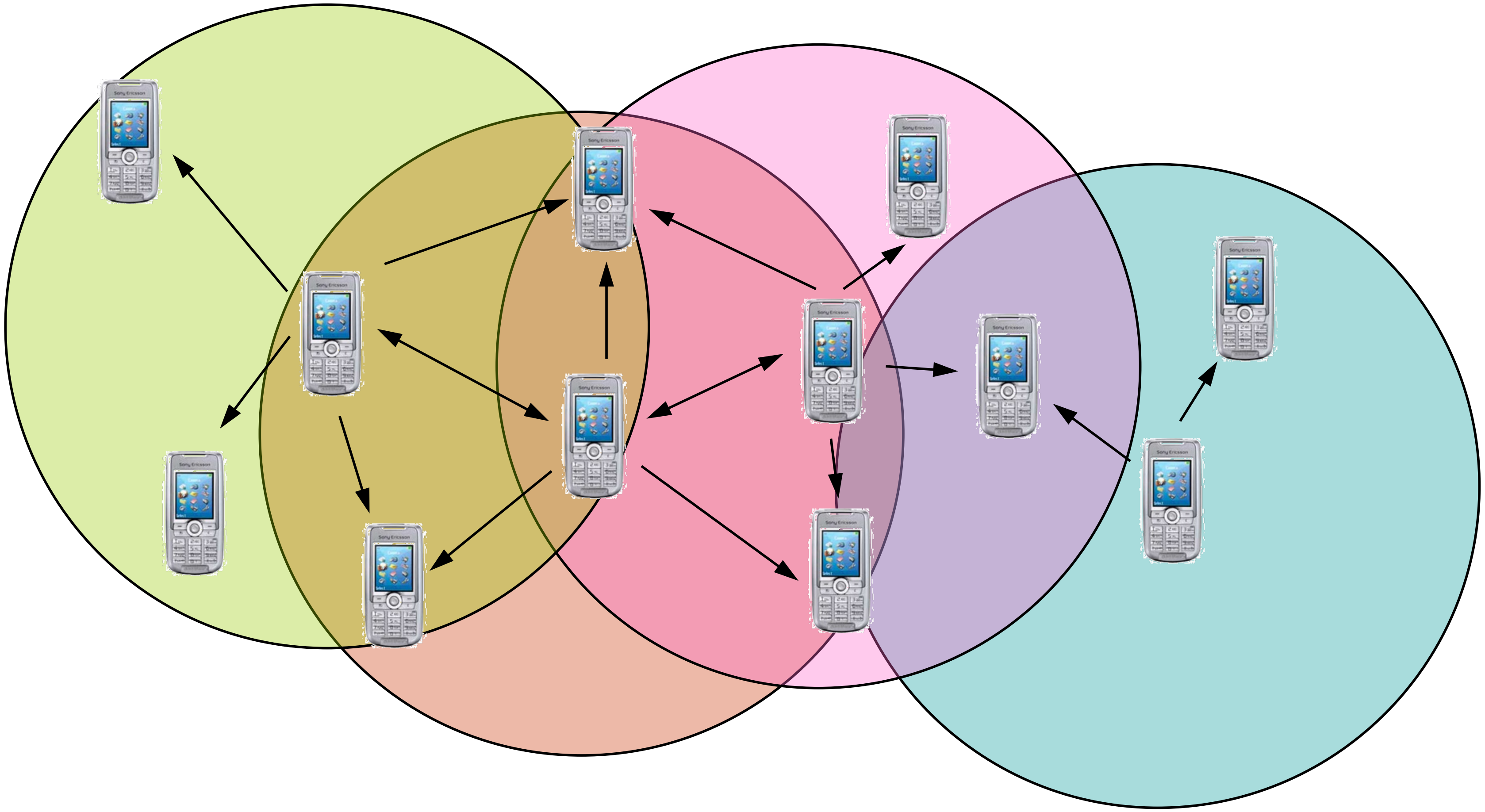
Routing in Wireless Networks

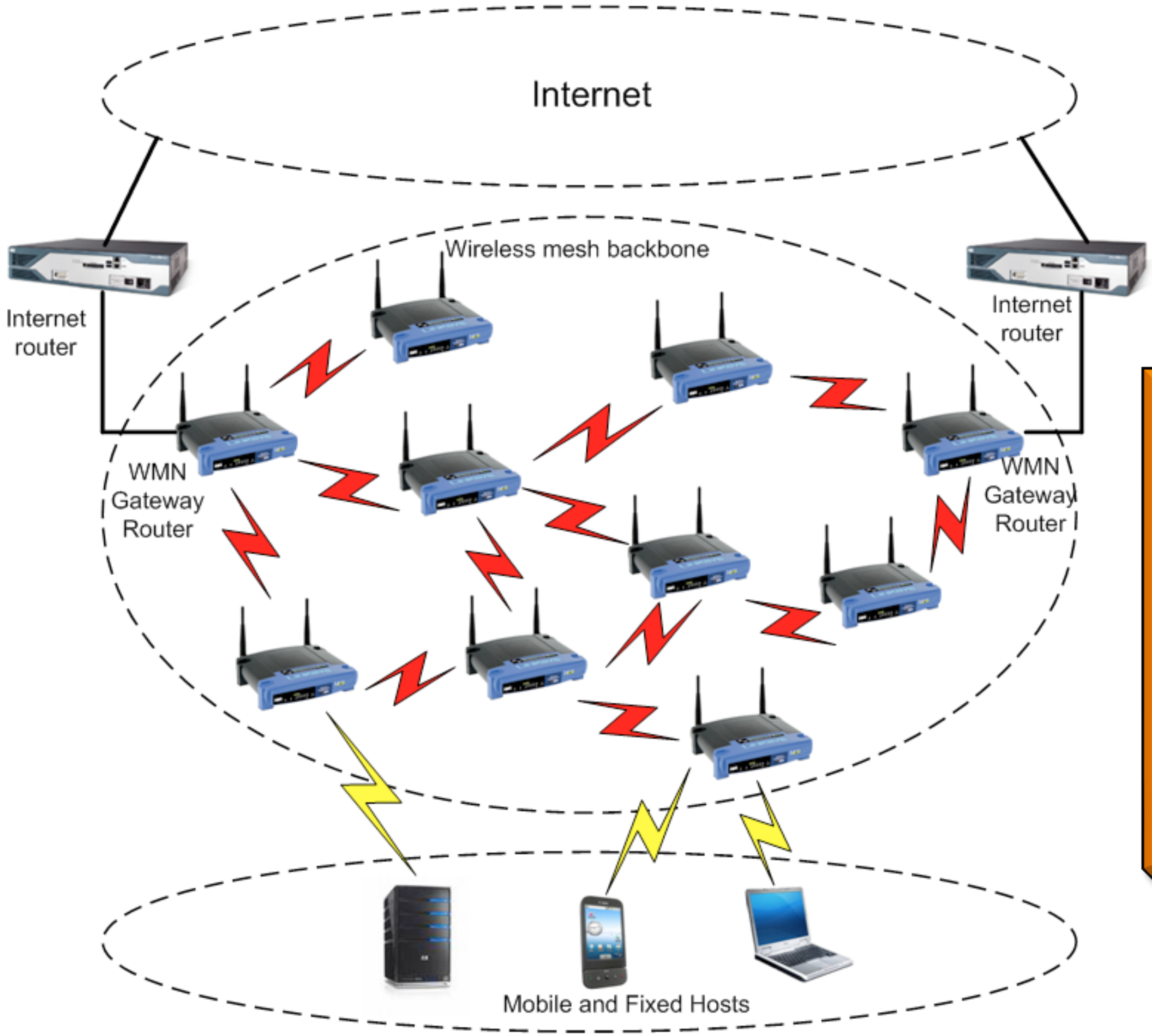
Prof. Congduc Pham
http://www.univ-pau.fr/~cpham
Université de Pau, France
Congduc.Pham@univ-pau.fr

Ad-Hoc networks

- (Mobile) Adhoc NETworks



wireless mesh networks



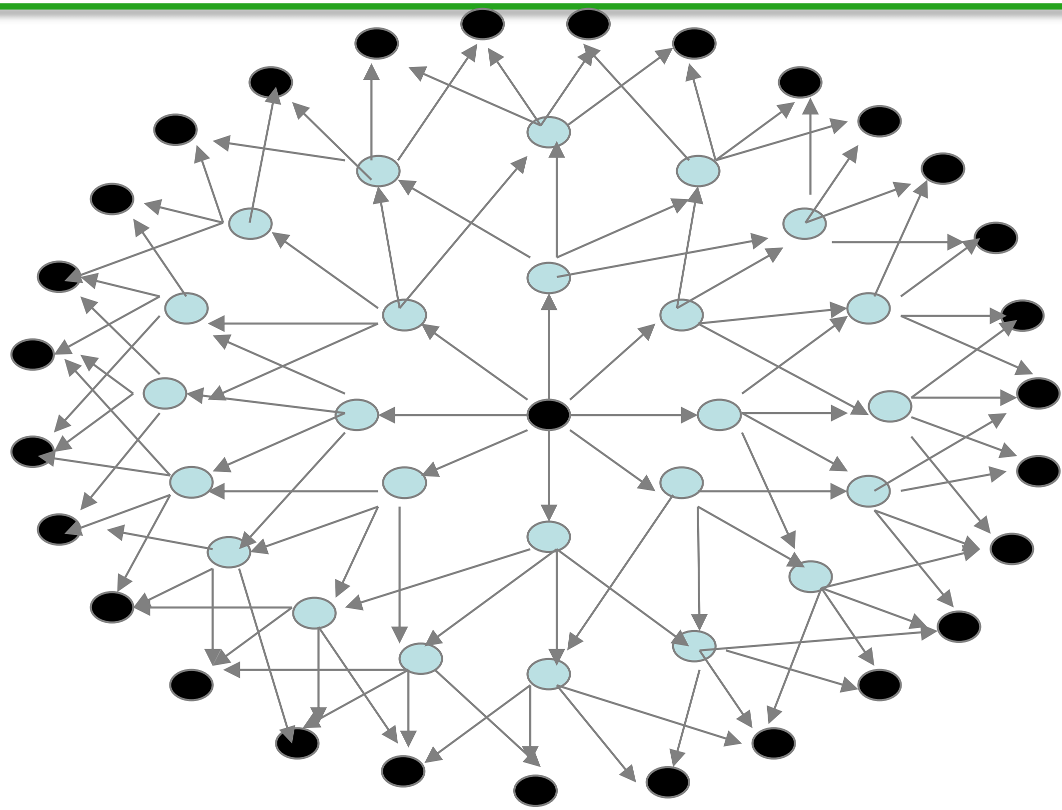
Cognitive, opportunistic, multi-channel radio for large-scale wireless infrastructures

Conventional wired routing limitations

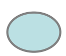
- Distance Vector (e.g., Bellman-Ford, BGP):
  - Tables grow linearly with # nodes
  - routing control O/H linearly increasing with network size
  - convergence problems (count to infinity); potential loops (mobility?)
- Link State (e.g., OSPF):
  - link update flooding O/H caused by network size and frequent topology changes
- CONVENTIONAL ROUTING DOES NOT SCALE TO SIZE AND MOBILITY

Flooding in Link State Routing

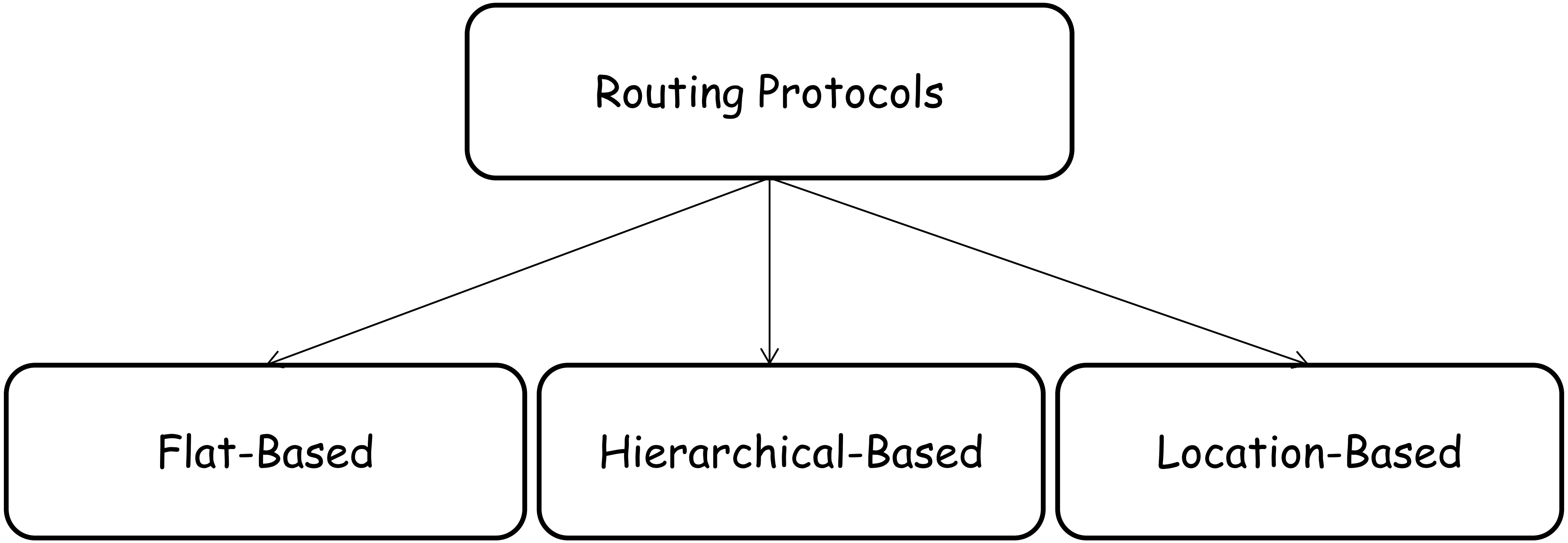
- In LSR protocol a lot of control msg unnecessary duplicated



24 retransmissions to diffuse a message up to 3 hops

 Retransmission node

Network Structure Categorization



Routing approach

- Proactive protocols
  - Traditional distributed shortest-path protocols
  - Maintain routes between every host pair at all times
  - Based on periodic updates; High routing overhead
- Reactive protocols
  - Determine route if and when needed
  - Source initiates route discovery
  - Example: DSR (dynamic source routing)

Routing Operation

1. Multipath routing

- Increases fault tolerance
- Sophisticated case: have back up paths

2. Query-based routing

- Query transmitted and the data is sent back

3. Negotiation-based routing

- High-level data description
- Elimination of redundant data transmission

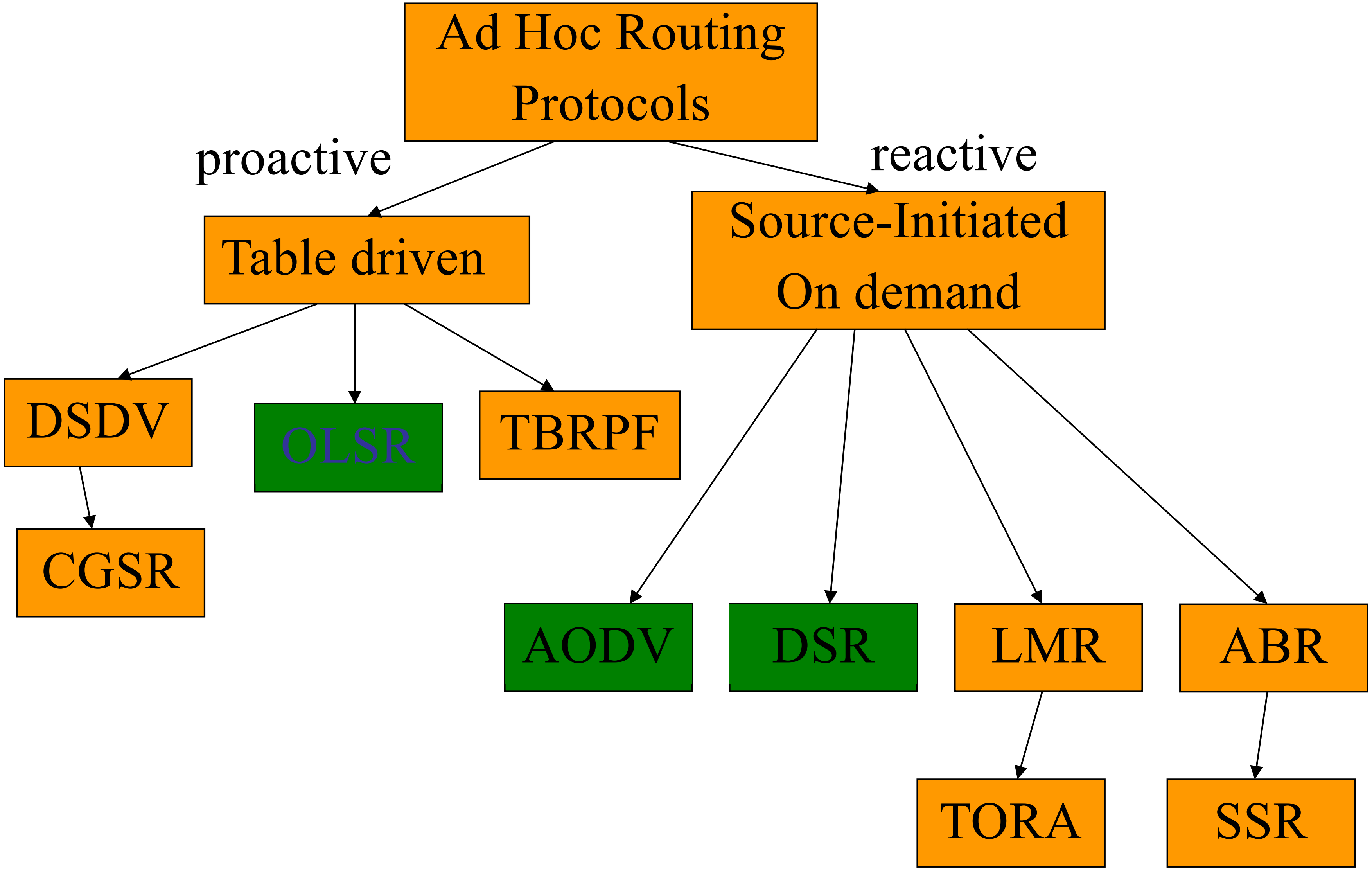
4. QoS-based routing

- Balance between data quality and energy consumption for instance

Protocol Trade-offs

- Proactive protocols
  - Always maintain routes
  - Little or no delay for route determination
  - Consume bandwidth to keep routes up-to-date
  - Maintain routes which may never be used
- Reactive protocols
  - Lower overhead since routes are determined on demand
  - Significant delay in route determination
  - Employ flooding (global search)
  - Control traffic may be bursty
- Which approach achieves a better trade-off depends on the traffic and mobility patterns

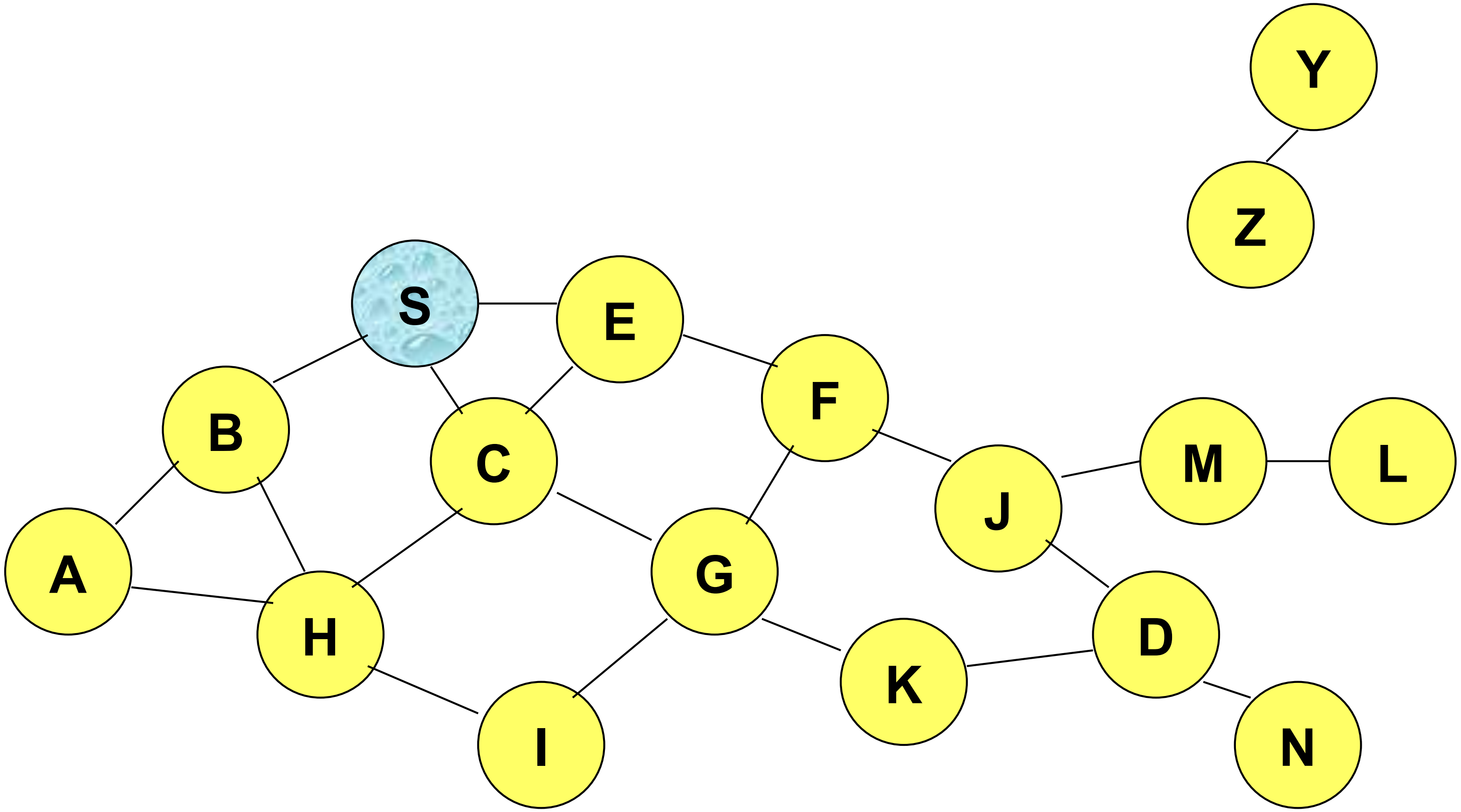
Classification



Dynamic Source Routing (DSR) [Johnson96]

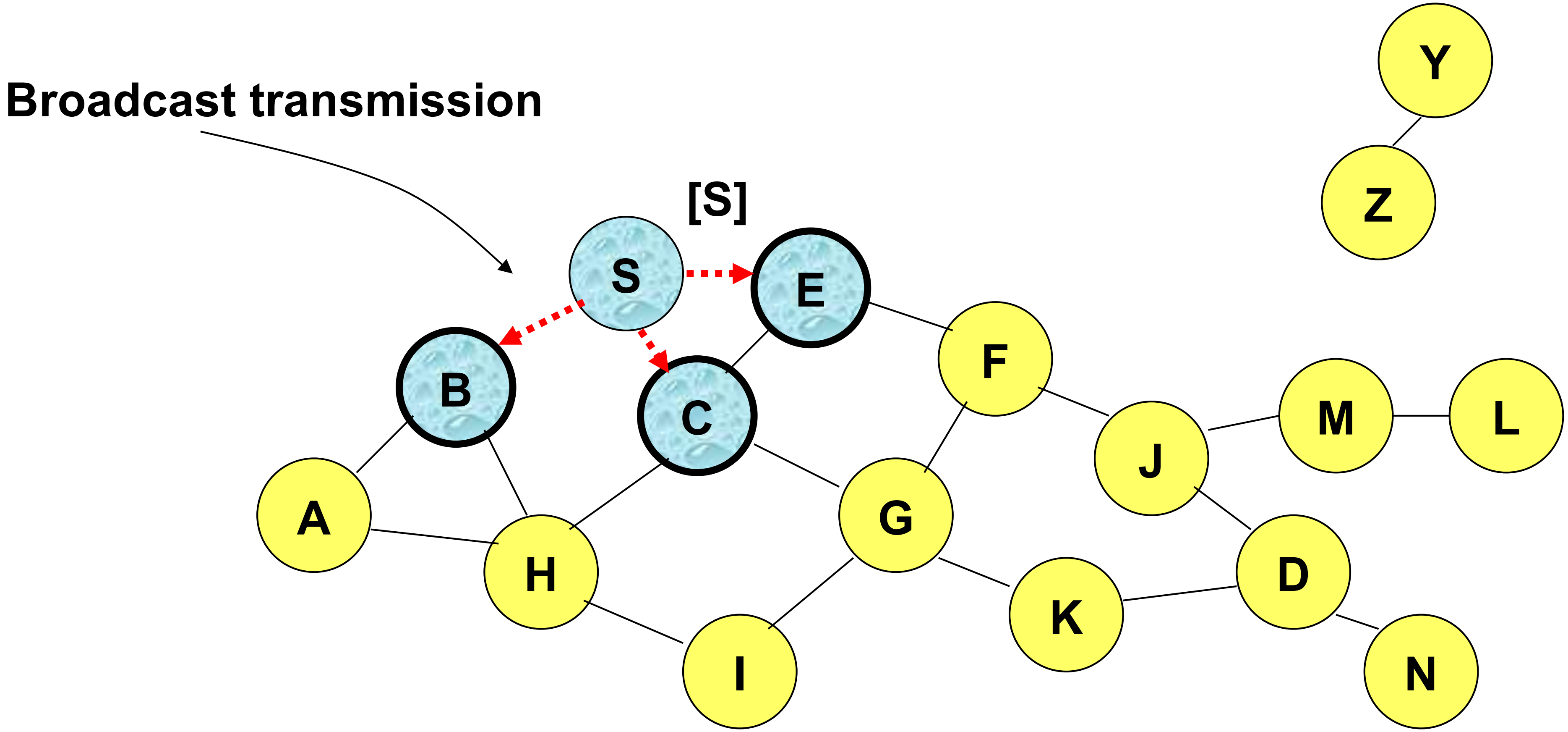
- When node S wants to send a packet to node D, but does not know a route to D, node S initiates a route discovery
- Source node S floods Route Request (RREQ)
- Each node *appends own identifier* when forwarding RREQ

Route Discovery in DSR



Represents a node that has received RREQ for D from S

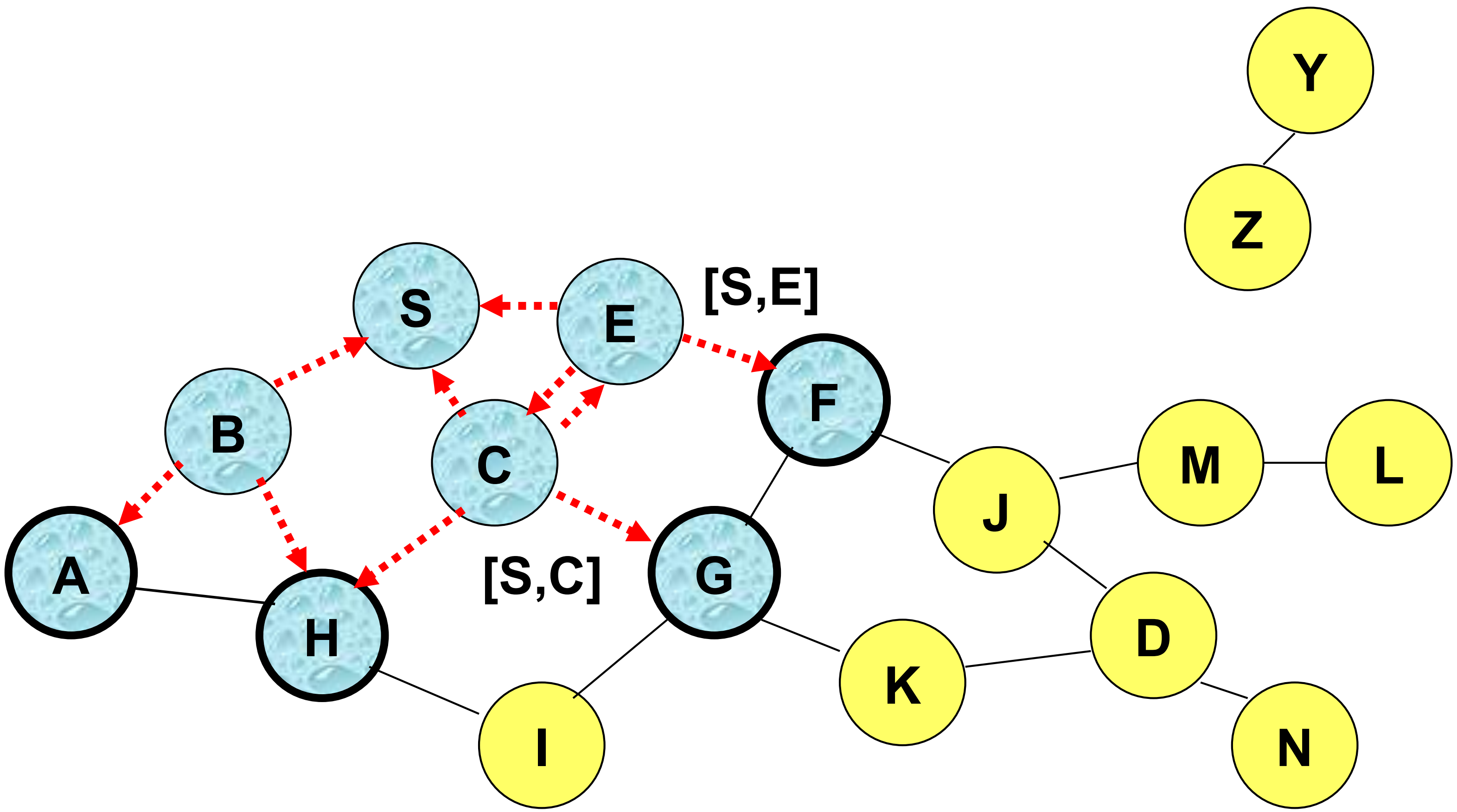
Route Discovery in DSR



Represents transmission of RREQ

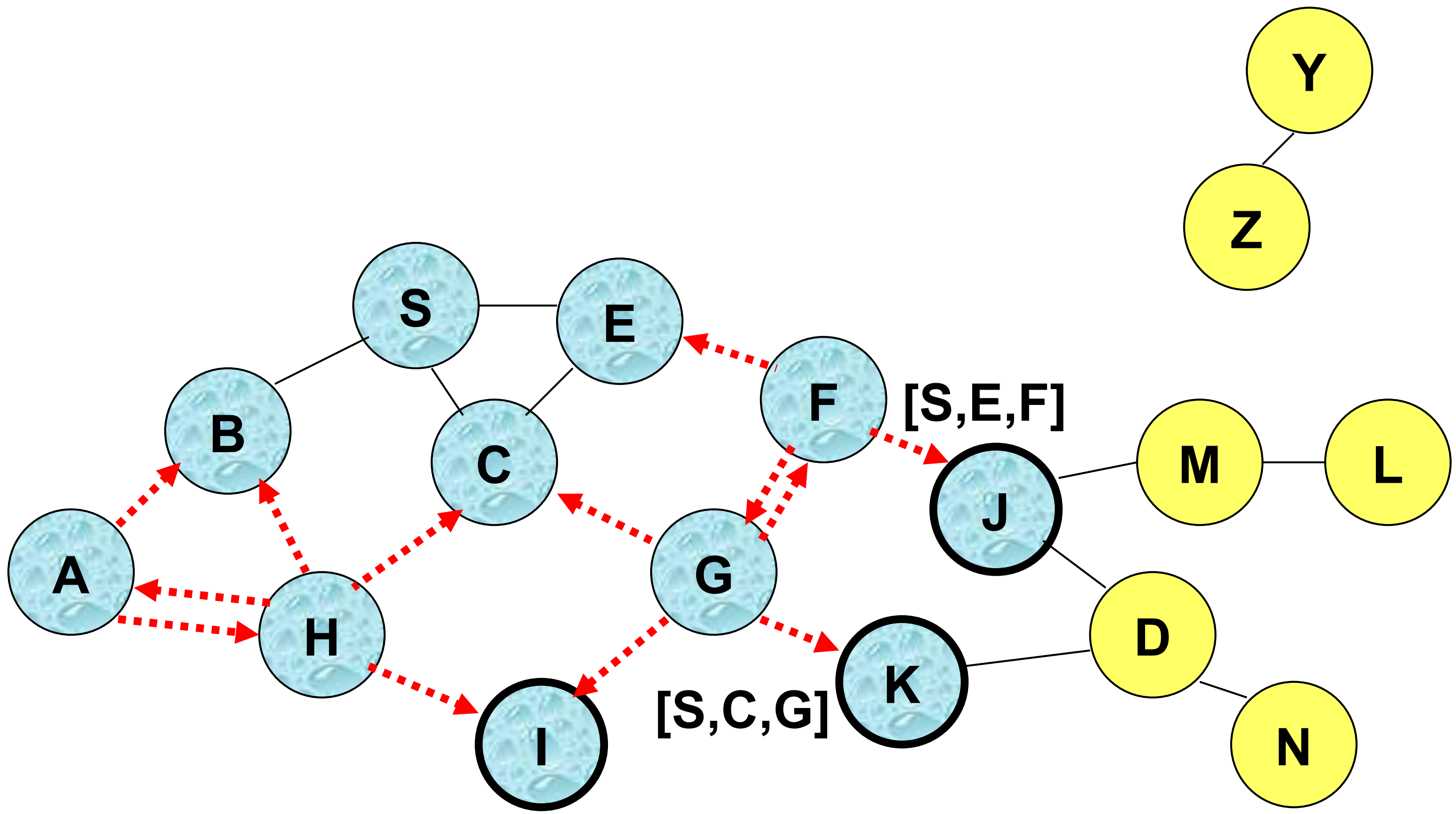
[X,Y] Represents list of identifiers appended to RREQ

Route Discovery in DSR



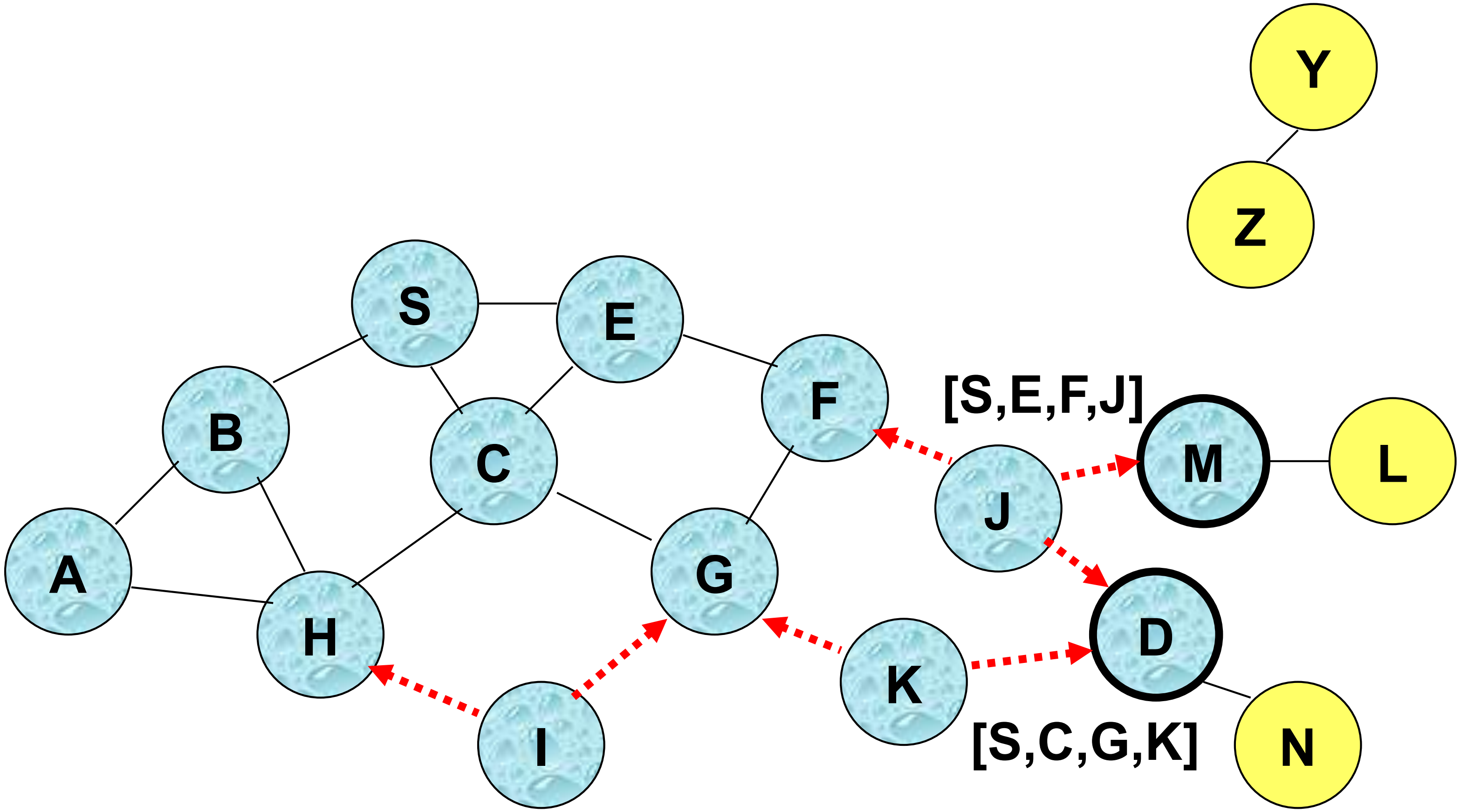
• **Node H receives packet RREQ from two neighbors: potential for collision**

Route Discovery in DSR



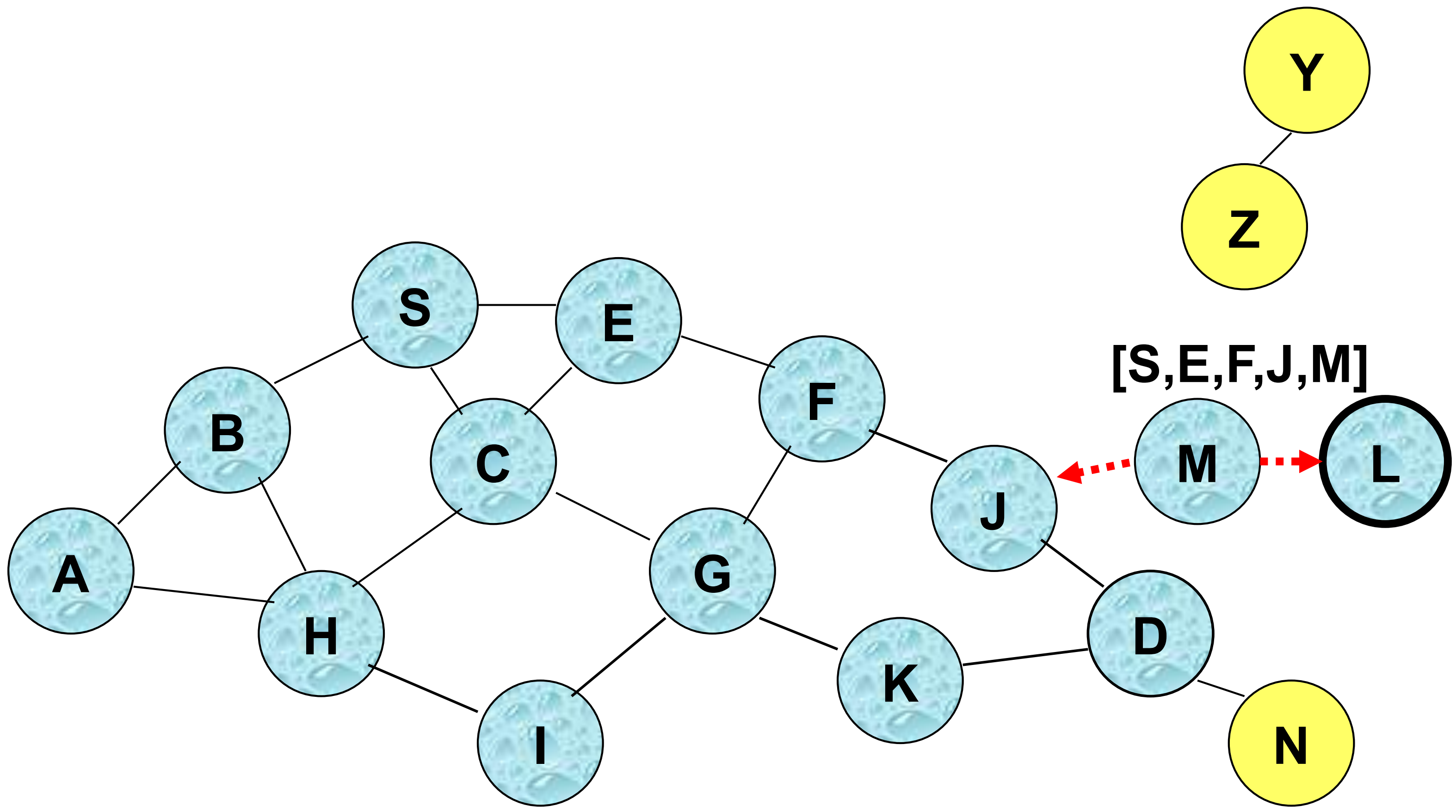
- Node C receives RREQ from G and H, but does not forward it again, because node C has **already forwarded RREQ** once

Route Discovery in DSR



- Nodes J and K both broadcast RREQ to node D
- Since nodes J and K are hidden from each other, their transmissions may collide

Route Discovery in DSR



- Node D **does not forward** RREQ, because node D is the **intended target** of the route discovery

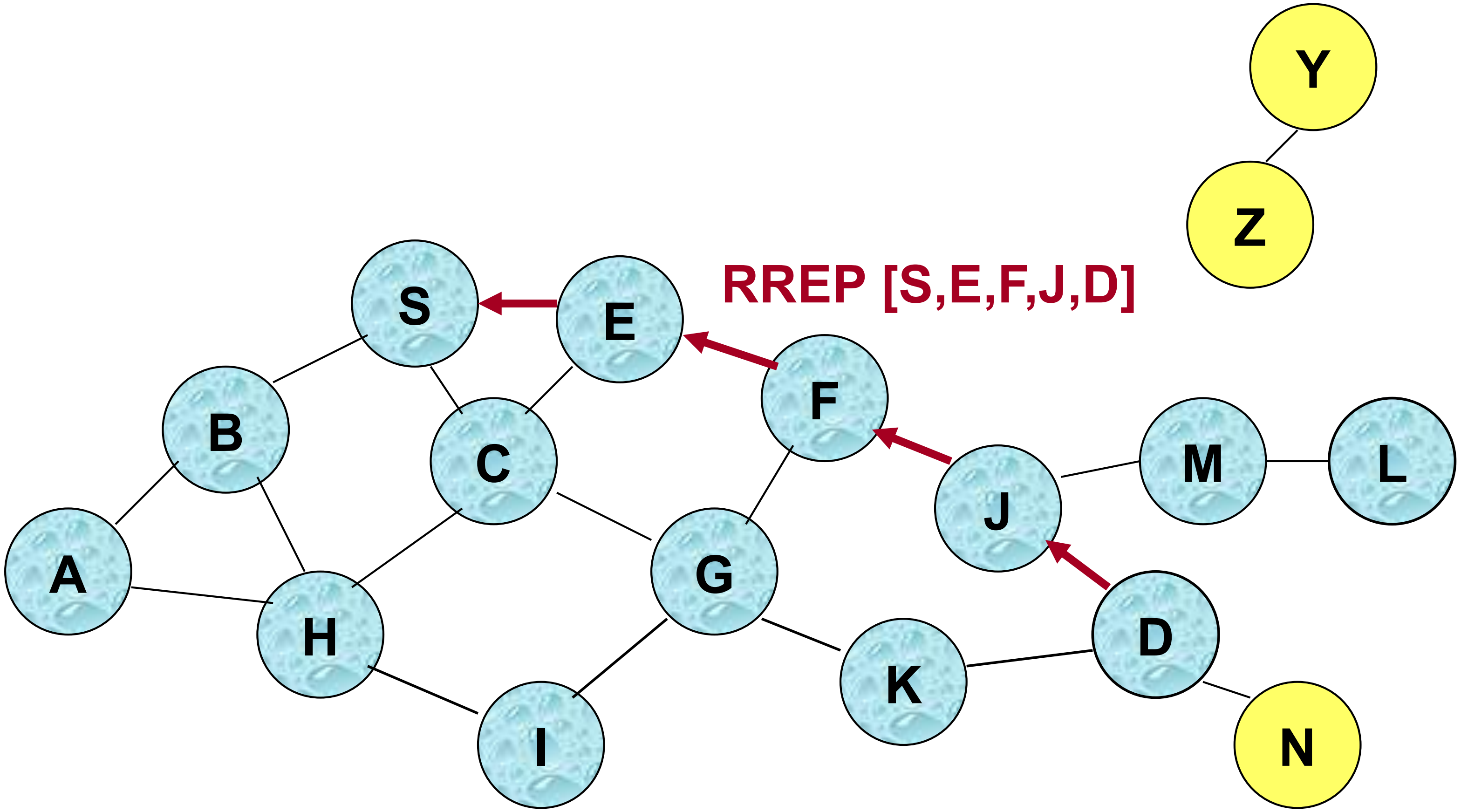
Route Discovery in DSR

- Destination D on receiving the first RREQ, sends a Route Reply (RREP)

- RREP is sent on a route obtained by reversing the route appended to received RREQ

- RREP includes the route from S to D on which RREQ was received by node D

Route Reply in DSR

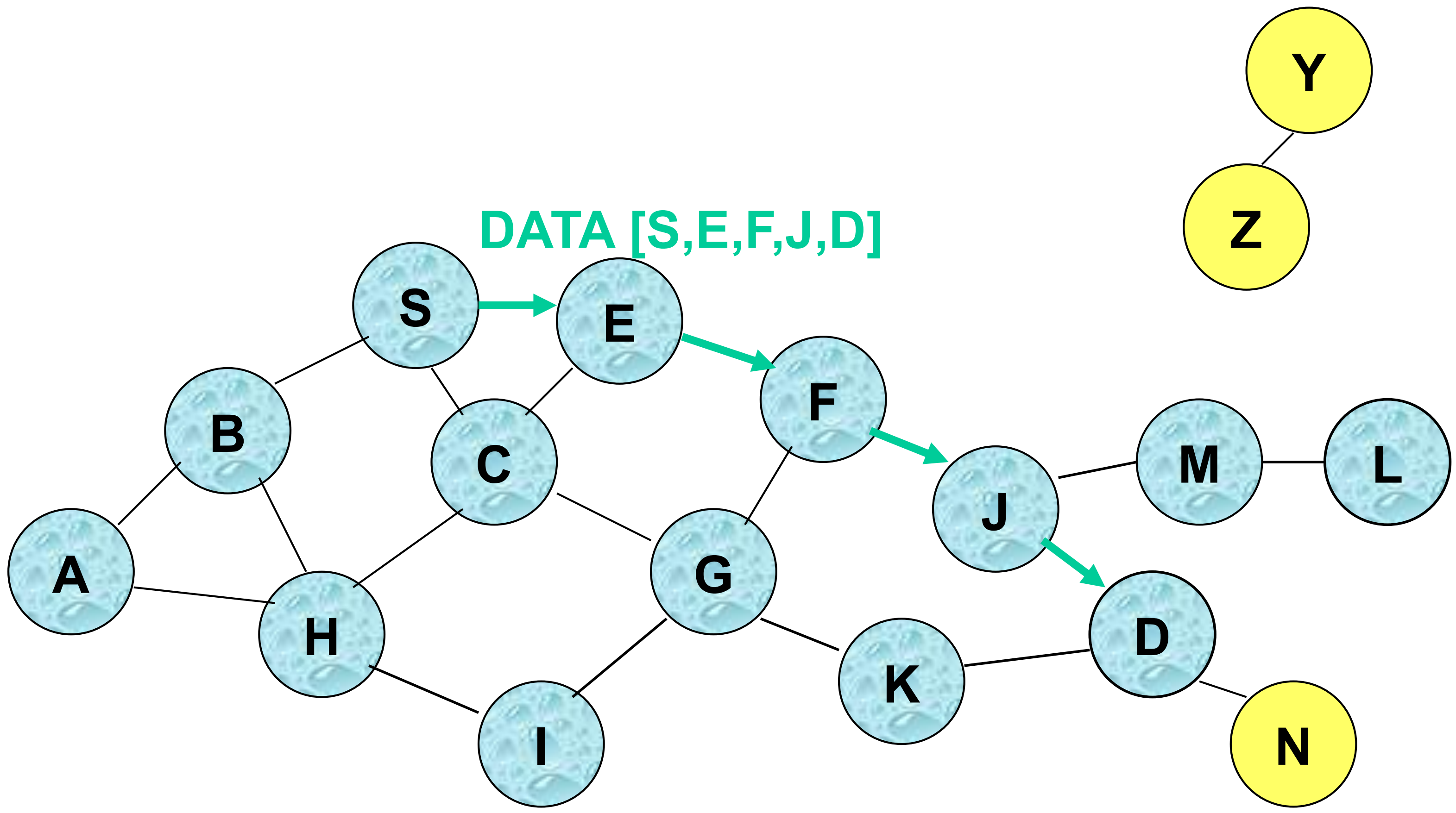


Represents RREP control message

Dynamic Source Routing (DSR)

- Node S on receiving RREP, caches the route included in the RREP
- When node S sends a data packet to D, the entire route is included in the packet header
  - hence the name source routing
- Intermediate nodes use the source route included in a packet to determine to whom a packet should be forwarded

Data Delivery in DSR



Packet header size grows with route length

Dynamic Source Routing: Disadvantages

- Packet header size grows with route length due to source routing
- Flood of route requests may potentially reach all nodes in the network
- Potential collisions between route requests propagated by neighboring nodes
  - insertion of random delays before forwarding RREQ
- Increased contention if too many route replies come back due to nodes replying using their local cache
  - Route Reply *Storm* problem
- Stale caches will lead to increased overhead

Ad Hoc On-Demand Distance Vector Routing (AODV) [Perkins99Wmcsa]

- DSR includes source routes in packet headers
- Resulting large headers can sometimes degrade performance
  - particularly when data contents of a packet are small

- AODV attempts to improve on DSR by maintaining routing tables at the nodes, so that data packets do not have to contain routes

- AODV retains the desirable feature of DSR that routes are maintained only between nodes which need to communicate

AODV

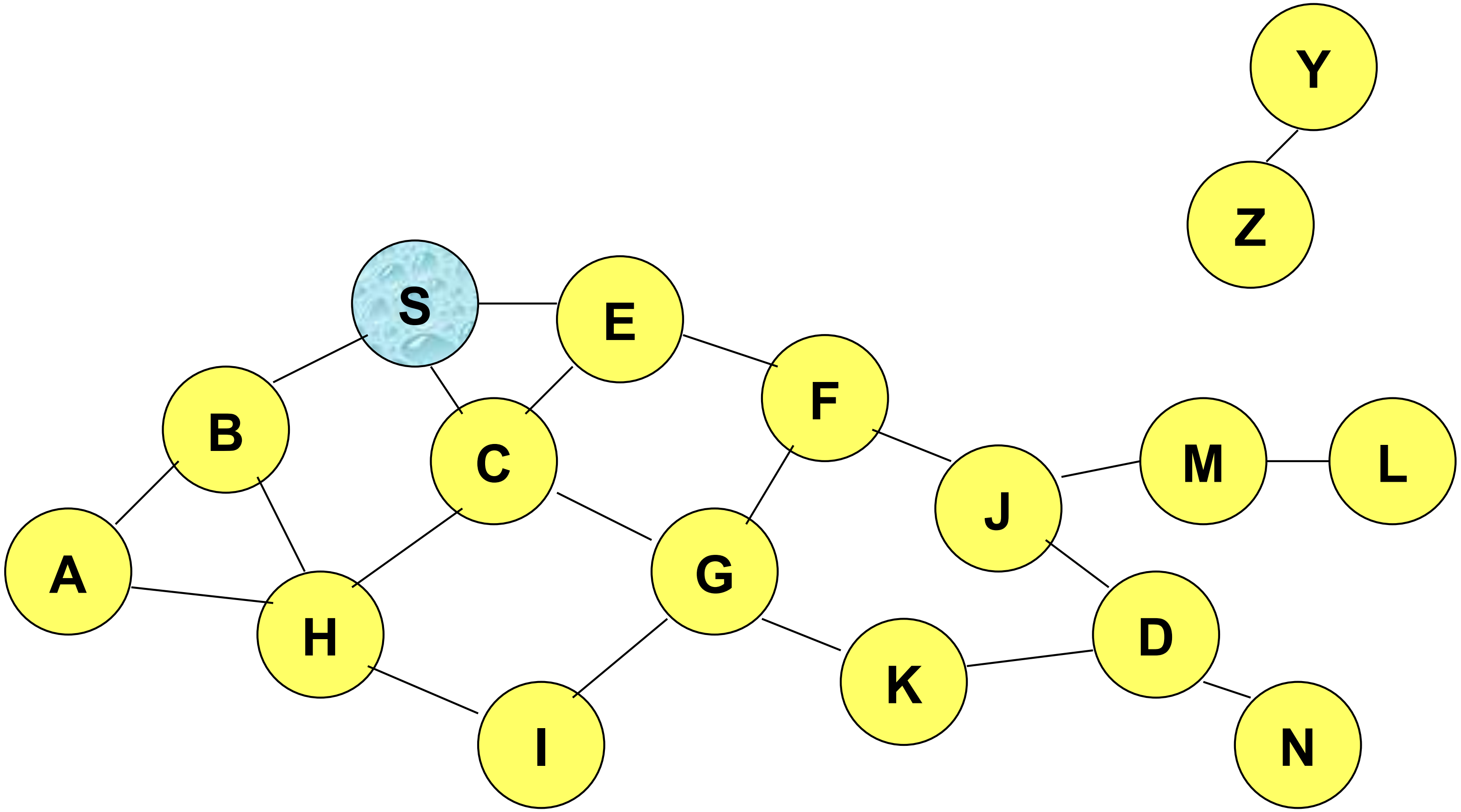
- Route Requests (RREQ) are forwarded in a manner similar to DSR

- When a node re-broadcasts a Route Request, it sets up a reverse path pointing towards the source
  - AODV assumes symmetric (bi-directional) links

- When the intended destination receives a Route Request, it replies by sending a Route Reply (RREP)

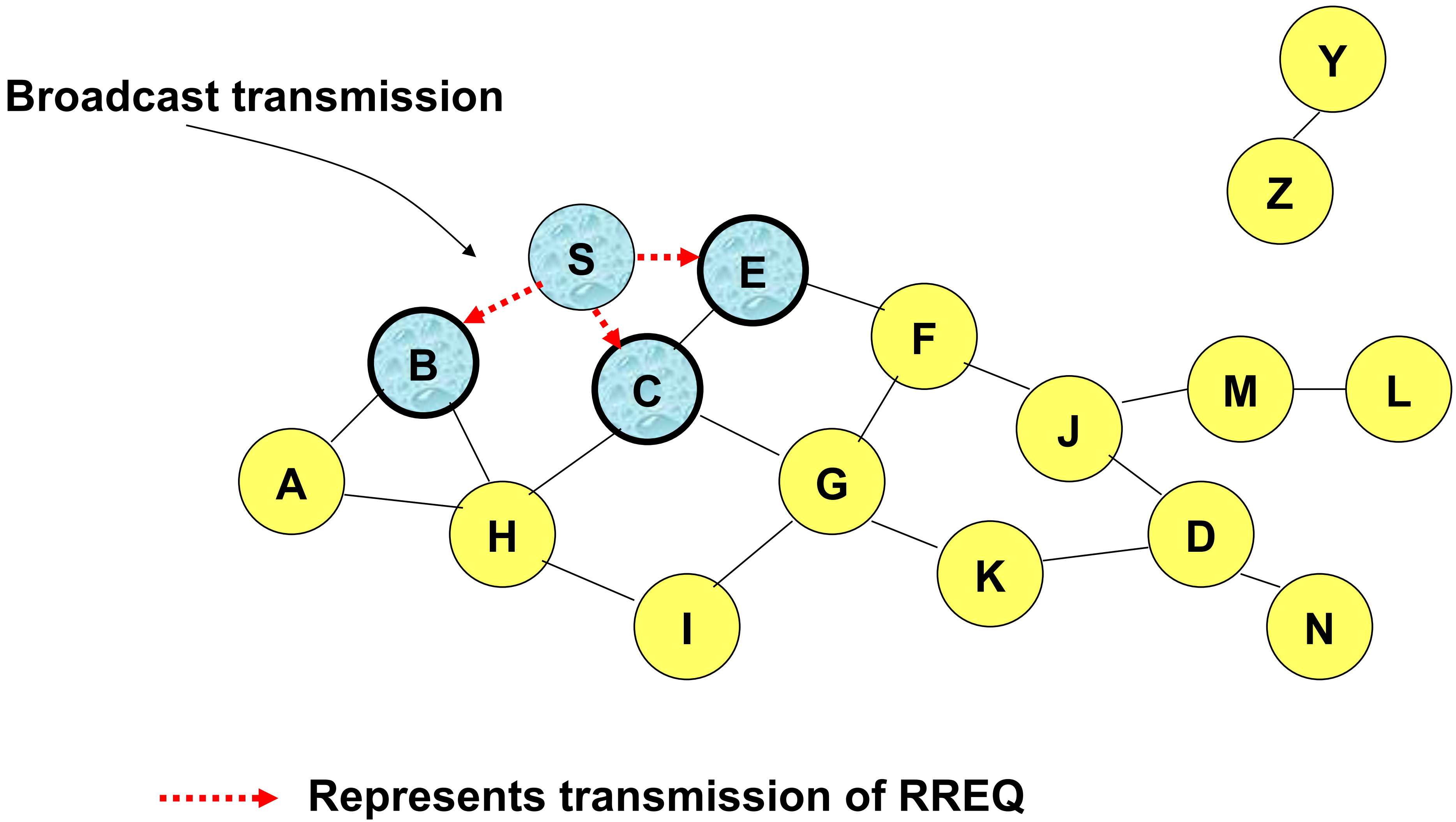
- Route Reply travels along the reverse path set-up when Route Request is forwarded

Route Requests in AODV

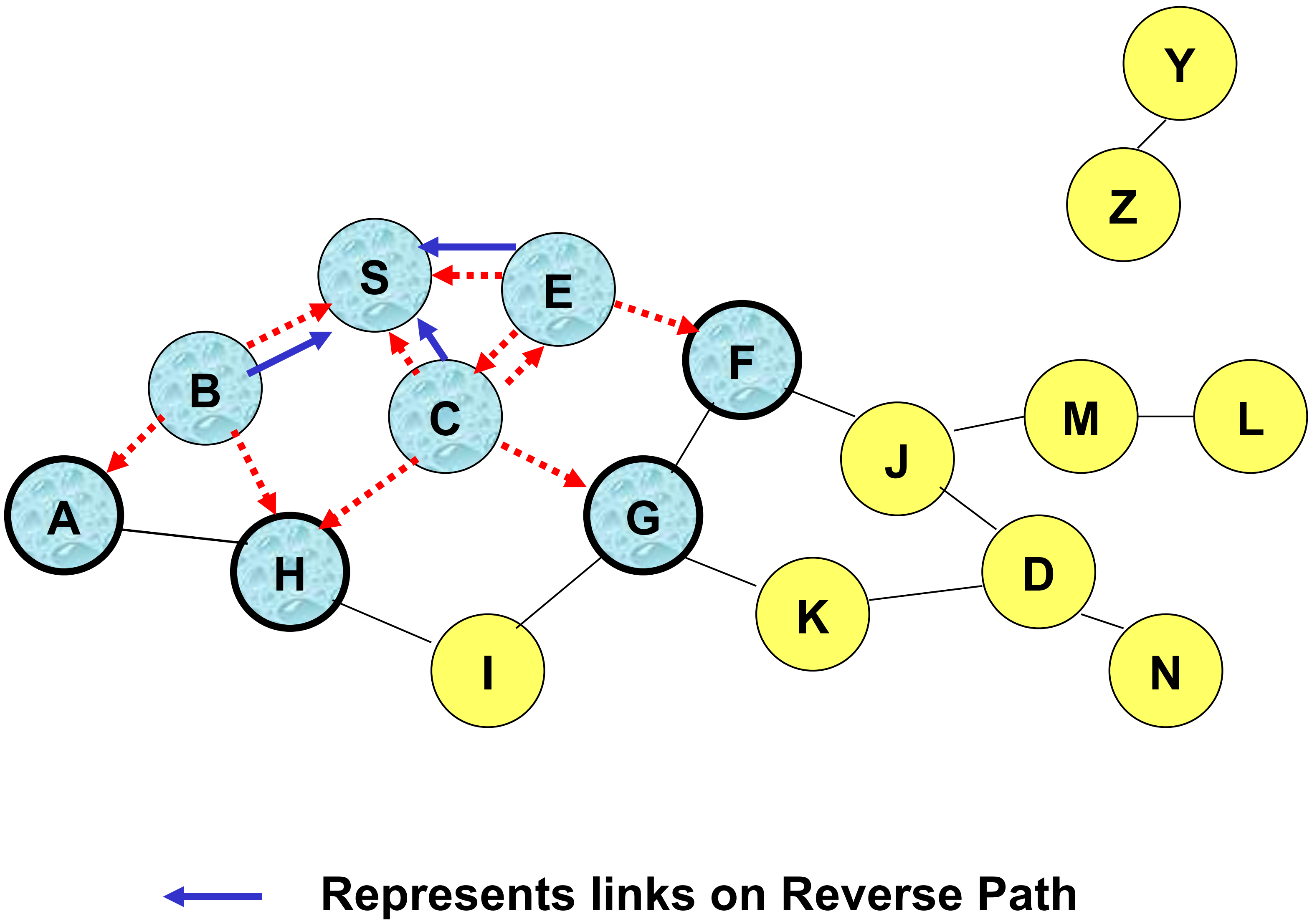


Represents a node that has received RREQ for D from S

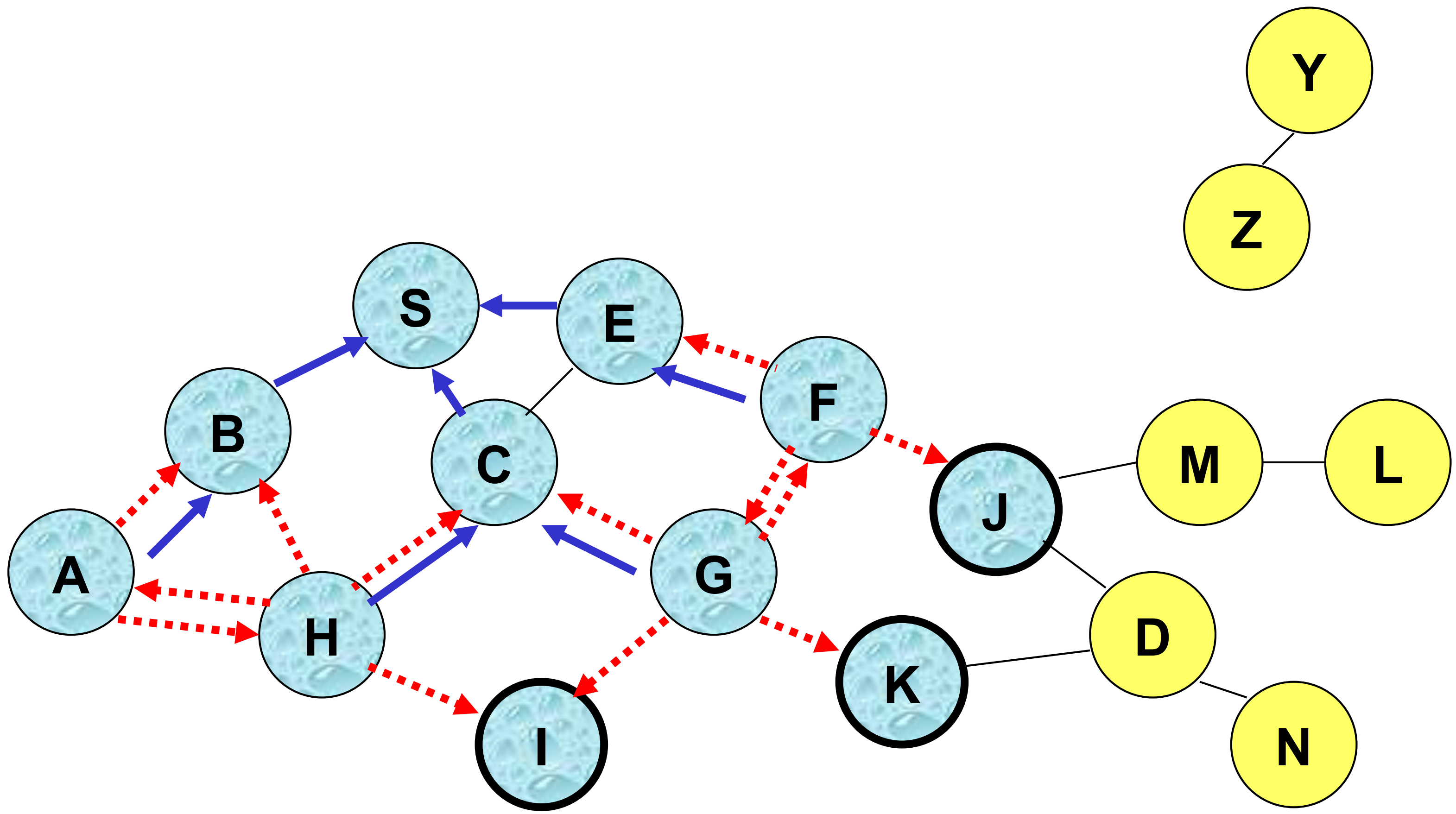
Route Requests in AODV



Route Requests in AODV

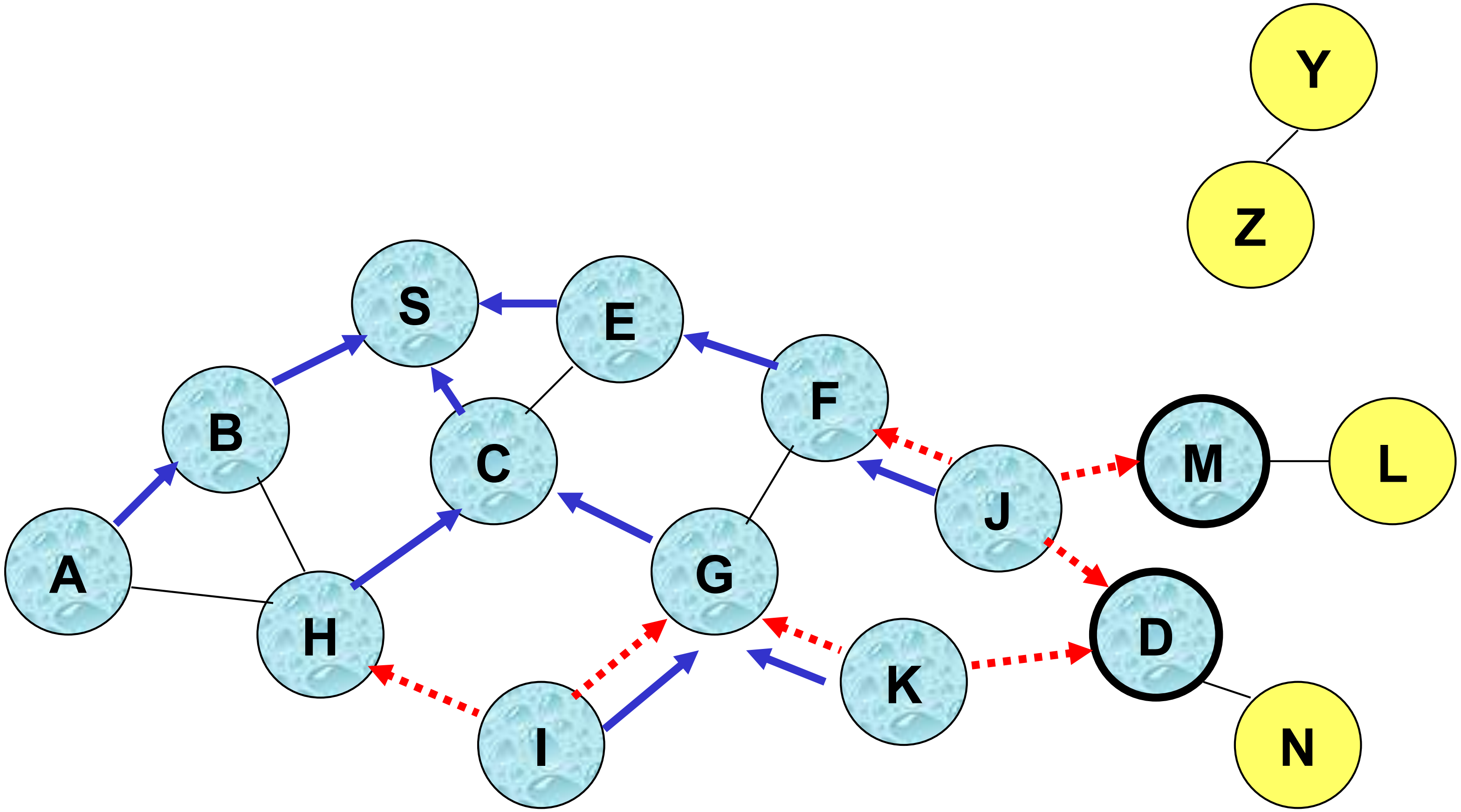


Reverse Path Setup in AODV

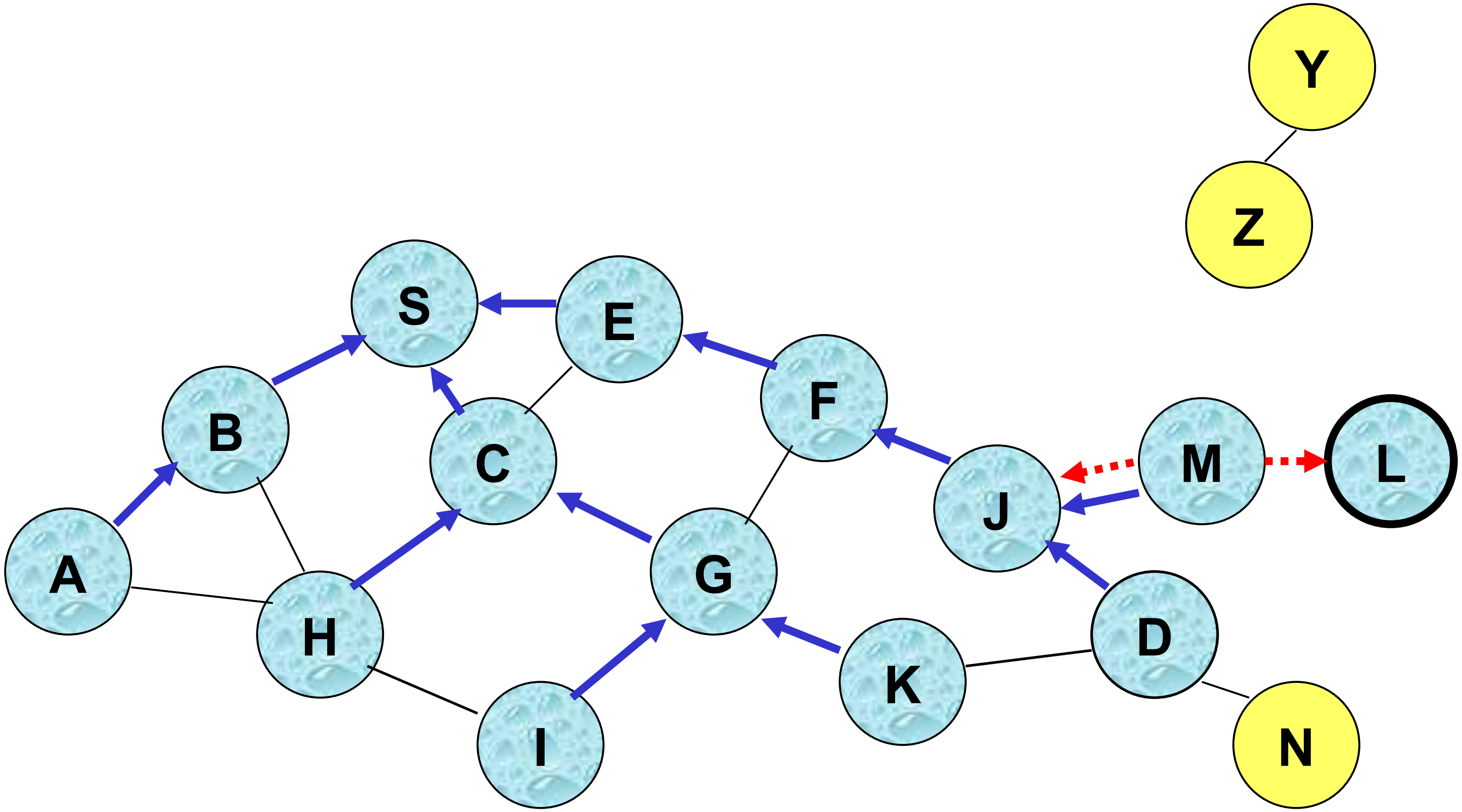


- Node C receives RREQ from G and H, but does not forward it again, because node C has **already forwarded RREQ** once

Reverse Path Setup in AODV

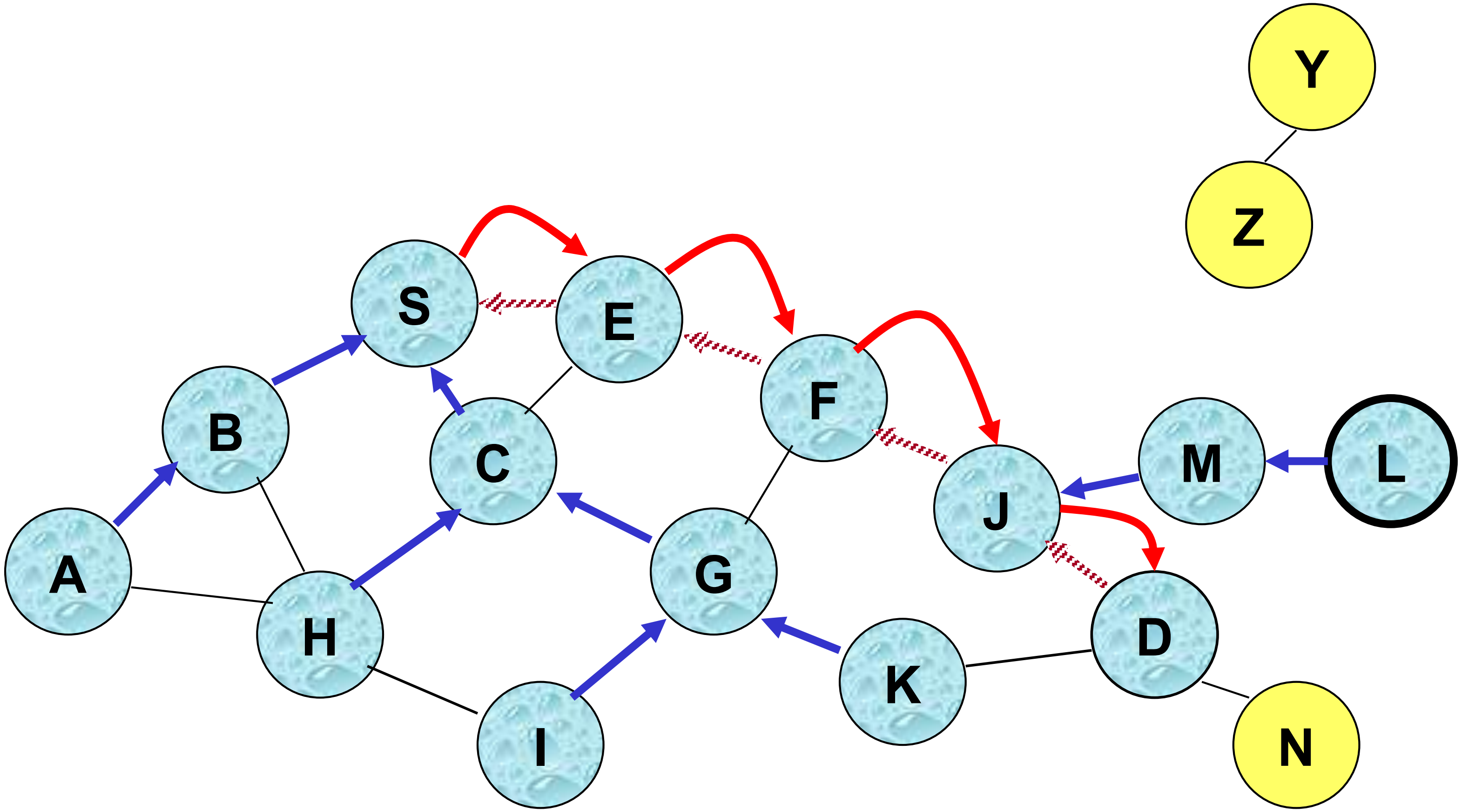


Reverse Path Setup in AODV

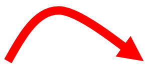


- Node D **does not forward** RREQ, because node D is the **intended target** of the RREQ

Forward Path Setup in AODV



Forward links are setup when RREP travels along the reverse path

 **Represents a link on the forward path**

Route Request and Route Reply

- Route Request (RREQ) includes the last known sequence number for the destination

- An intermediate node may also send a Route Reply (RREP) provided that it knows a more recent path than the one previously known to sender
- Intermediate nodes that forward the RREP, also record the next hop to destination

- A routing table entry maintaining a reverse path is purged after a timeout interval
- A routing table entry maintaining a forward path is purged if *not used* for a *active_route_timeout* interval

Link Failure

- A neighbor of node X is considered active for a routing table entry if the neighbor sent a packet within *active_route_timeout* interval which was forwarded using that entry

- Neighboring nodes periodically exchange hello message

- When the next hop link in a routing table entry breaks, all active neighbors are informed

- Link failures are propagated by means of Route Error (RERR) messages, which also update destination sequence numbers

Route Error

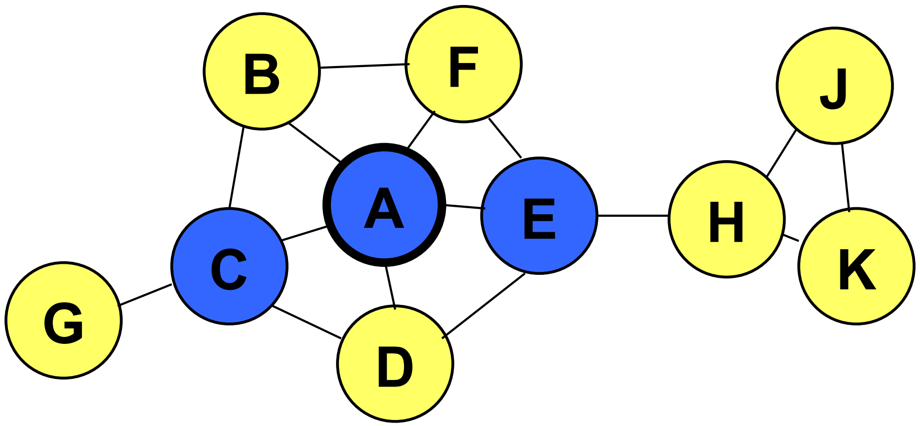
- When node X is unable to forward packet P (from node S to node D) on link (X,Y), it generates a RERR message
- Node X increments the destination sequence number for D cached at node X
- The incremented sequence number *N* is included in the RERR
- When node S receives the RERR, it initiates a new route discovery for D using destination sequence number at least as large as *N*
- When node D receives the route request with destination sequence number N, node D will set its sequence number to N, unless it is already larger than N

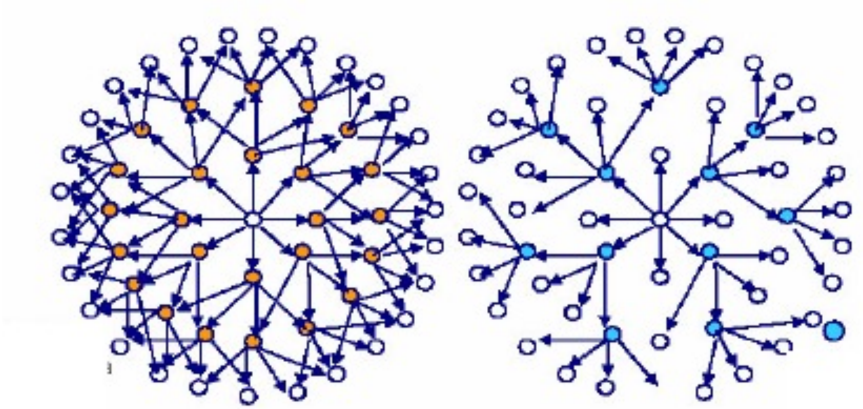
AODV: Summary

- Routes need not be included in packet headers
- Nodes maintain routing tables containing entries only for routes that are in active use
- At most one next-hop per destination maintained at each node
  - DSR may maintain several routes for a single destination
- Sequence numbers are used to avoid old/broken routes
- Sequence numbers prevent formation of routing loops
- Unused routes expire even if topology does not change

Optimized Link State Routing Protocol

- Proactive & Table-driven
- Link State Routing
  - Each node expands a spanning tree
  - Each node can obtain the whole network topology
- Utilizes a technique to reduce message flooding
  - MultiPoint Relaying (MPR)
  - MPR are nodes N at 1-hop of A such that 2-hop neighbors of A are 1-hop neighbors of N





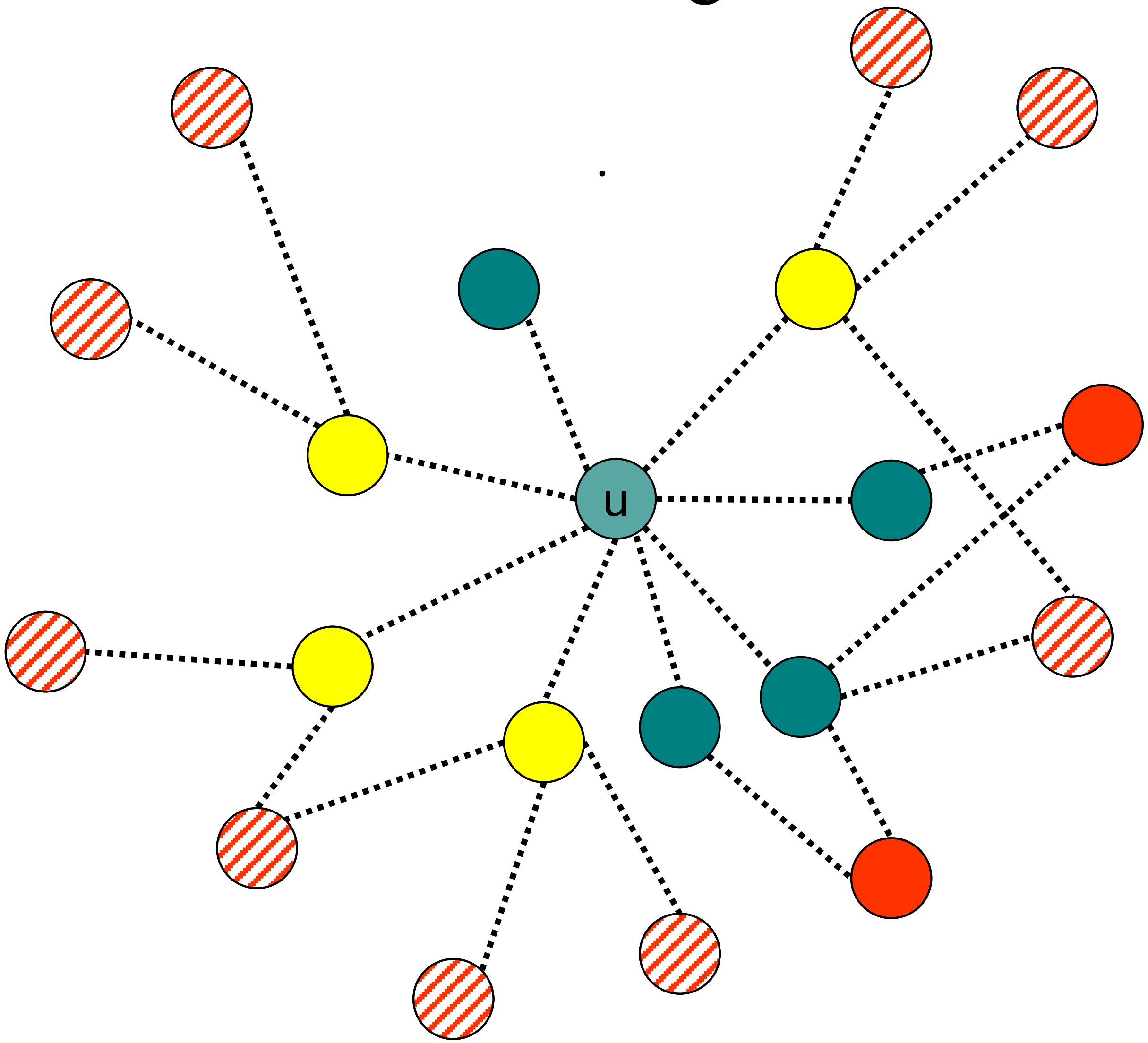
MPR selection algorithm

- Each point u has to select its set of MPR.
- Goal :

  Select in the 1-neighborhood of u ($N_1(u)$) a set of nodes as small as possible which covers the whole 2-neighborhood of u($N_2(u)$).

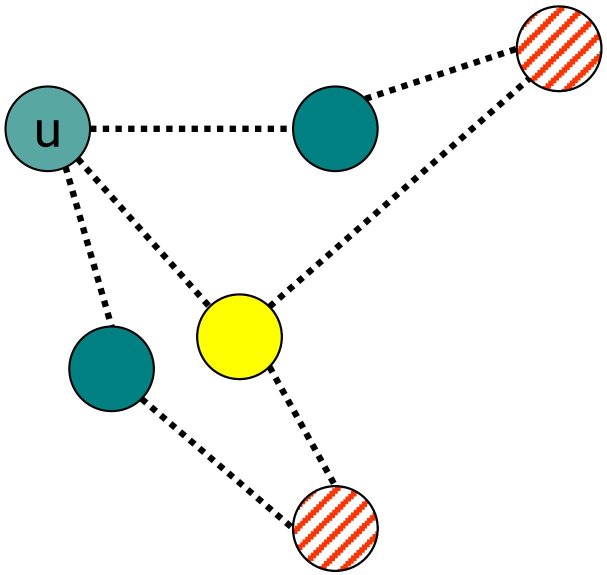
  - Step 1: Select nodes of $N_1(u)$ which cover isolated points of $N_2(u)$.
  - Step 2: Select among the nodes of $N_1(u)$ not selected at the first step, the node which covers the highest number of points of $N_2(u)$ and go on till every points of $N_2(u)$ are covered.

MPR selection algorithm



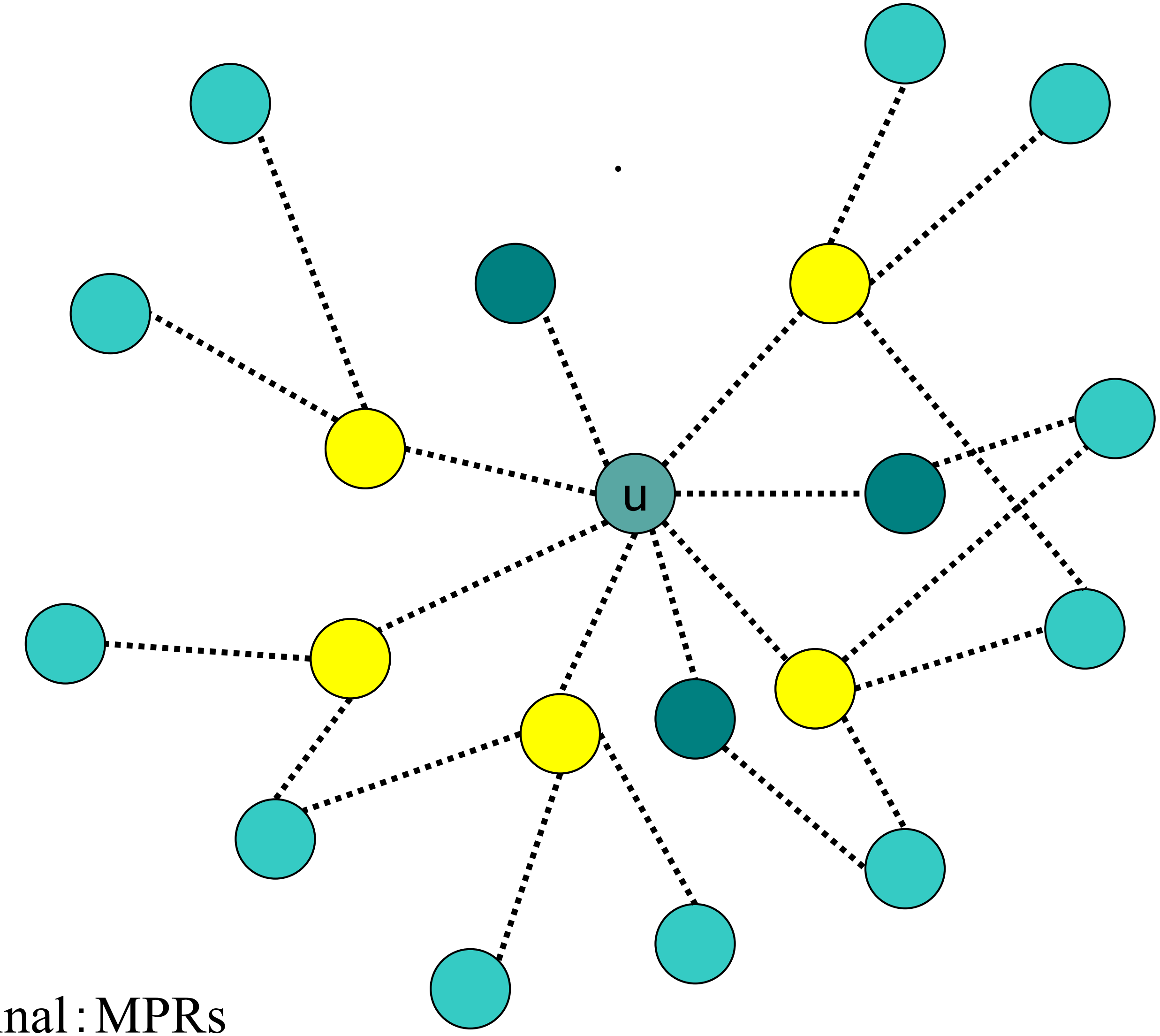
- **First step:** Select nodes in N1(u) which cover "isolated points" of N2(u).

MPR selection algorithm



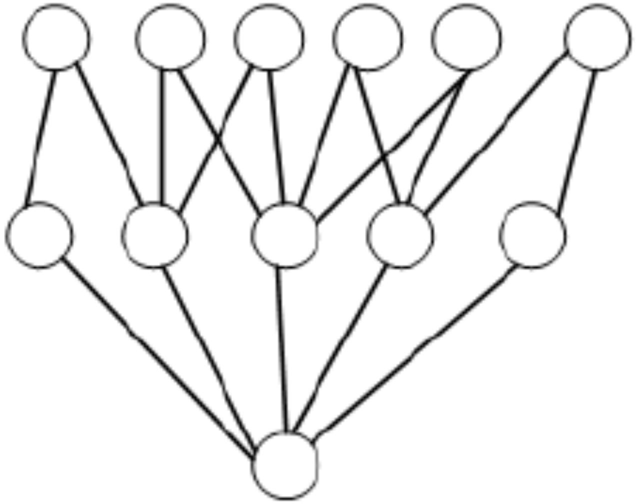
- **Second step :** Consider in N1(u) only points which are not already selected at the first step NPR1(u) and points in N2(u) which are not covered by the NPR1(u) . While there exists points in N2(u) not covered by the selected MPR, select in N2(u), the node which covers the highest number of non-covered nodes in N2(u).

MPR selection algorithm

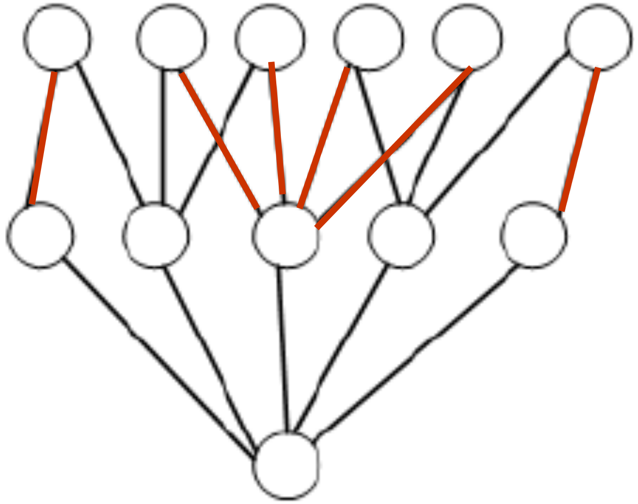


- Final : MPRs

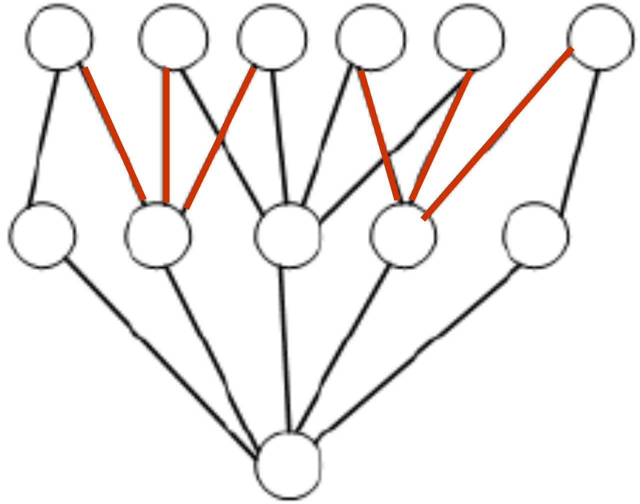
Can find non-optimal solution



topology



Solution found

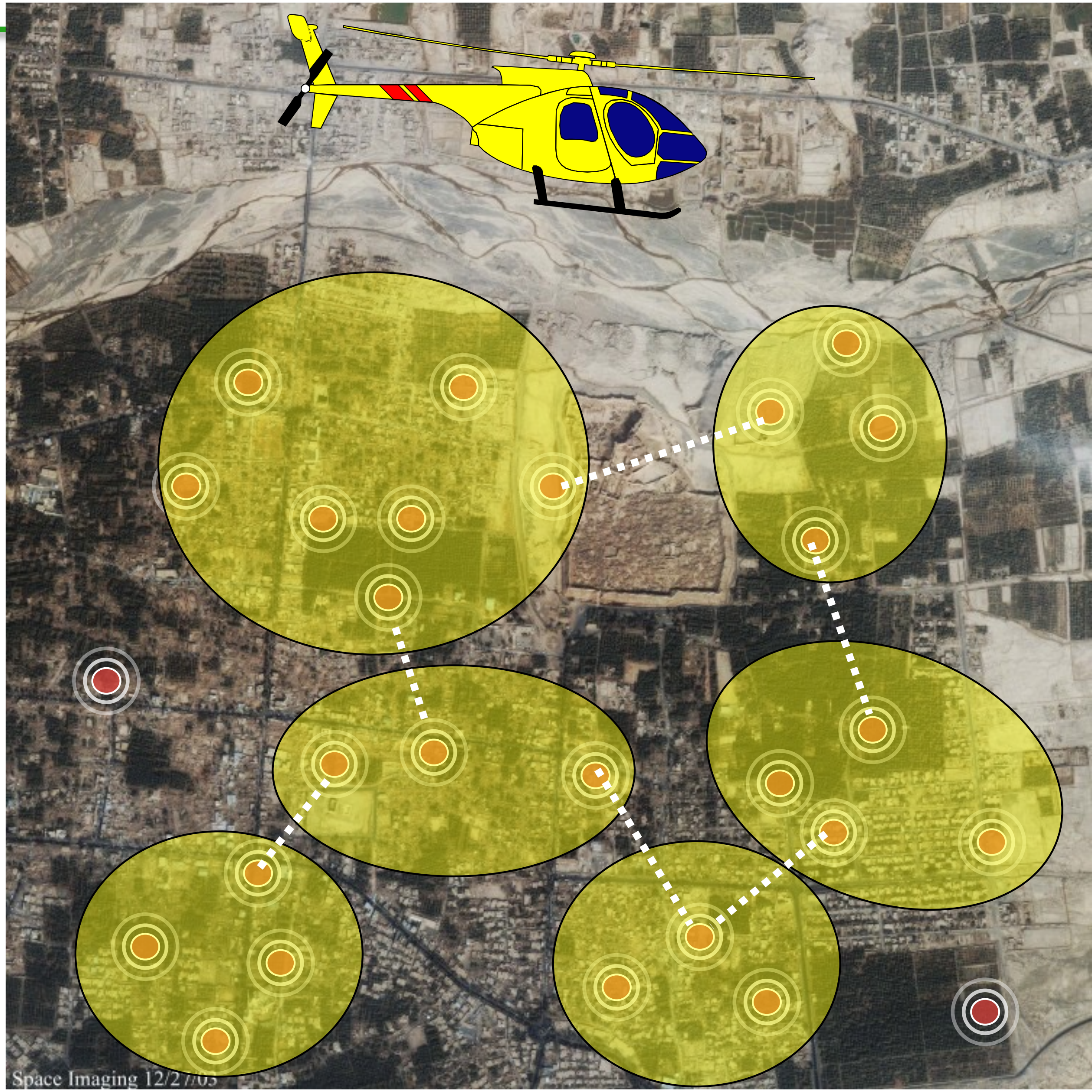


Best solution

Step 1: Select nodes of $N_1(u)$ which cover isolated points of $N_2(u)$.

Step 2: Select among the nodes of $N_1(u)$ not selected at the first step, the node which covers the highest number of points of $N_2(u)$ and go on till every points of $N_2(u)$ are covered.

Wireless Sensor Network
report to fixed sink

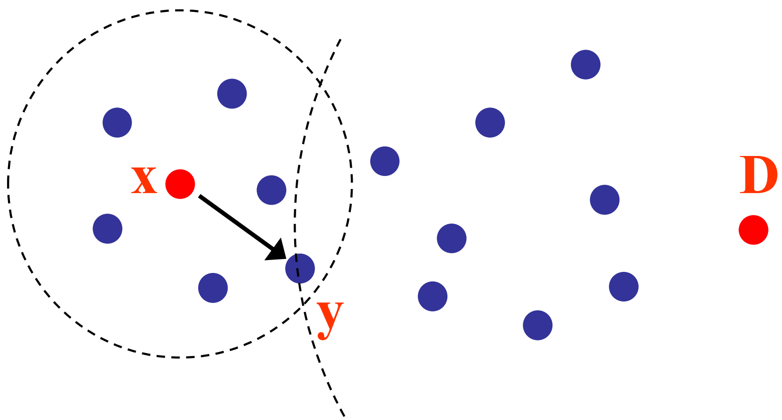


Routing challenges and design issues

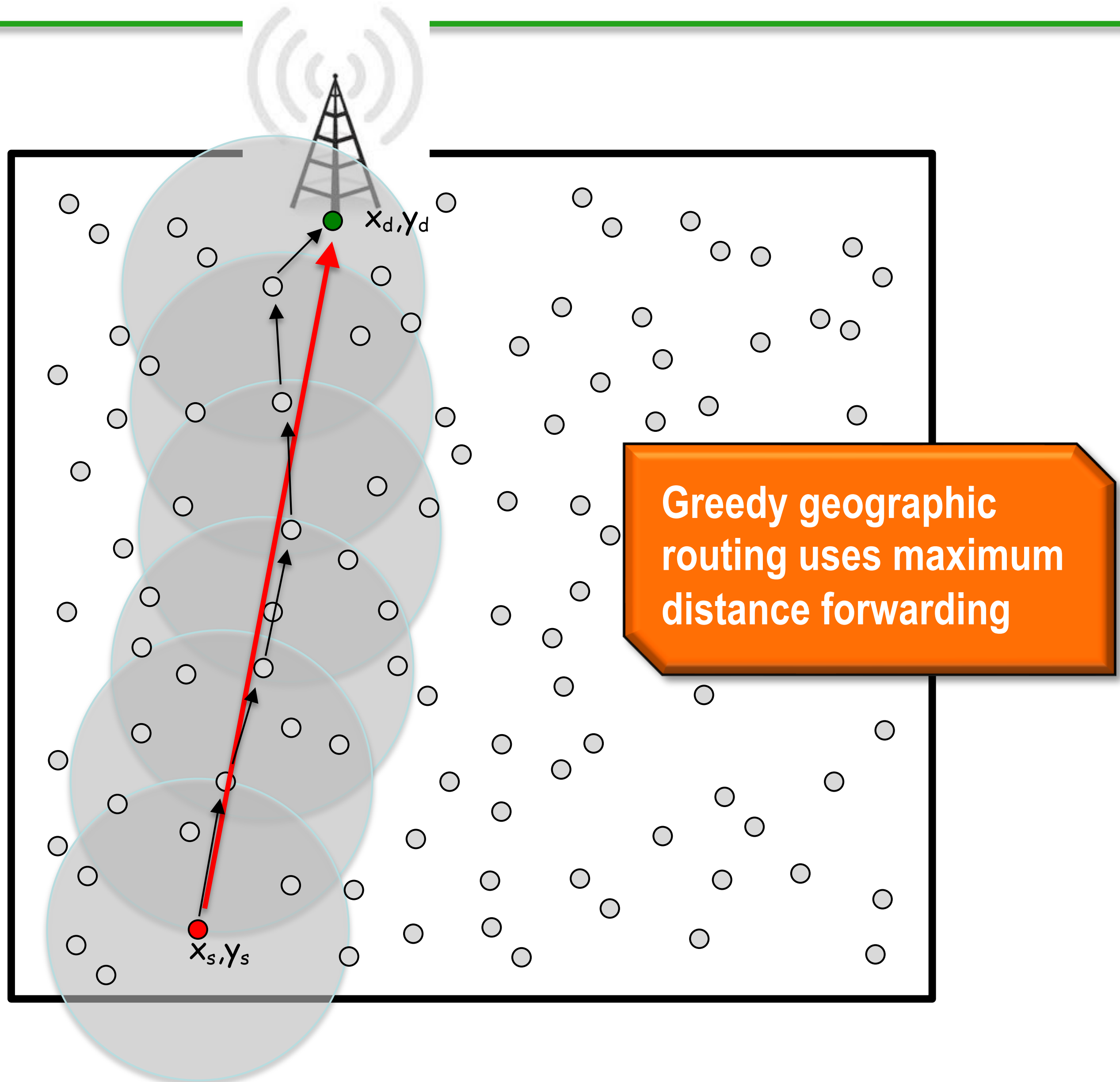
- Node deployment
  - Manual deployment
    - Sensors are manually deployed
    - Data is routed through predetermined path
  - Random deployment
    - Optimal clustering is necessary to allow connectivity & energy-efficiency
    - Multi-hop routing
- Coverage
  - An individual sensor's view is limited
  - Area coverage is an important design factor

Geographic routing

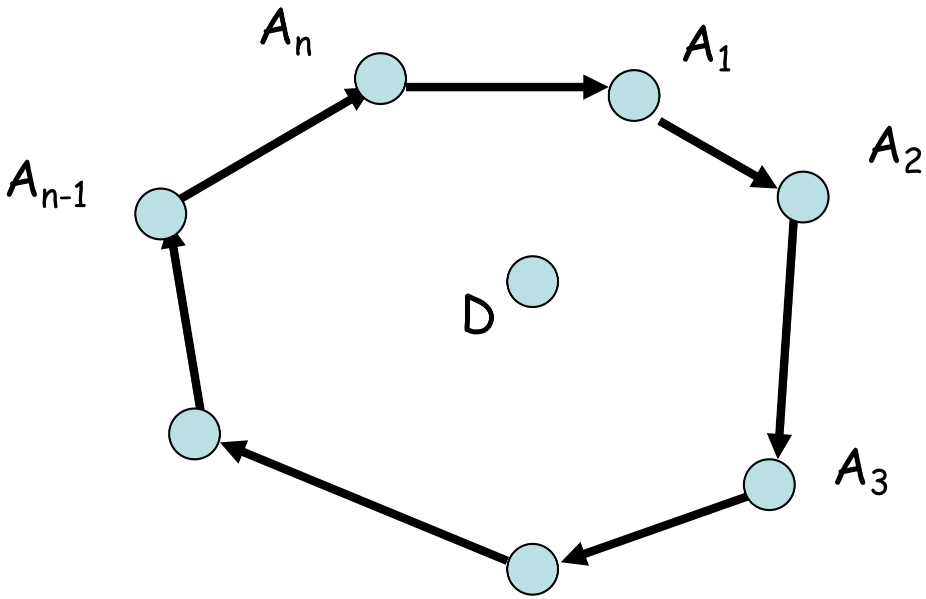
- A node knows its own location, the locations of its neighbors, and the destination's location (*D*)
- The destination's location is included in the packet header
- Forwarding decision is based on local distance information
- *Greedy Forwarding*: achieve max progress towards *D*



Greedy forwarding



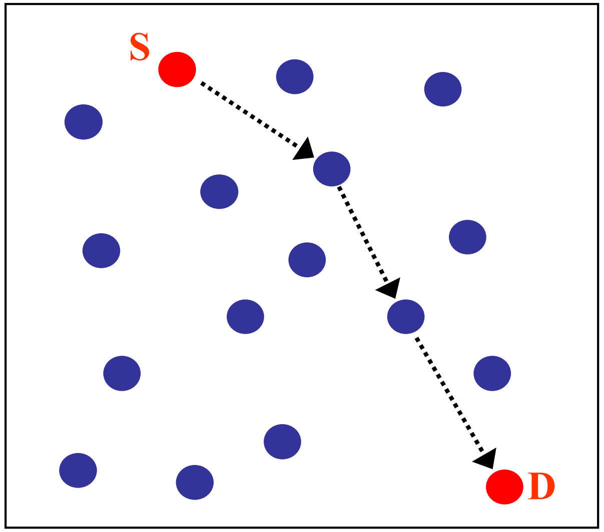
Greedy is loop-free



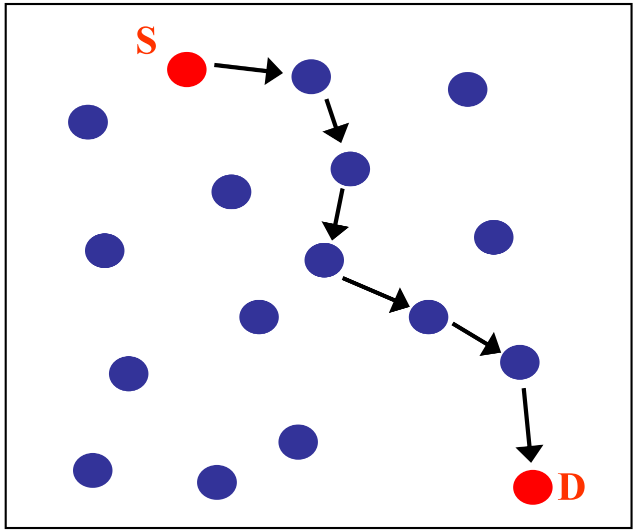
Assume A_1 closest to D

A_2 sends to A_3 – contradiction, A_1 is closer

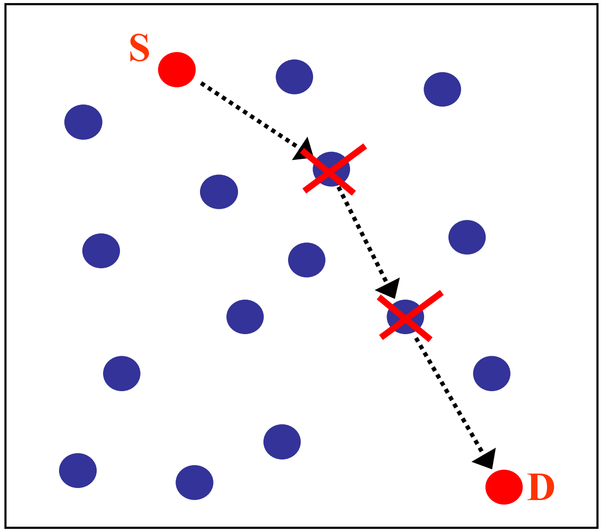
Is maximum distance always good?



Few long links with low quality

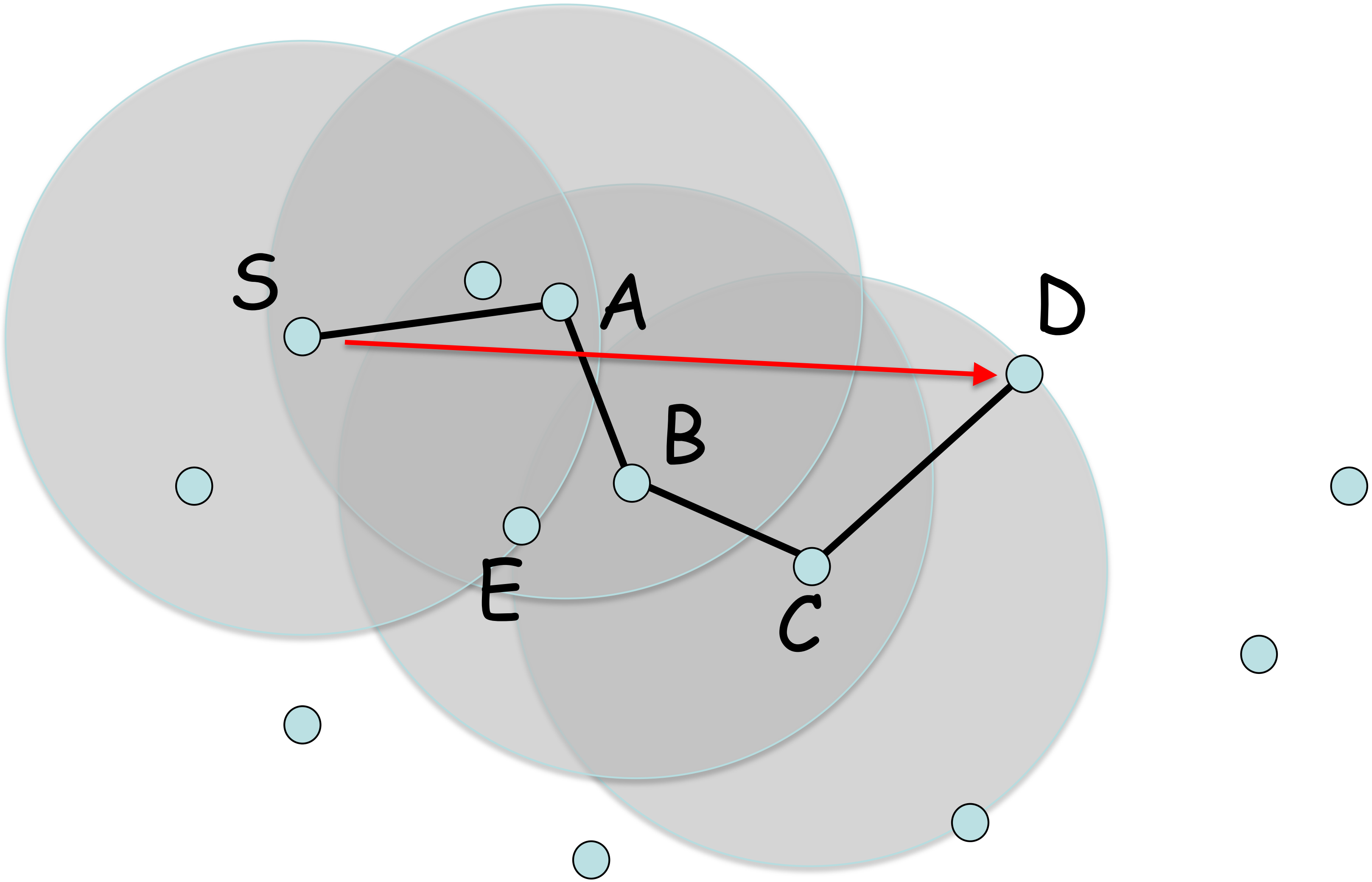


Many short links with high quality

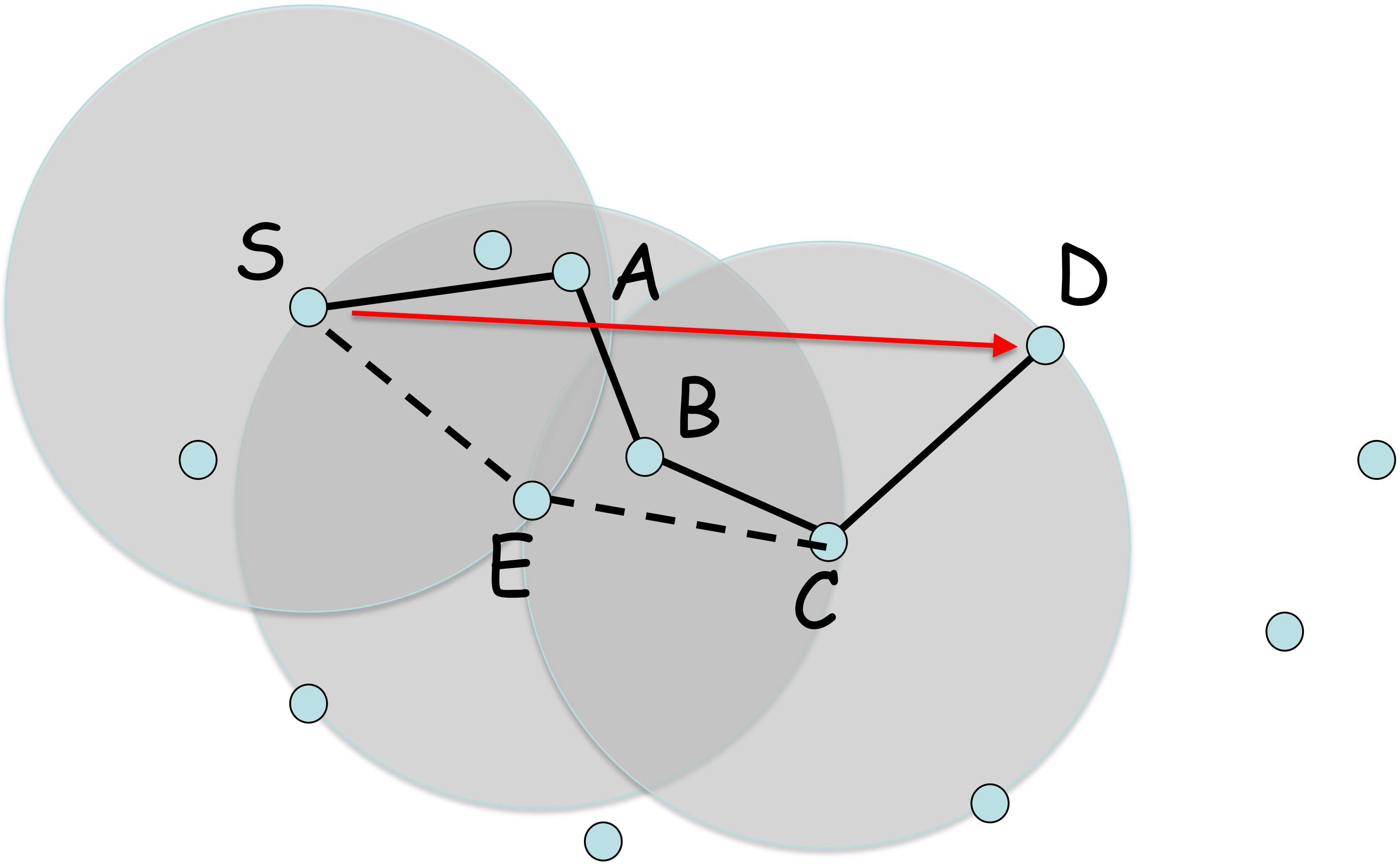


Intermediate nodes that are more sollicited die first

Greedy=shortest path?

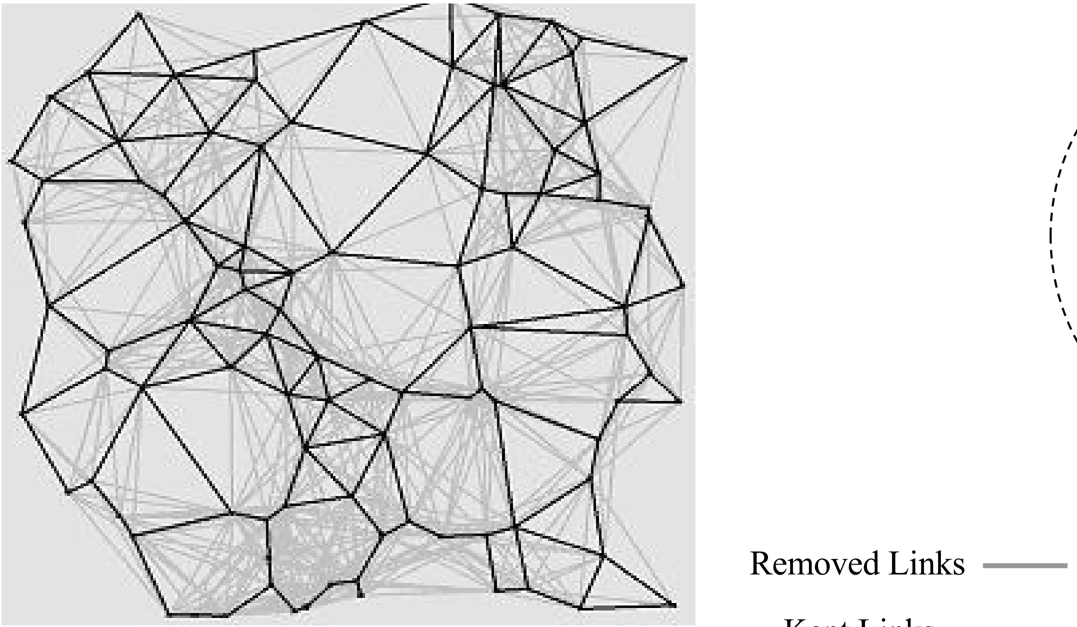


Greedy=shortest path?

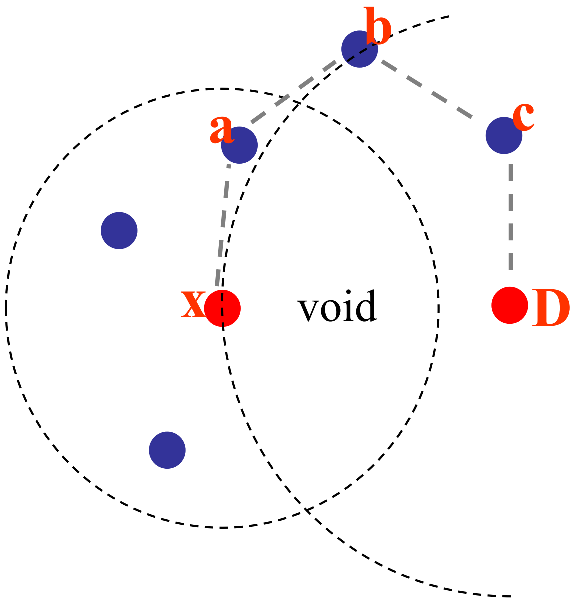


Avoiding holes

- (II) Dead-end Resolution (Local Minima)
  - Getting around voids using *face routing* in planar graphs
  - Need a *planarization* algorithm



Planarized Wireless Network



*Face Routing**

* P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks". *DialM Workshop,* 99.

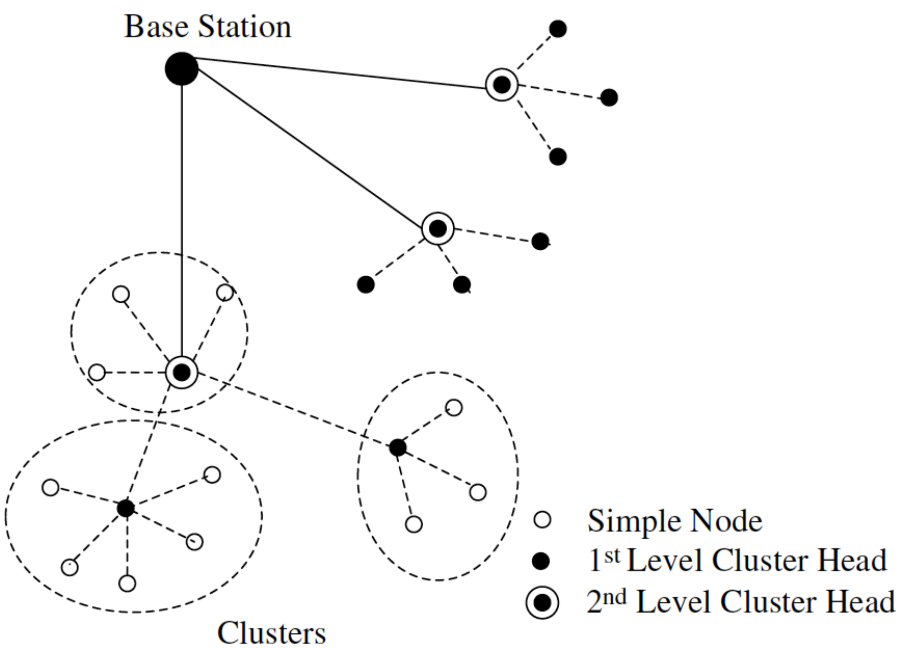
* **GPSR**: Karp, B. and Kung, H.T., Greedy Perimeter Stateless Routing for Wireless Networks, *ACM MobiCom,* , pp. 243-254, August, 2000.

Organizing the network

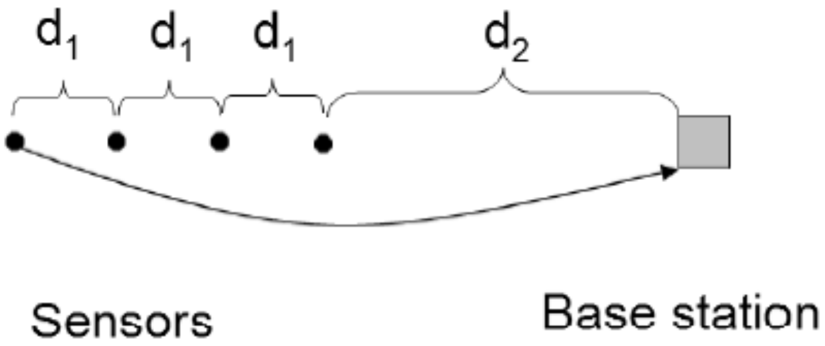
- The network is no longer useful when node's battery dies
- Organizing the network allows for spacing out the lifespan of the nodes
- Hierarchical routing protocols give priority to energy
- Low-Energy Adaptive Clustering Hierarchy (LEACH)

LEACH & Cluster head

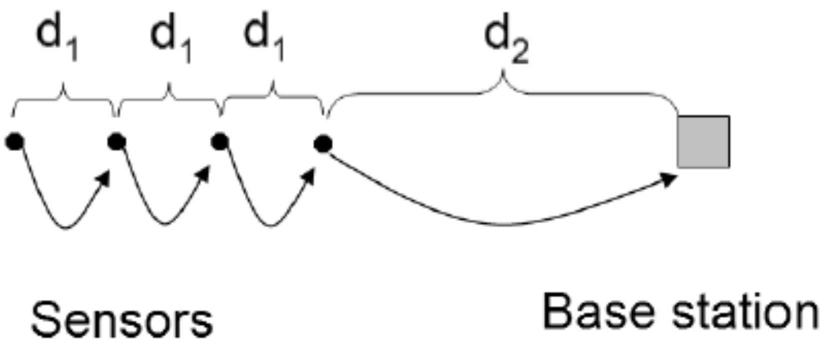
- A cluster-head collect data from their surrounding nodes and pass it on to the base station
- The job of cluster-head rotates



Direct vs. Minimum Transmission



(a) Direct transmission



(b) Minimum transmission energy

- The amount of energy used in figure (a) can be modeled by this formula:
  - $\varepsilon_{amp}k(3d_1 + d_2)^2$
- Whereas the amount of energy used in figure (b) uses this formula:
  - $\varepsilon_{amp}k(3d_1^2 + d_2^2)$

LEACH's Two Phases

- The LEACH network has two phases: the set-up phase and the steady-state
  - The Set-Up Phase
    - Where cluster-heads are chosen
  - The Steady-State
    - The cluster-head is maintained
    - When data is transmitted between nodes

Selecting cluster-head

- Cluster-heads can be chosen *stochastically* (randomly based) on this algorithm:

$$T(n) = \frac{P}{1 - P \times (r \bmod P^{-1})} \qquad \forall n \in G$$

$$T(n) = 0 \qquad \forall n \notin G$$

Where n is a random number between 0 and 1
P is the cluster-head probability and
G is the set of nodes that weren't cluster-heads the previous rounds

- R is the round number
- If n < T(n), then that node becomes a cluster-head
- The algorithm is designed so that each node becomes a cluster-head at least once

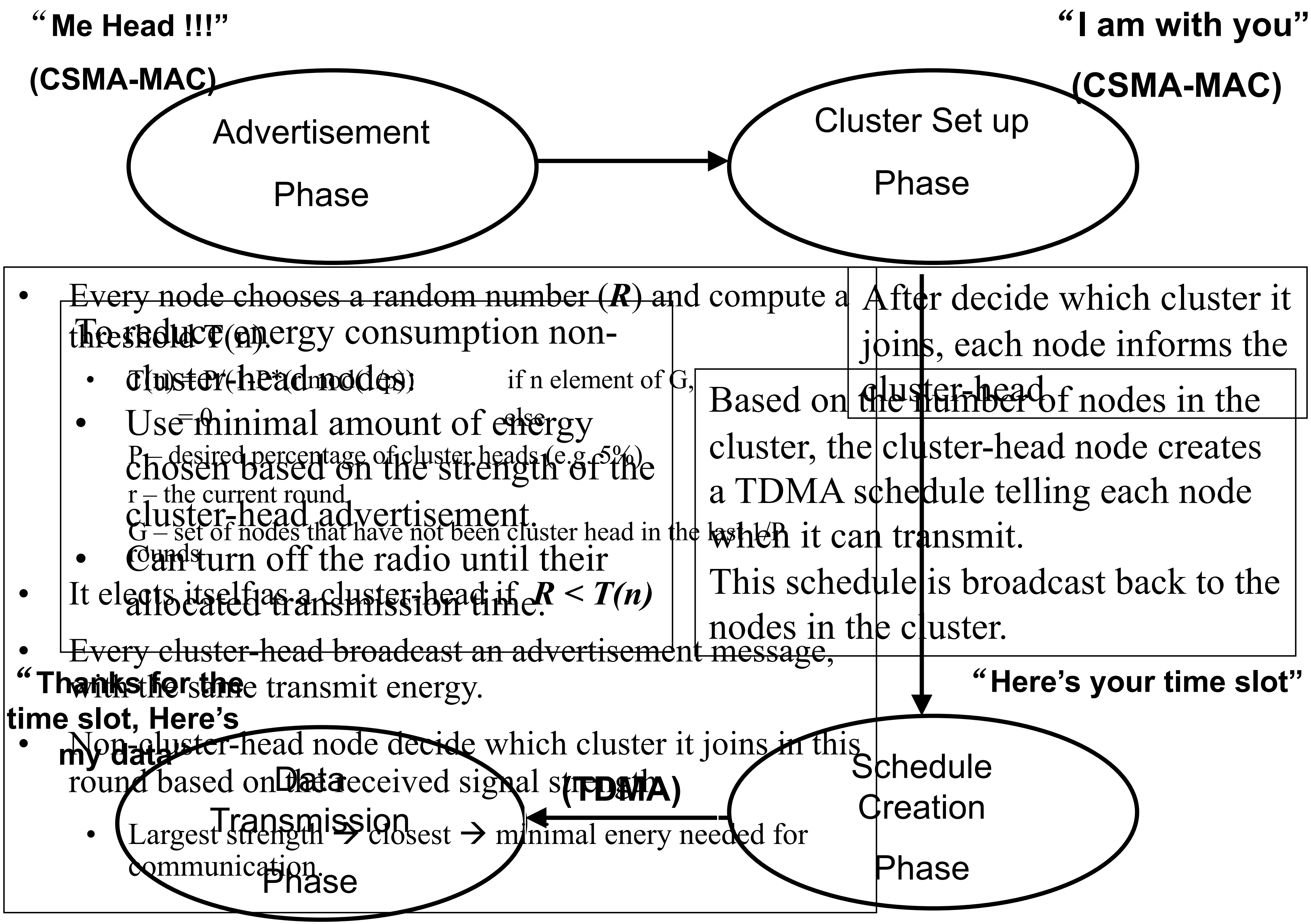
Example

```
p=0.05

0.0500 = 0.05/(1-0.05*0)
0.0526 = 0.05/(1-0.05*1)
0.0555 = 0.05/(1-0.05*2)
0.0588 = 0.05/(1-0.05*3)
0.0625 = 0.05/(1-0.05*4)
0.0666 = 0.05/(1-0.05*5)
0.0714 = 0.05/(1-0.05*6)
0.0769 = 0.05/(1-0.05*7)
0.0833 = 0.05/(1-0.05*8)
0.0909 = 0.05/(1-0.05*9)
0.1000 = 0.05/(1-0.05*10)
...
0.5000 = 0.05/(1-0.05*18)
1.0000 = 0.05/(1-0.05*19)
```

- Number of clusters may not fixed in any round.

$$T(n) = \begin{cases} \dfrac{P}{1 - P[r \bmod (1/P)]} & \text{if } n \in G, \\ 0 & \text{otherwise,} \end{cases}$$



- Every node chooses a random number (***R***) and compute a threshold T(n).
  - T(n) = P/(1-P*(r mod 1/P)) if n element of G,
    = 0 else
    P – desired percentage of cluster heads (e.g. 5%)
    r – the current round
    G – set of nodes that have not been cluster head in the last 1/P rounds
- It elects itself as a cluster-head if ***R < T(n)***
- Every cluster-head broadcast an advertisement message, with the same transmit energy.
- Non-cluster-head node decide which cluster it joins in this round based on the received signal strength.
  - Largest strength → closest → minimal enery needed for communication.

To reduce energy consumption non-cluster-head nodes:

- Use minimal amount of energy chosen based on the strength of the cluster-head advertisement.
- Can turn off the radio until their allocated transmission time.

After decide which cluster it joins, each node informs the cluster-head

Based on the number of nodes in the cluster, the cluster-head node creates a TDMA schedule telling each node when it can transmit.
This schedule is broadcast back to the nodes in the cluster.

Modified from http://faculty.cs.tamu.edu/dzsong/teaching/fall2004/netbot/John_G.ppt

Optimize selection

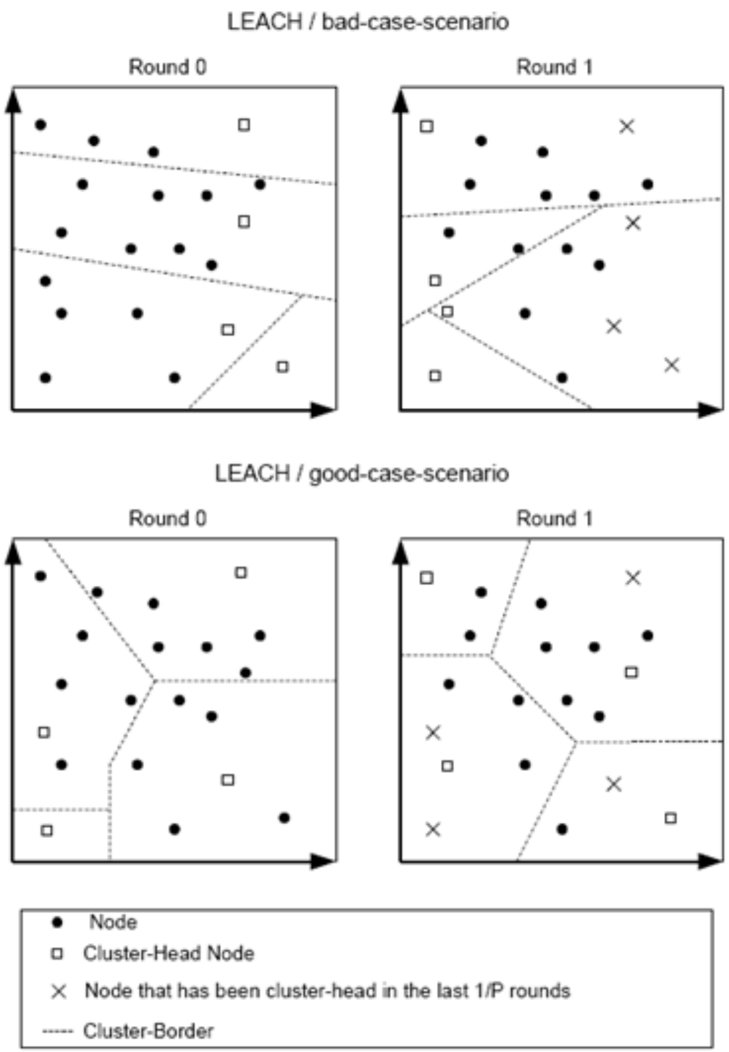
- A modified version of this protocol is known as LEACH-C (or LEACH Centralized)

- This version has a *deterministic* threshold algorithm, which takes into account the amount of energy in the node

$$T(n)_{new} = \frac{P}{1 - P \times (r \bmod P^{-1})} \frac{E_{n\_current}}{E_{n\_max}}$$

Where $E_{n\_current}$ is the current amount of energy and E_{n_max} is the initial amount of energy

Location of CH is important



- ❑ While neither of these diagrams is the optimum scenario, the second is better because the cluster-heads are spaced out and the network is more properly sectioned