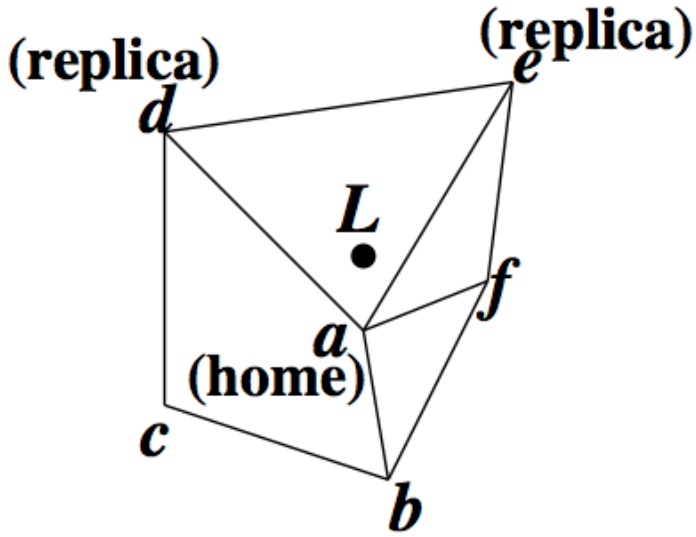


GHT

- What if there is no sensor at (12, 07) ?
- What if geographic routing gets stuck before it gets to (12, 07) ?

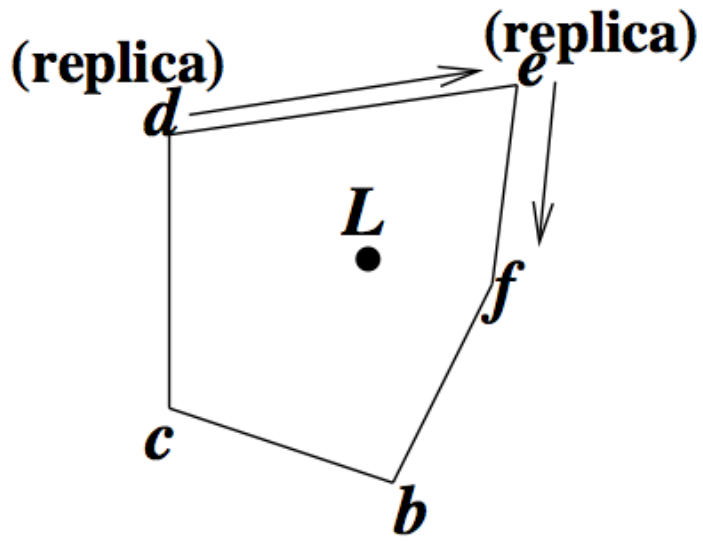
GHT

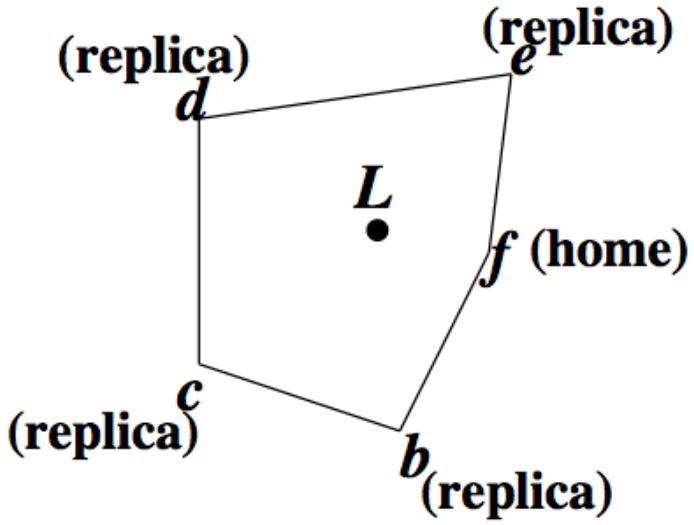
- L = hash location
- *ade* : face that contains L
- GHT stores copies of data on a,d,e
- a is in charge : home node : makes sure data is fresh, all nodes on perimeter has data



Fault handling

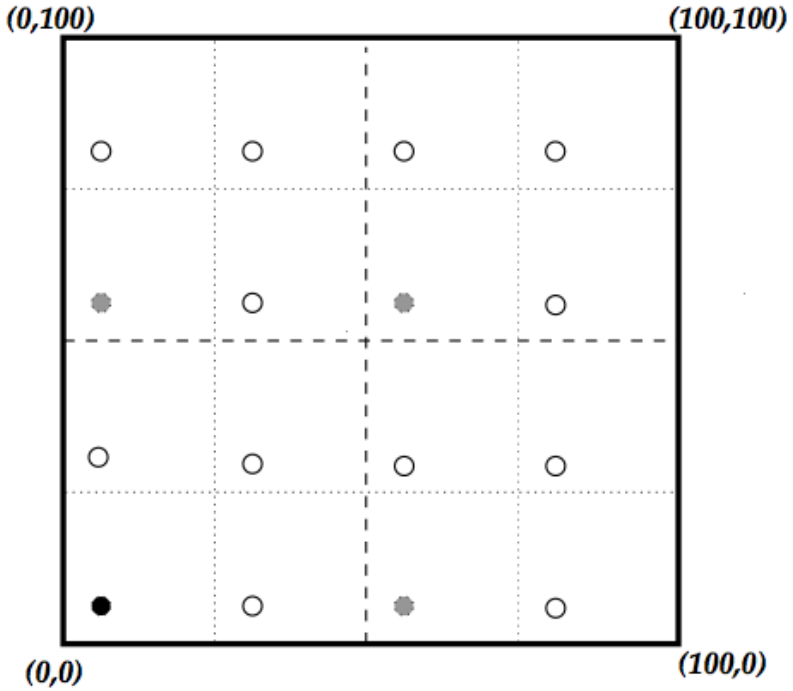
- What if home node a dies?
- Replicas have a timer that triggers a new check
- A new node becomes home





Fault and load handling

- A few nodes have all the responsibility: too much load, susceptible to failure
- Hash location is replicated at each level of a quadtree down to some fixed depth



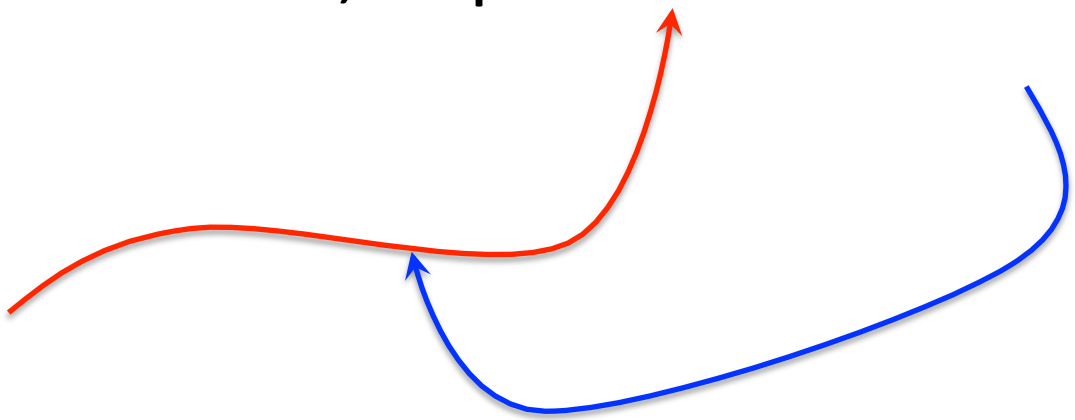
GHT

- Advantages
  - Simple
  - Handles load balancing and faults
- Disadvantages
  - Not distance sensitive: everyone has to go to hash node even if producer and consumer are close
  - Overloads boundaries of holes
  - If a data is queried or updated often, that node has a lot of traffic – bottleneck

Rumor Routing

- Producer: Send data along a curve or random walk, leave data or pointers on nodes

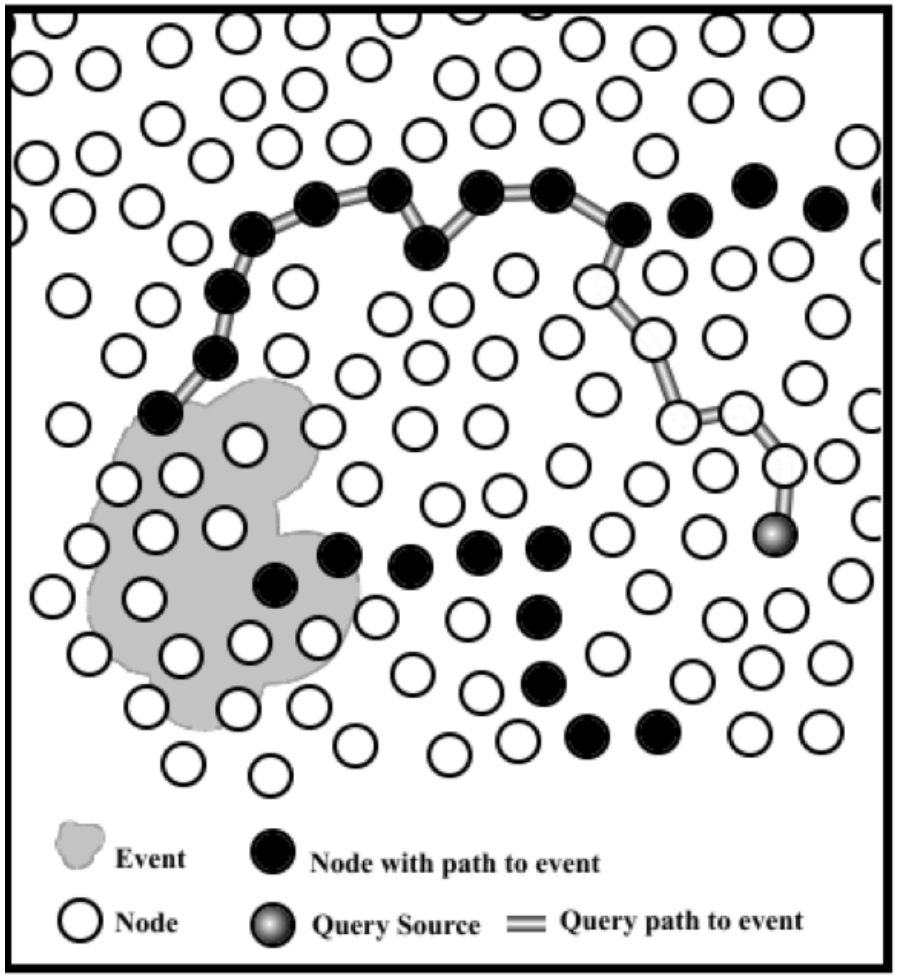
- Consumer: Route along another curve or random walk, hope to meet data or pointer

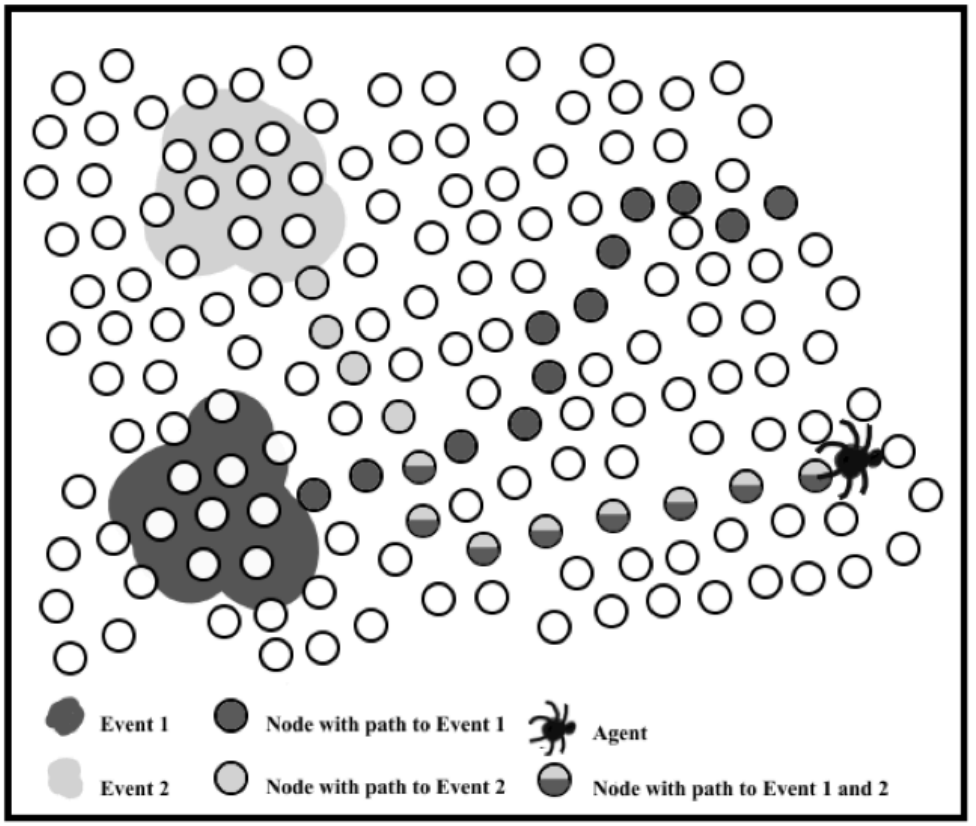


Rumor routing

- Each node maintains a list of events
- Adds events as they happen

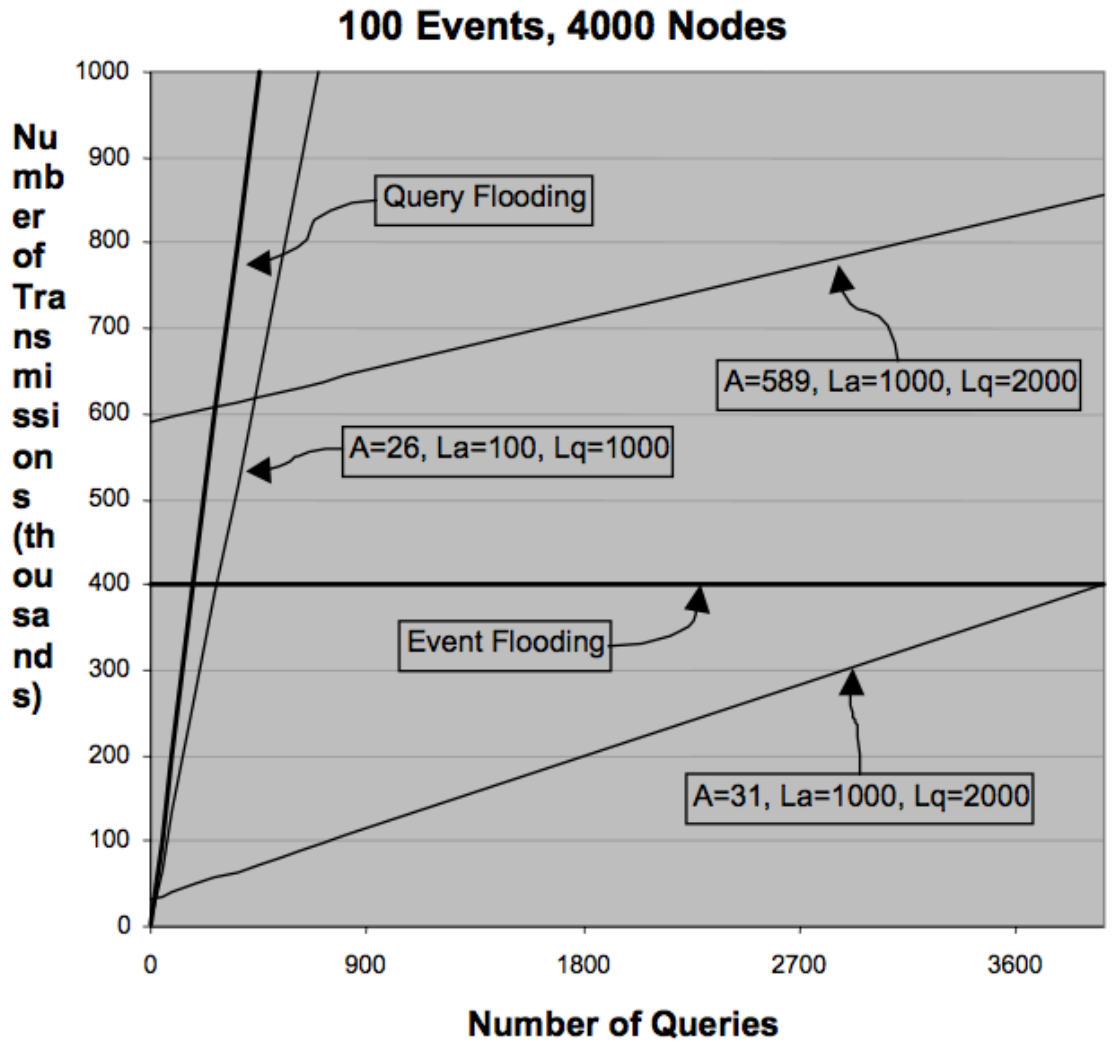
- Agents: Packets that carry events in the network
  - Aggregate events of each node they pass through
- Agents move in random walk. From 1-hop neighbors select one that has not been visited recently


Event
Node with path to event
Node
Query Source
Query path to event


Event 1
Node with path to Event 1
Agent
Event 2
Node with path to Event 2
Node with path to Event 1 and 2

Simulation

- 200x200 field, communication radius 5
- A = #agents, La=agent TTL, Lq=query TTL



Problems

- Each agent carries list of events : can be large
- Random walk can take a long time to reach far away regions
- Harder to analyze for the specific algorithm in the paper
- Inefficient: may visit same nodes many times

Double rulings

- Store data on a curve, like rumor routing.
- Not a random curve, a more structured approach, like GHT
- The curve depends on the data

Rectilinear Double Ruling

- Producer stores data on horizontal lines
- Consumer searches along vertical lines
- Correctness: every horizontal line intersects every vertical line
- Distance sensitive: q finds p in time O(d) where d = |pq|. How?