Advanced Internet and IoT Technologies

- Routing in Wireless Edge Networks -

Prof. Dr. Thomas Schmidt

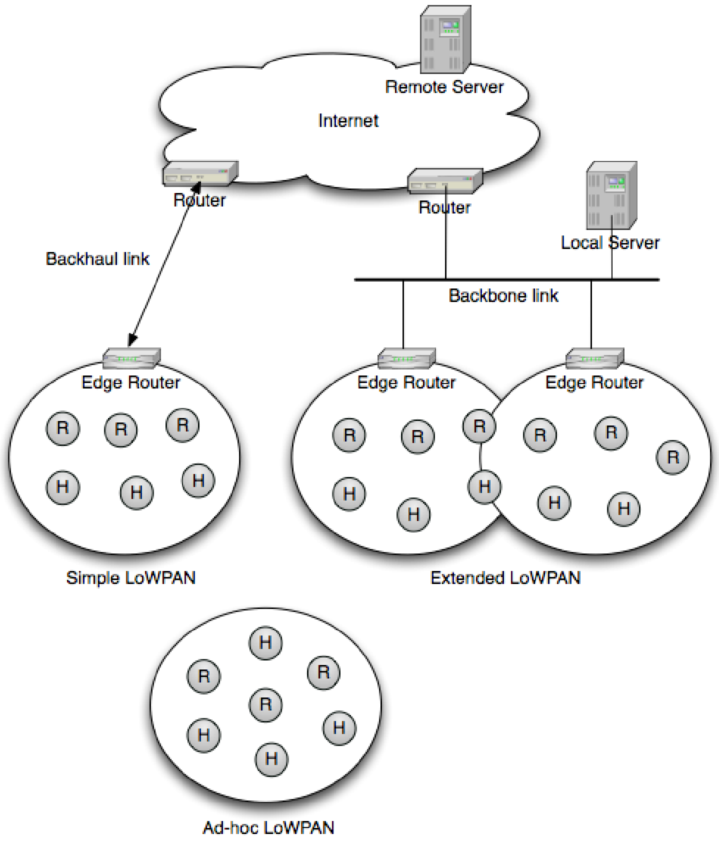
http://inet.haw-hamburg.de | t.schmidt@haw-hamburg.de

Agenda

- 🕐 Routing at the Wireless Edge
  - ➦ Motivation and Characteristics
- 🕑 Mobile Ad-hoc Networks - MANETs
- 🕒 Routing towards a Gateway

Source of animations on traditional MANET routing: INFOCOM tutorial by Nitin H. Vaidya

Routing at the Wireless Edge



Wireless edge networks in the IoT may be

- single hop – no routing needed
- spontaneous –ad hoc routing needed
- mobile – routing must cope with mobility
- centered toward a stationary gateway – may simplify routing control significantly

Different routing approaches attempt to optimize for the various use cases

Mobile Ad-hoc Networks: Many Variations of MANETs

Fully Symmetric Environment

- all nodes have identical capabilities and responsibilities

Asymmetric Capabilities

- transmission ranges and radios may differ (→ asymmetric links)
- battery life at different nodes may differ
- processing capacity may be different at different nodes
- speed of movement

Asymmetric Responsibilities

- only some nodes may route packets
- some nodes may act as leaders of nearby nodes (e.g., cluster head)

Varying Traffic Characteristics

Performance Properties of Multihop Wireless Networks

One-Hop Capacity:
Consider MANET of *n* equal nodes, each acting as router, with constant node density. Then the One-Hop Capacity grows linearly ➔ O(n)

Total Capacity surprisingly low:

- Consider MANET of *n* equal nodes, each acting as router in an *optimal* set-up, then the Node Capacity to reach an arbitrary destination reads ➔ O$(1/\sqrt{n})$
- Node Capacity further decreases under wireless transmission ➔ O$(1/\sqrt{(n \ln(n))})$

Unicast Routing in the IoT - Why is it different ?

Host mobility

- link failure/repair due to mobility may have different characteristics than those due to other causes

Rate of link failure/repair may be high when nodes move fast

New performance criteria may be used

- route stability despite mobility
- energy consumption

Many routing protocols proposed – no universal solution

Routing Protocols

Proactive protocols

- Determine routes independent of traffic pattern
- Traditional link-state and distance-vector routing protocols are proactive

Reactive protocols

- Maintain routes only if needed
- Saves bandwidth and energy at sparse scenarios

Hybrid protocols

- Proactive route discovery for the relevant, e.g. Gateways
- Reactive route discovery for the remainders

Trade-Offs

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 after X attempts to send

Overhead of route discovery/maintenance

- Reactive protocols may have lower overhead since routes are determined only if needed
- Proactive protocols can (but not necessarily) result in higher overhead due to continuous route updating

Trade-off depends on the traffic and mobility patterns

Flooding for Data Delivery

Sender S broadcasts data packet P to all its neighbors

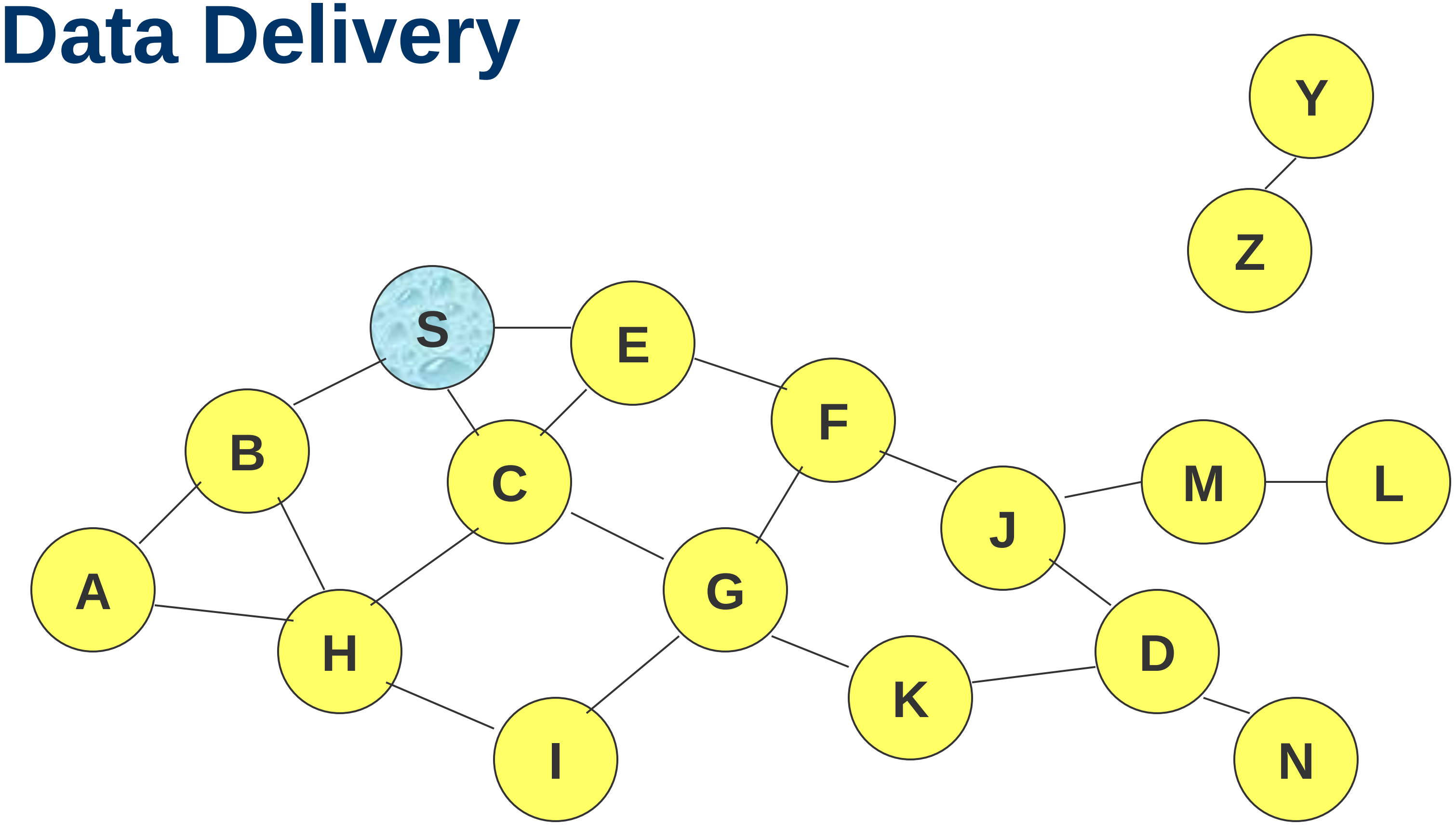
Each node receiving P forwards P to its neighbors

Sequence numbers used to avoid the possibility of forwarding the same packet more than once

Packet P reaches destination D provided that D is reachable from sender S

Node D does not forward the packet

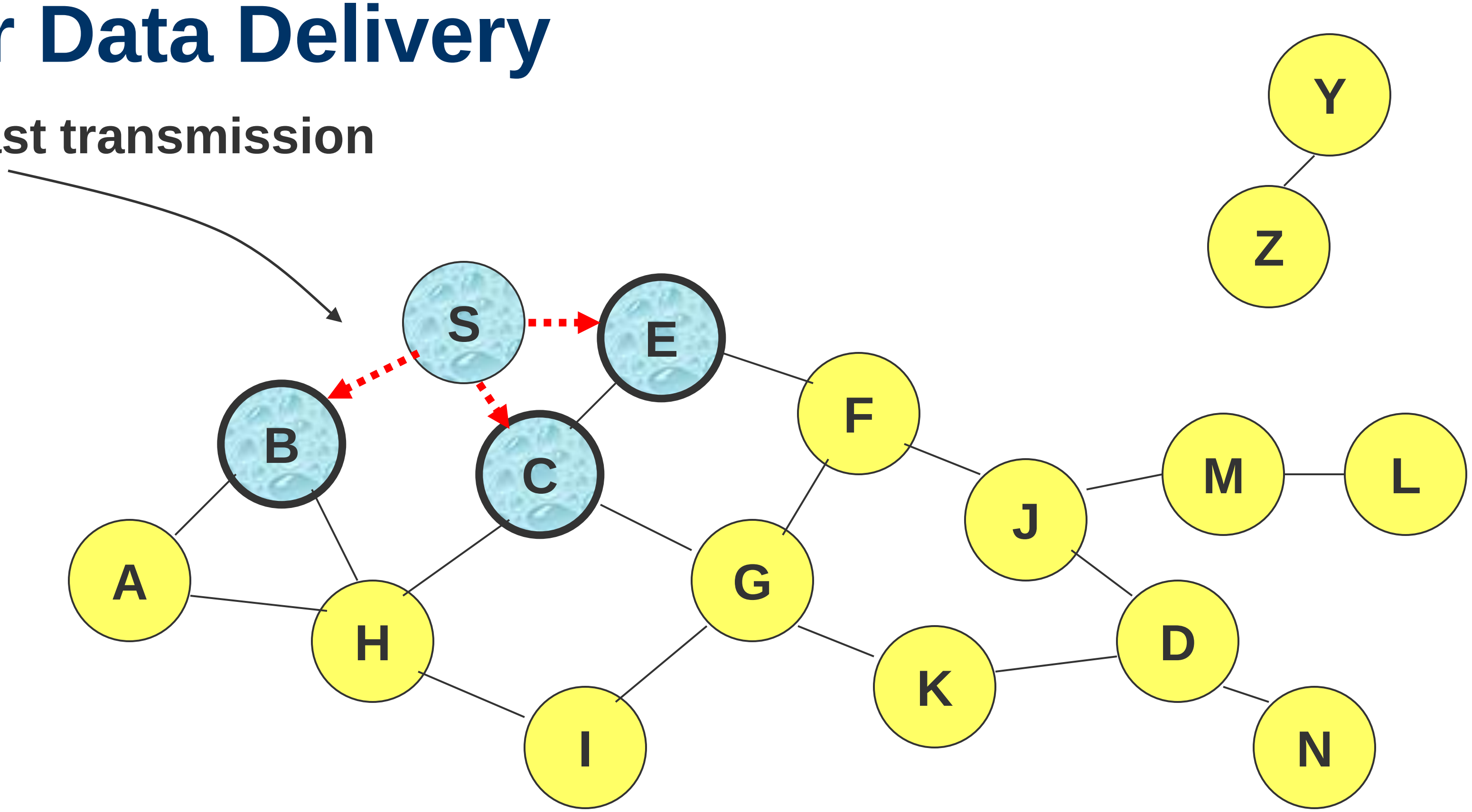
Flooding for Data Delivery



● Represents a node that has received packet P

— Represents that connected nodes are within each other’s transmission range

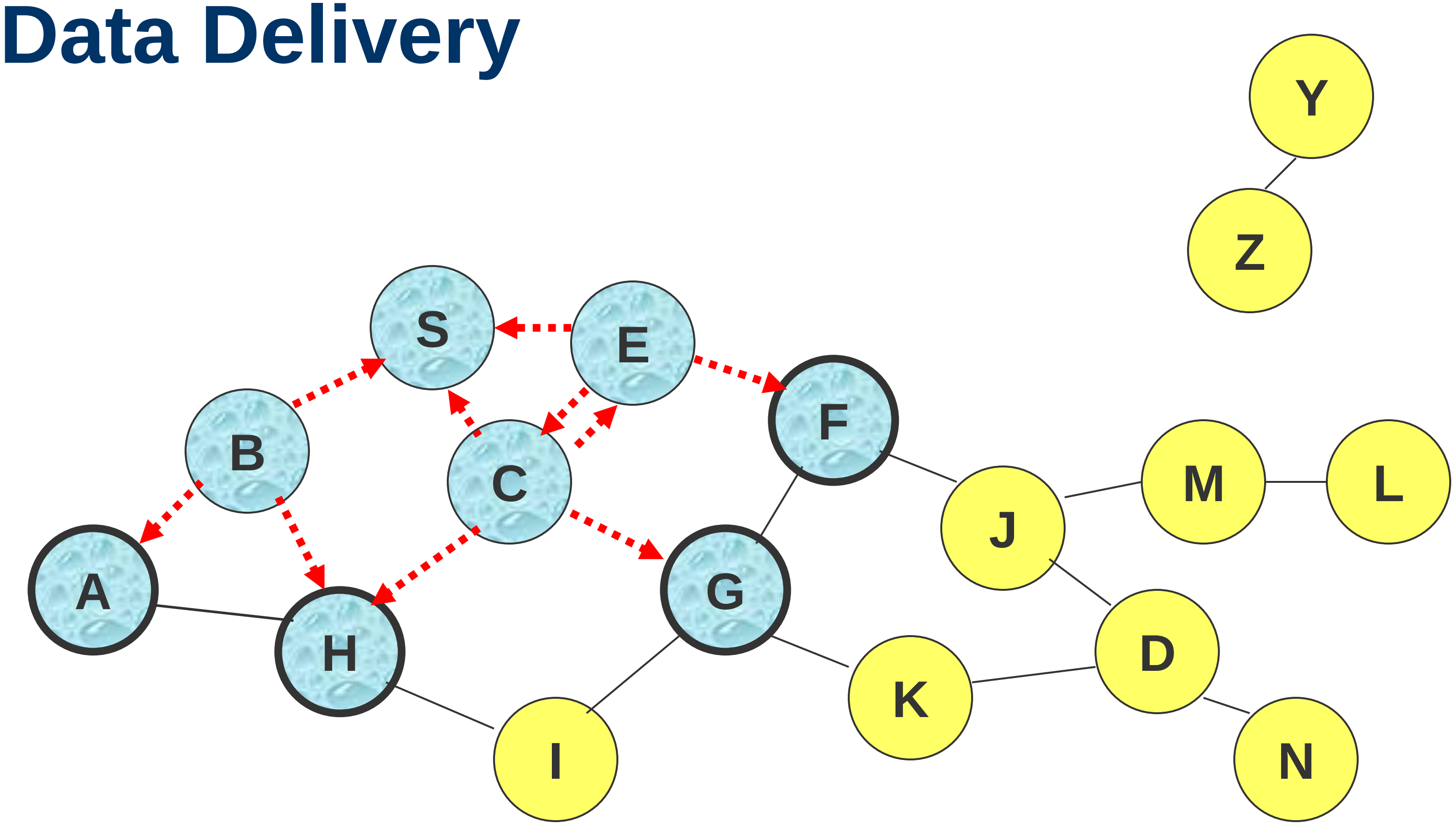
Flooding for Data Delivery


Broadcast transmission

● Represents a node that receives packet P for the first time

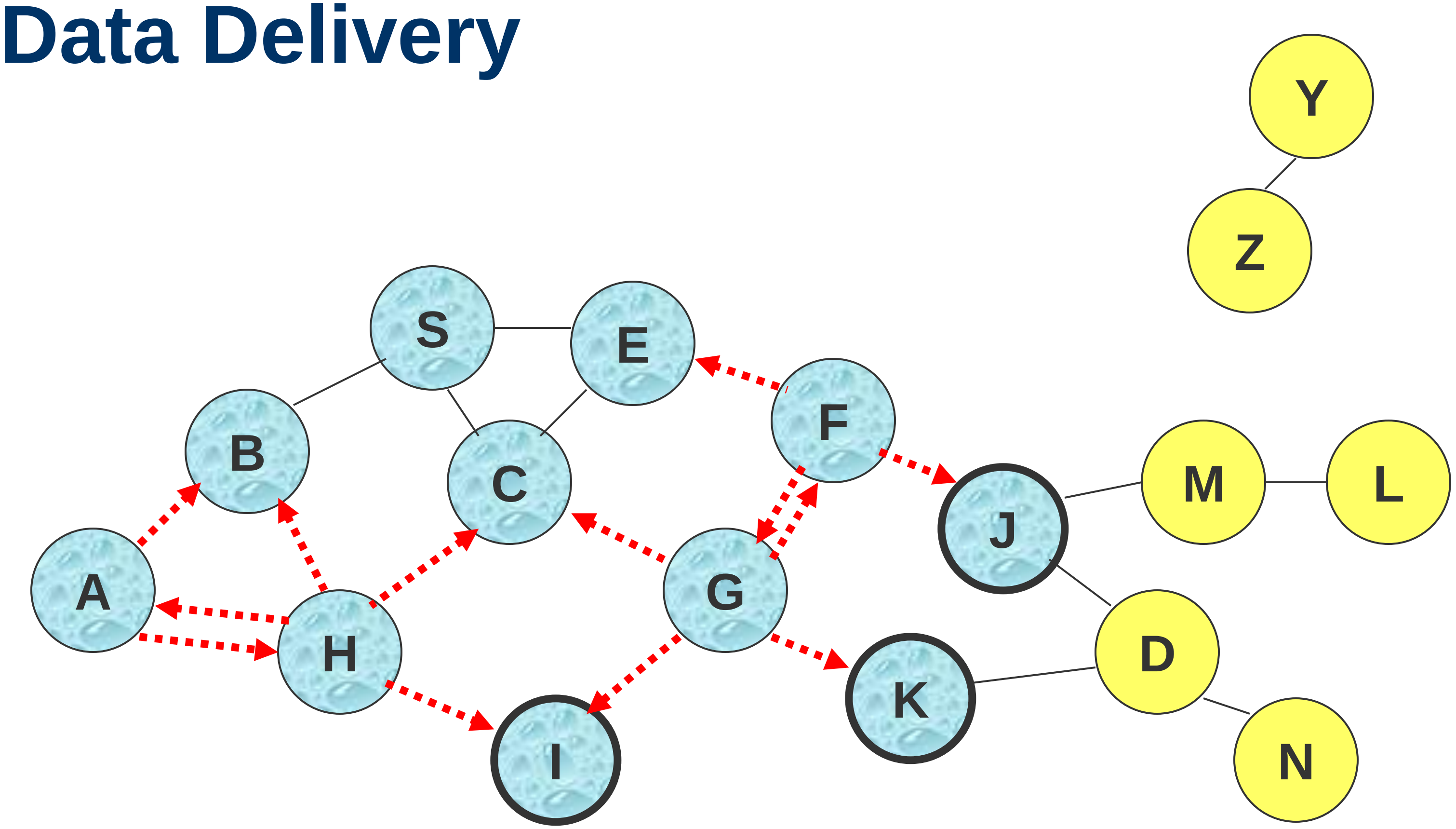
⇢ Represents transmission of packet P

Flooding for Data Delivery



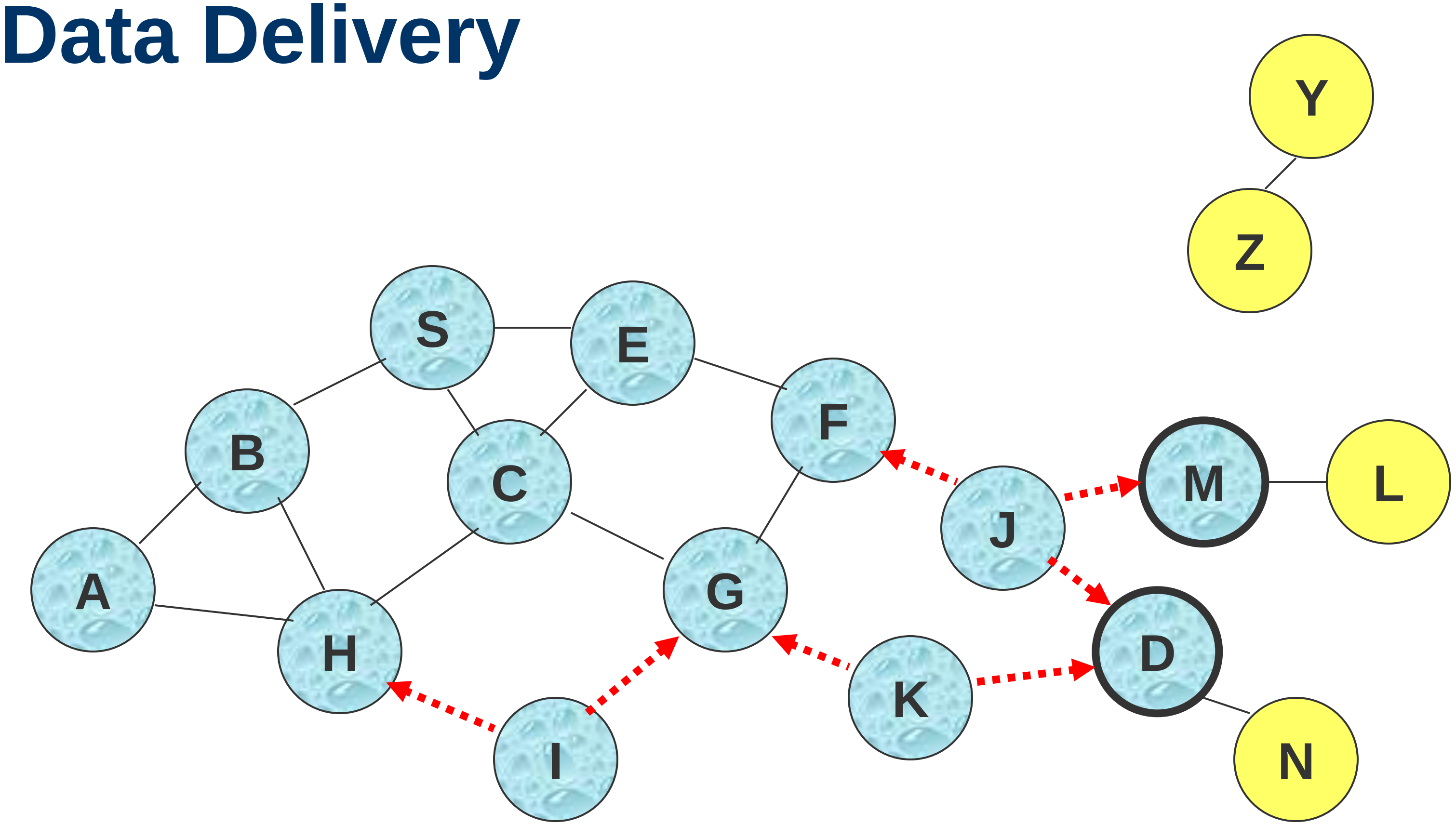
- **Node H receives packet P from two neighbors:**
  potential for collision

Flooding for Data Delivery



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

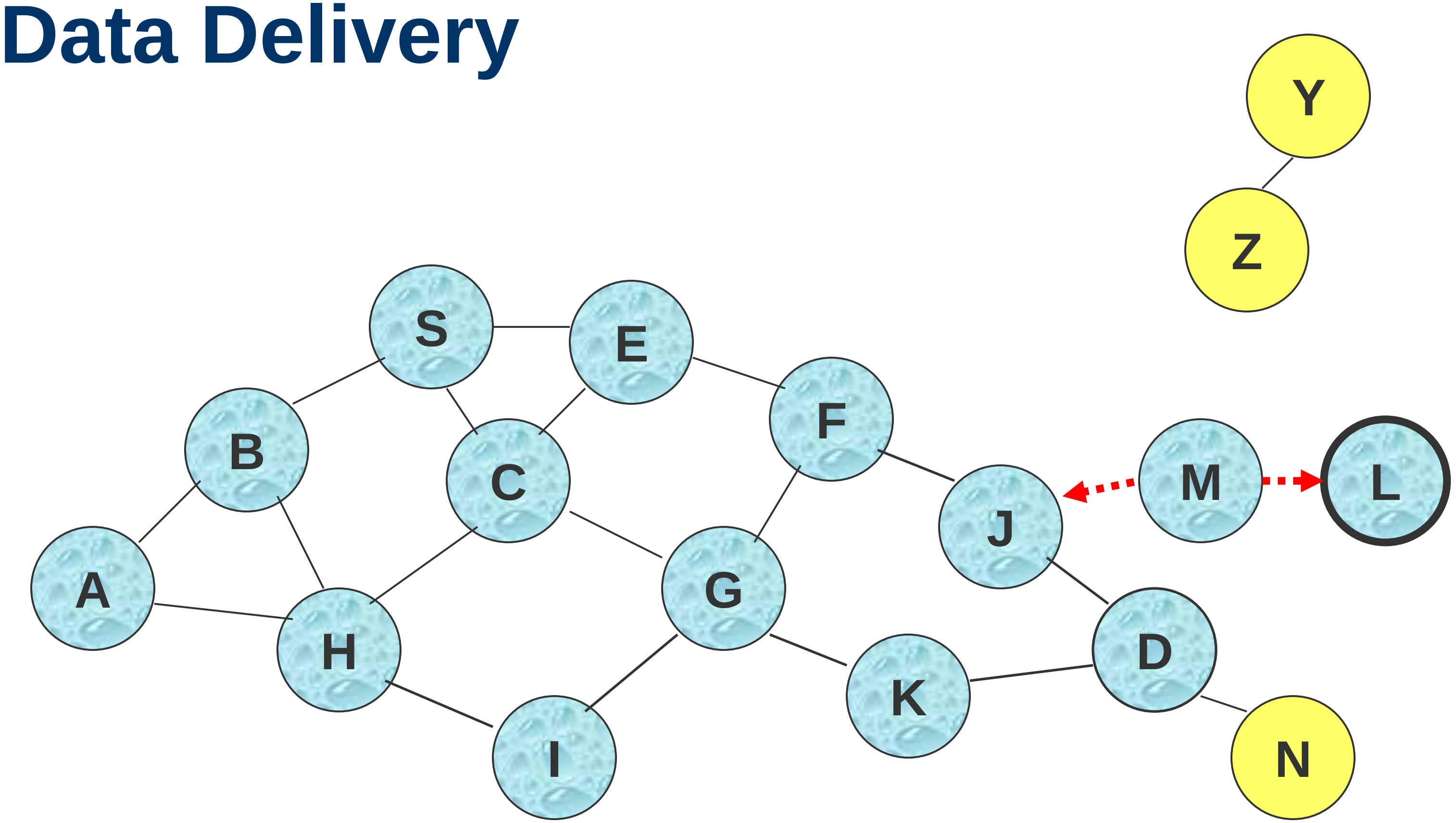
Flooding for Data Delivery



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

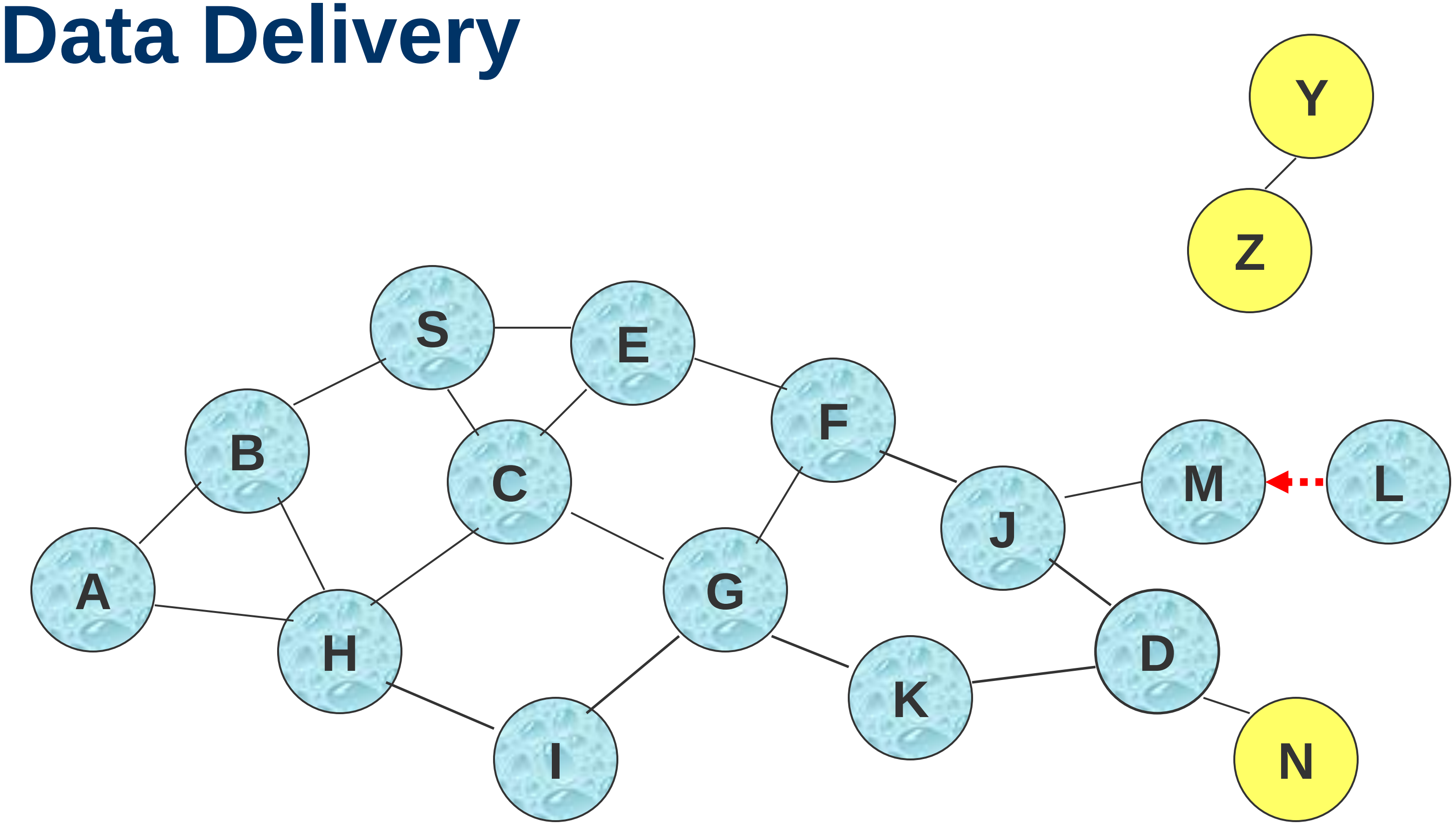
  => Packet P may not be delivered to node D at all, despite the use of flooding

Flooding for Data Delivery



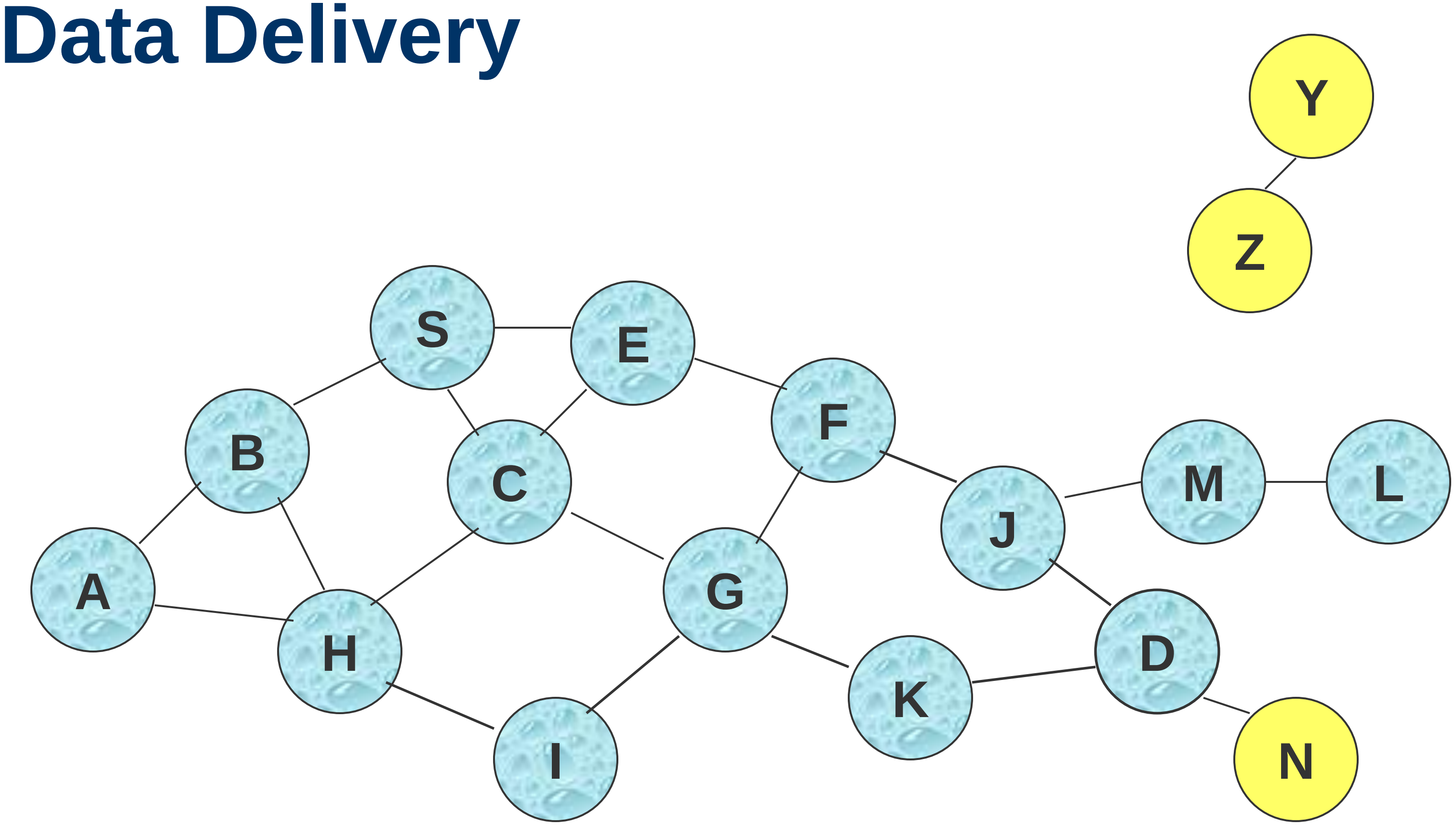
- Node D **does not forward** packet P, because node D is the intended destination of packet P

Flooding for Data Delivery



- Flooding completed
- Nodes unreachable from S do not receive packet P (e.g., node Z)
- Nodes for which all paths from S go through the destination D also do not receive packet P (example: node N)

Flooding for Data Delivery



- Flooding may deliver packets to too many nodes (in the worst case, all nodes reachable from sender may receive the packet)

Flooding for Data Delivery: Advantages

Simplicity

May be more efficient than other protocols when rate of information transmission is low enough that the overhead of explicit route discovery/maintenance incurred by other protocols is relatively higher

- this scenario may occur, for instance, when nodes transmit small data packets relatively infrequently, and many topology changes occur between consecutive packet transmissions

Potentially higher reliability of data delivery

- Because packets may be delivered to the destination on multiple paths

Flooding for Data Delivery: Disadvantages

Potentially, very high overhead

- Data packets may be delivered to too many nodes who do not need to receive them

Potentially lower reliability of data delivery

- Flooding uses broadcasting -- hard to implement reliable broadcast delivery without significantly increasing overhead
  - Broadcasting in most wireless MACs is unreliable
- In our example, nodes J and K may transmit to node D simultaneously, resulting in loss of the packet
  - in this case, destination would not receive the packet at all

Flooding of Control Packets

Many protocols perform (potentially *limited*) flooding of control packets, instead of data packets

The control packets are used to discover routes

Discovered routes are subsequently used to send data packet(s)

Overhead of control packet flooding is amortized over data packets transmitted between consecutive control packet floods

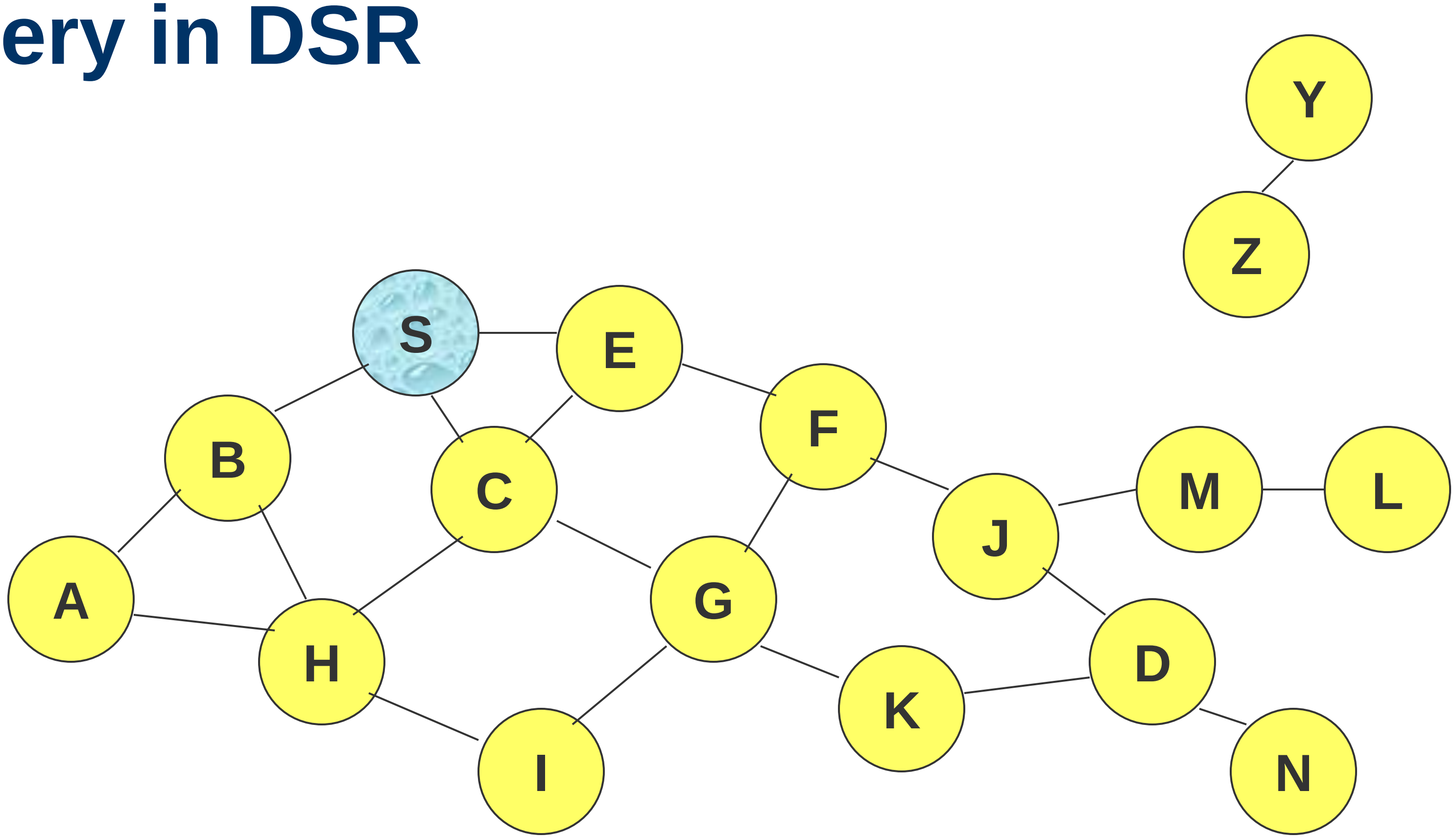
Dynamic Source Routing (DSR) [Johnson96]

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)

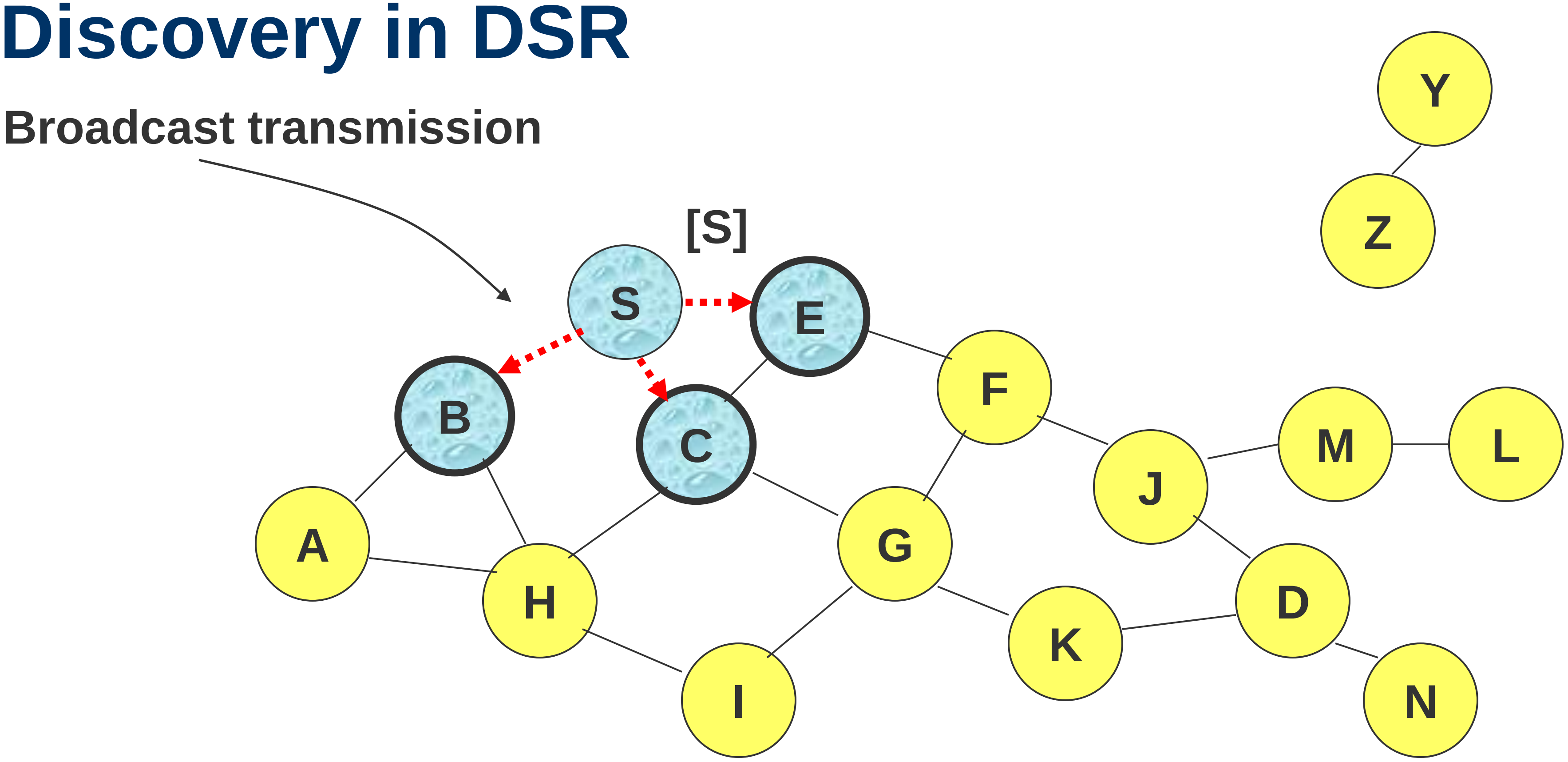
Each node appends own identifier when forwarding RREQ

Route Discovery in DSR



Represents a node that has received RREQ for D from S

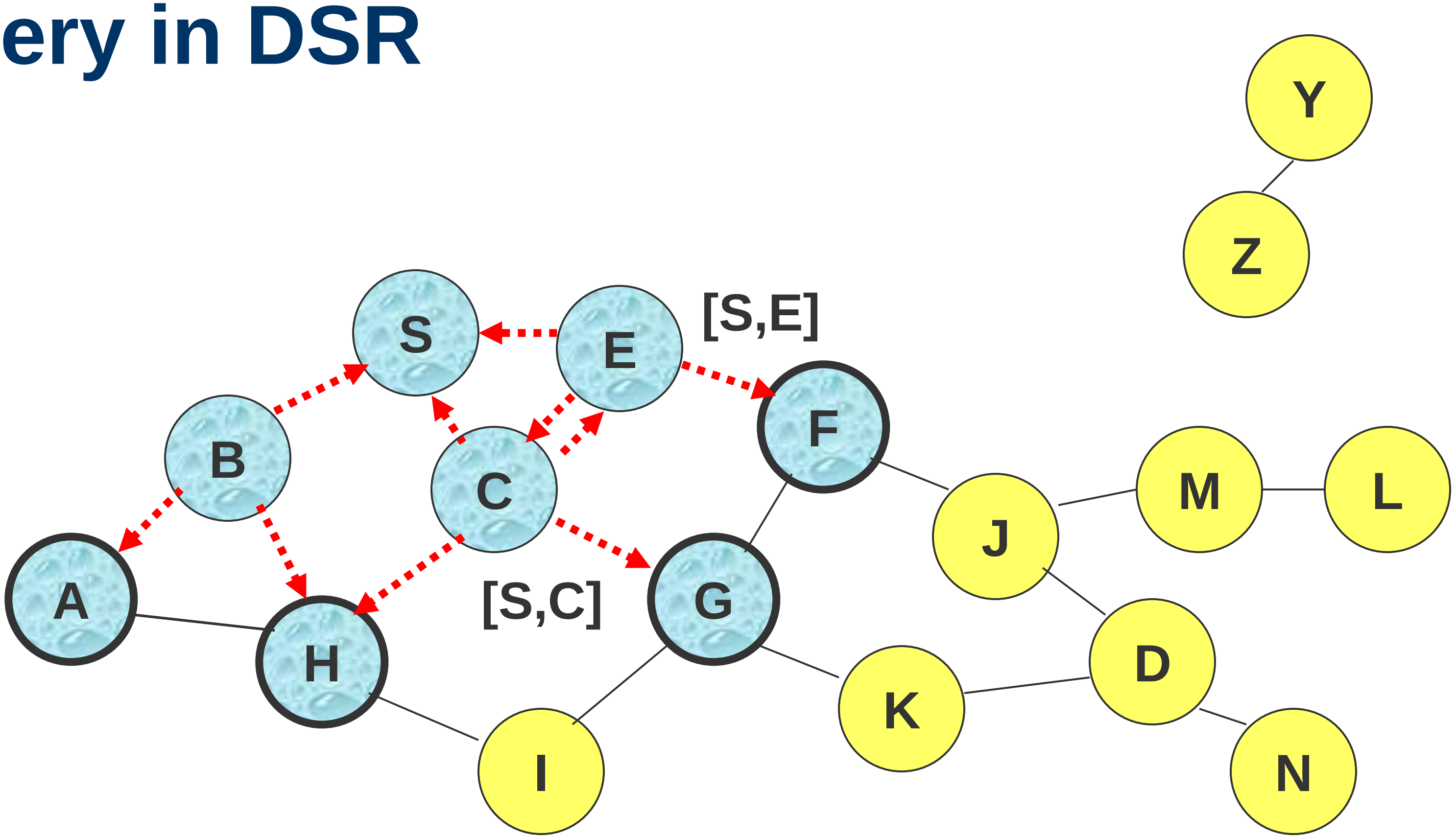
Route Discovery in DSR



Represents transmission of RREQ

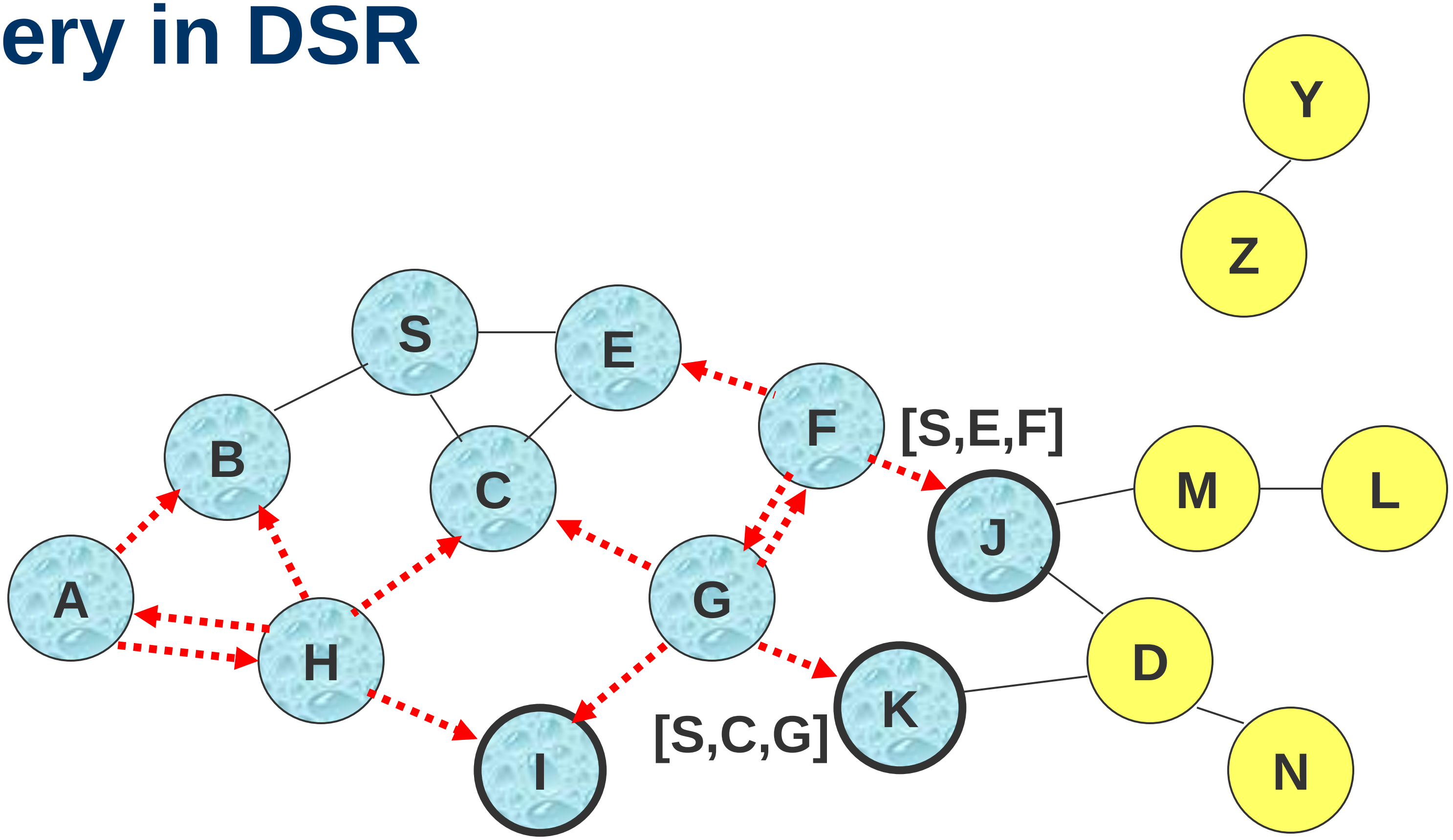
[X,Y] Represents list of identifiers appended to RREQ

Route Discovery in DSR



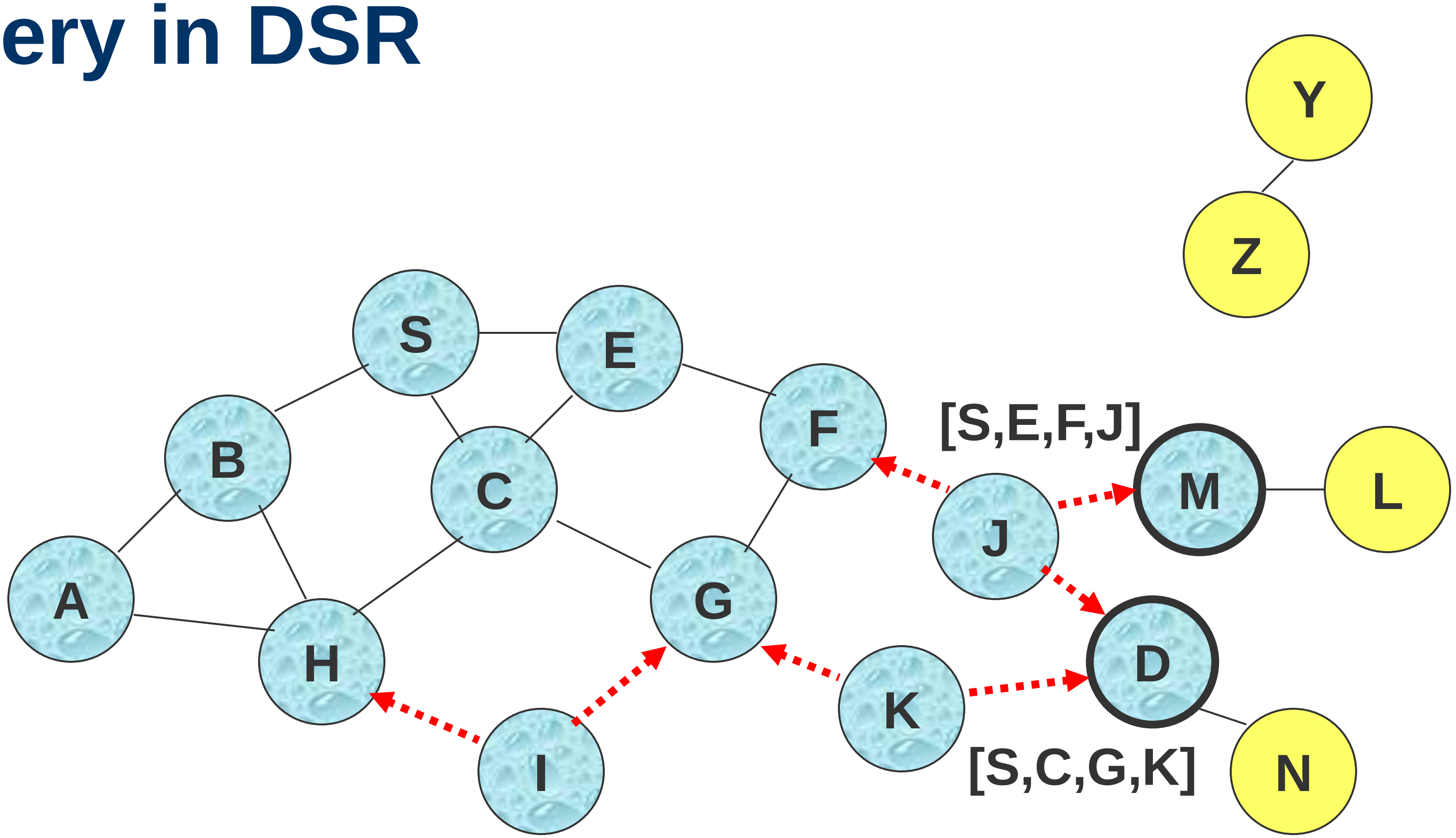
- **Node H receives packet RREQ from two neighbors:** **potential for collision**

Route Discovery in DSR



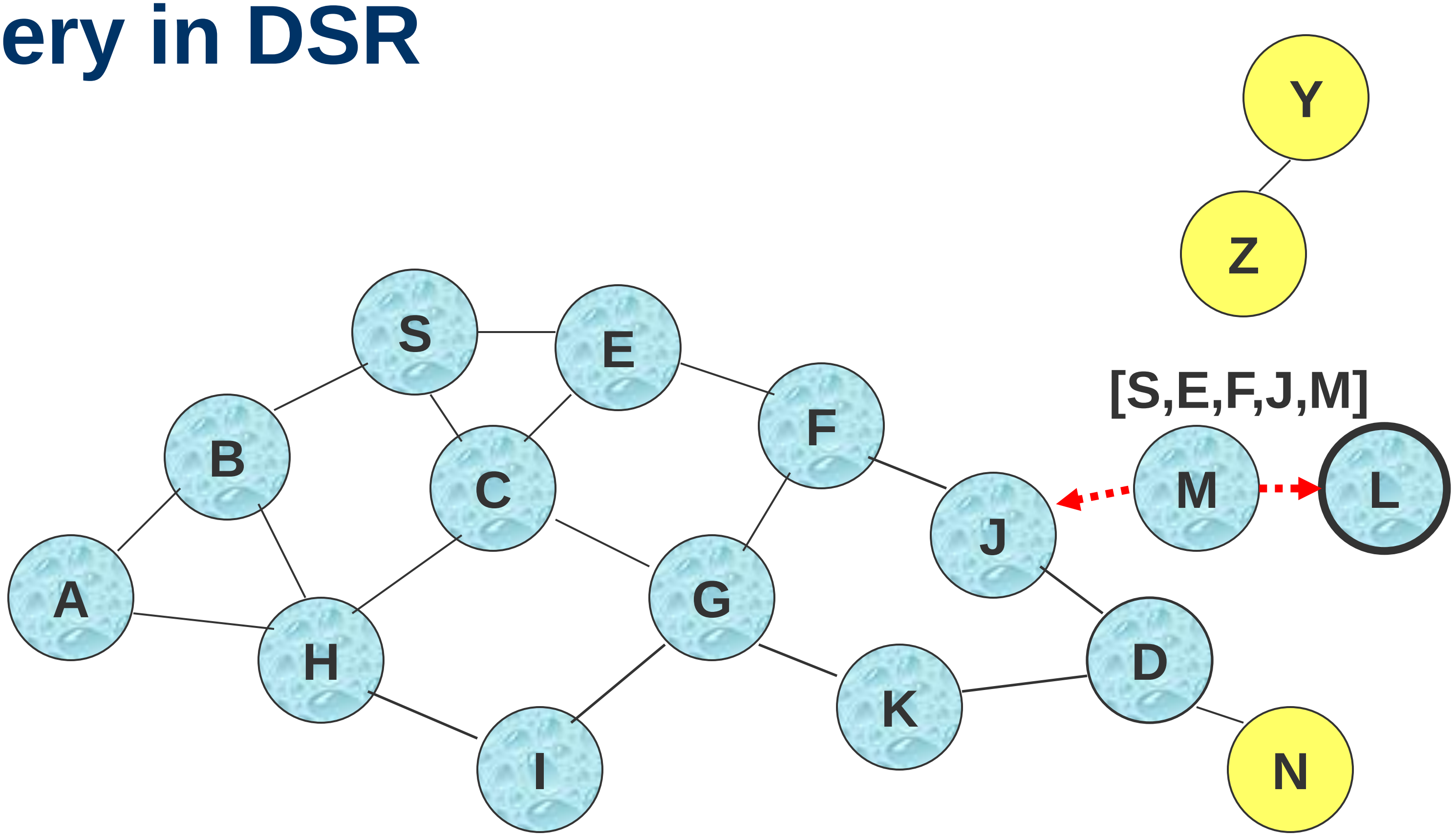
- Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once

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

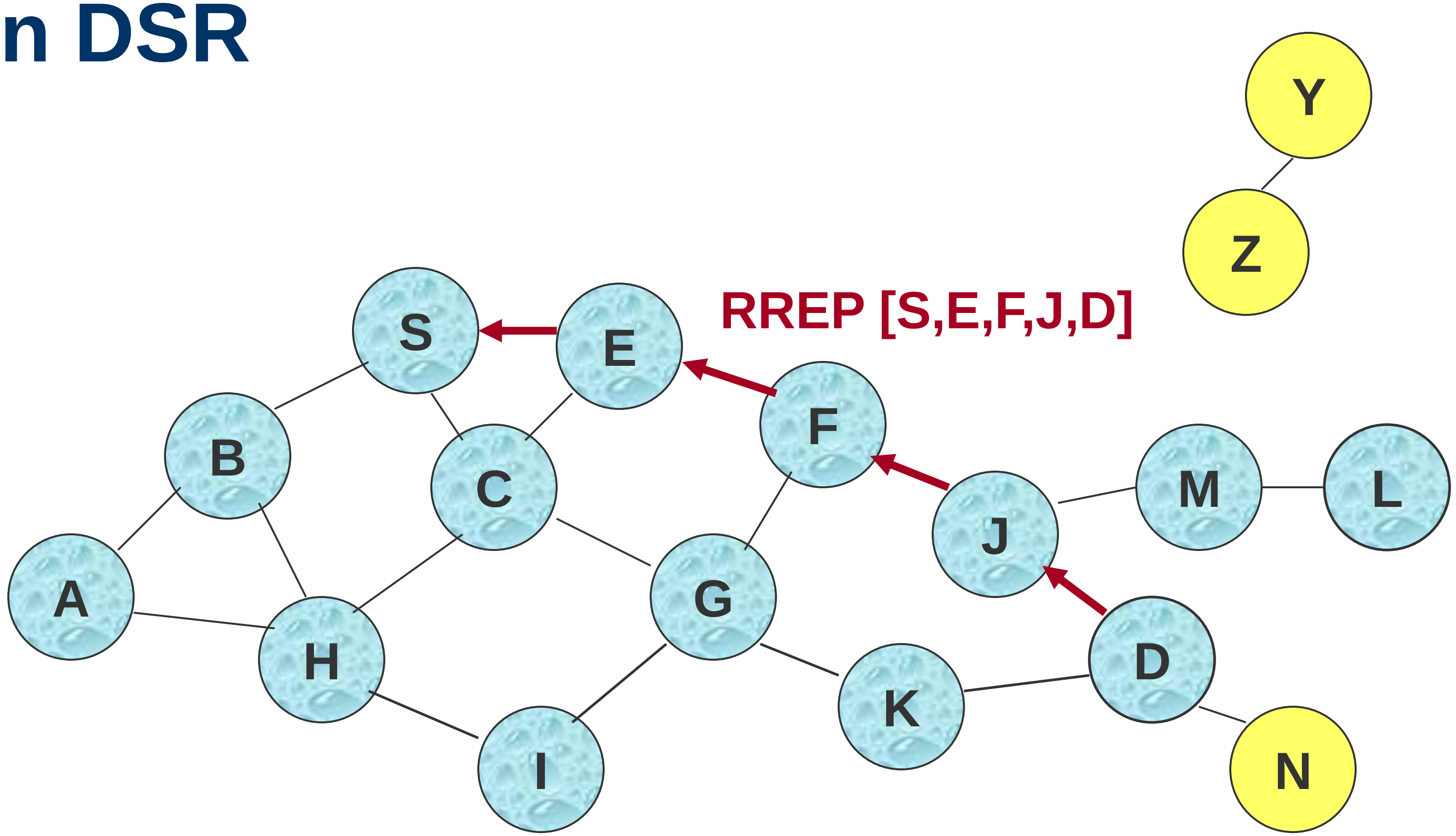
Route Discovery in DSR

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

Route Reply in DSR

Route Reply can be sent by reversing the route in Route Request (RREQ) only if links are guaranteed to be bi-directional

- To ensure this, RREQ should be forwarded only if it received on a link that is known to be bi-directional

If unidirectional (asymmetric) links are allowed, then RREP may need a route discovery for S from node D

- Unless node D already knows a route to node S
- If a route discovery is initiated by D for a route to S, then the Route Reply is piggybacked on the Route Request from D.

If IEEE 802.11 MAC is used to send data, then links need to be bi-directional (since Ack is used)

Dynamic Source Routing (DSR)

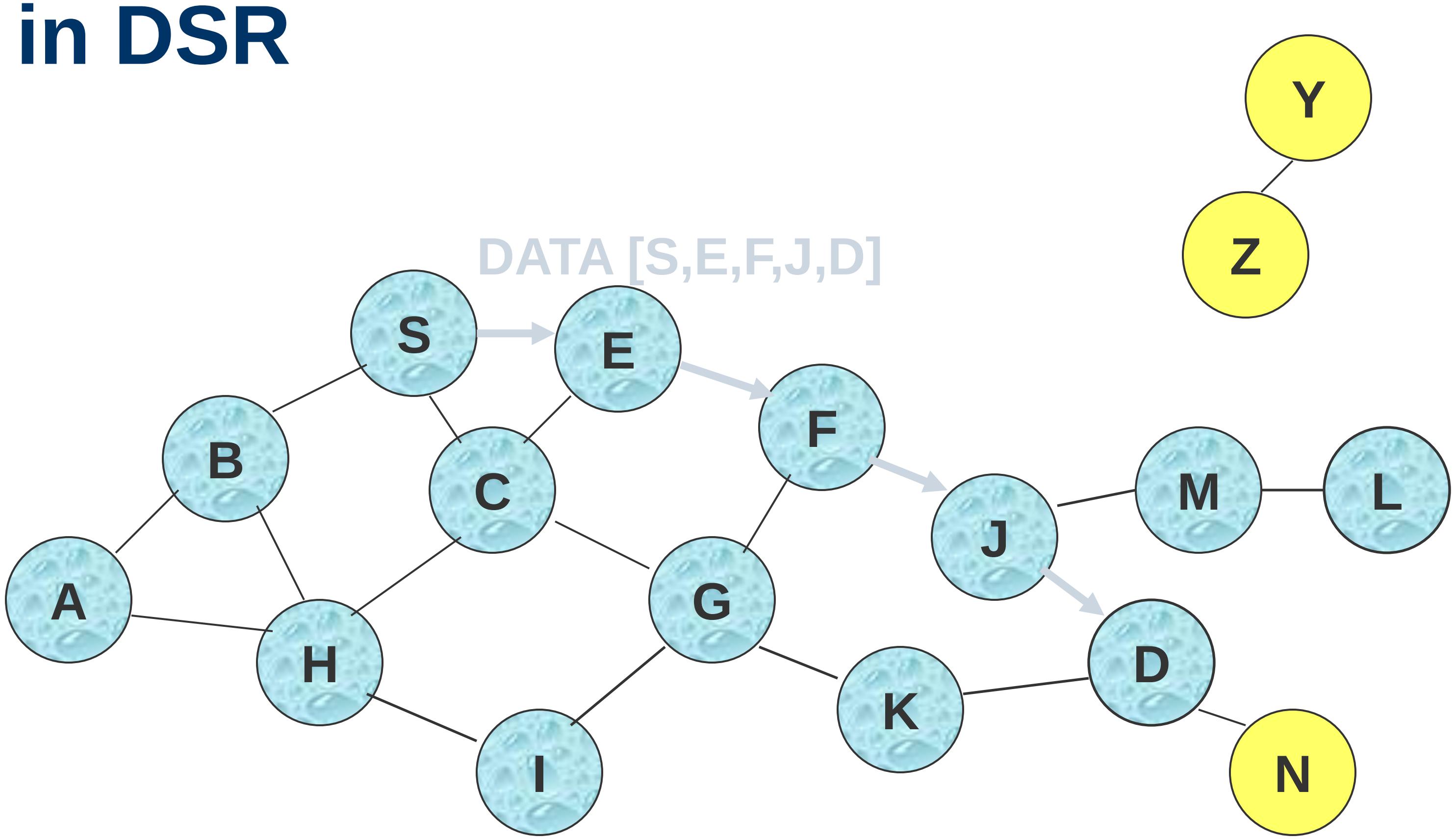
Node S on receiving RREP, caches the route included in the RREP

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: Advantages

Routes maintained only between nodes who need to communicate

- reduces overhead of route maintenance

Route caching can further reduce route discovery overhead

A single route discovery may yield many routes to the destination, due to intermediate nodes replying from local caches

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

Care must be taken to avoid 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
- Reply storm may be eased by preventing a node from sending RREP if it hears another RREP with a shorter route

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

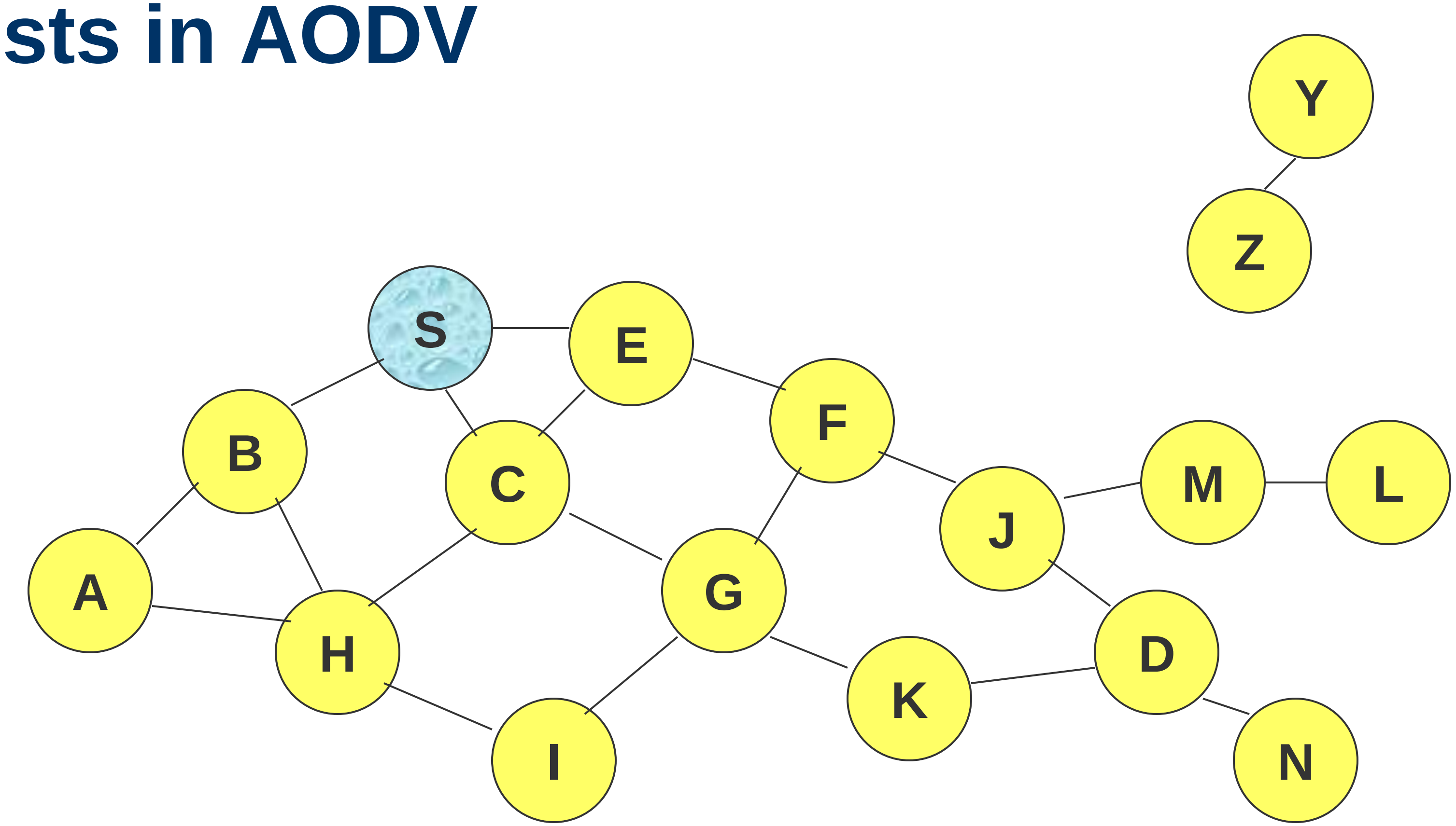
When a node re-broadcasts a Route Request, it sets up a reverse path pointing towards the source

- AODV assumes symmetric (bi-directional) links

When the intended destination receives a Route Request, it replies by sending a Route Reply

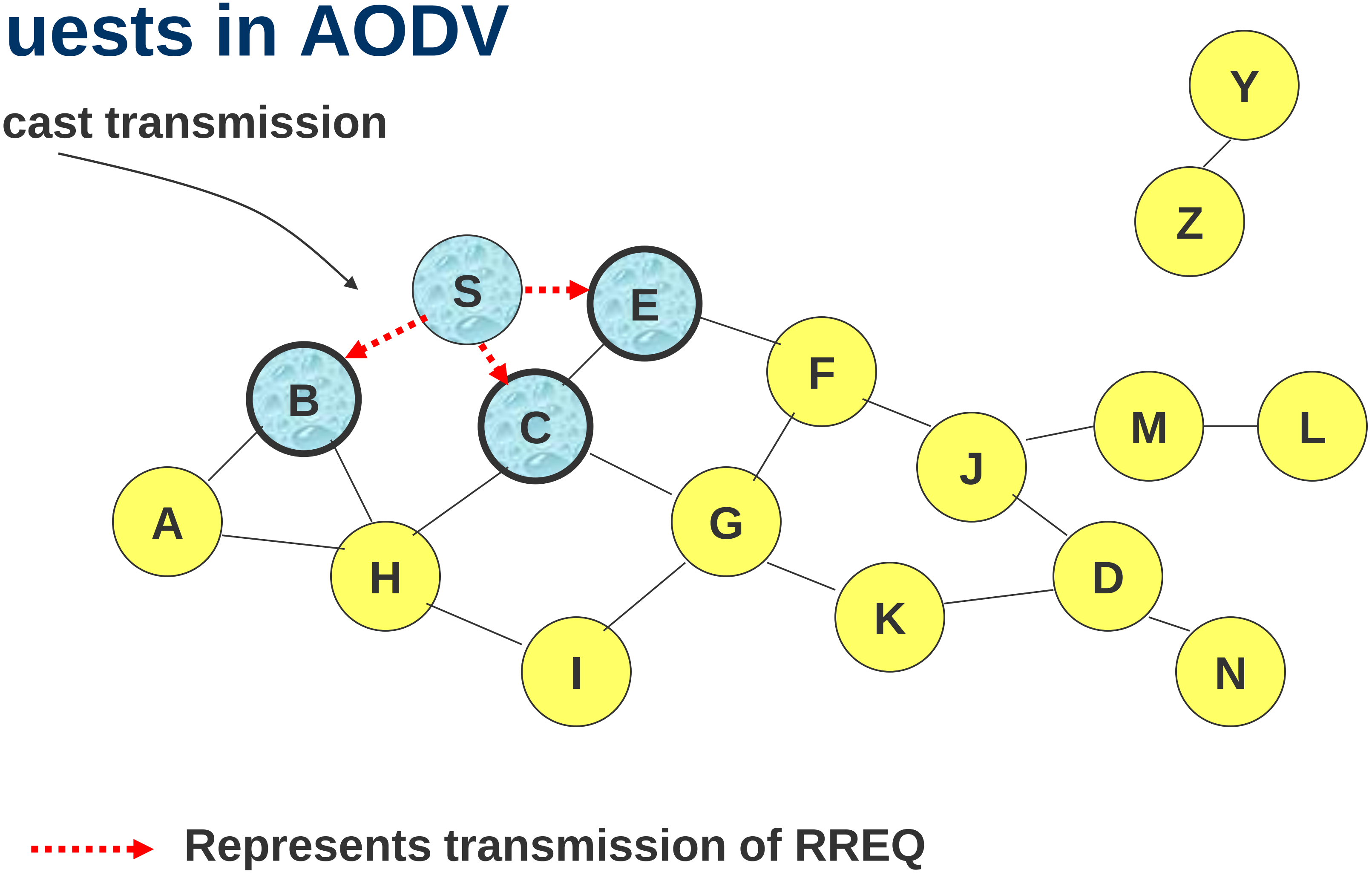
Route Reply travels along the reverse path set-up when Route Request is forwarded

Route Requests in AODV

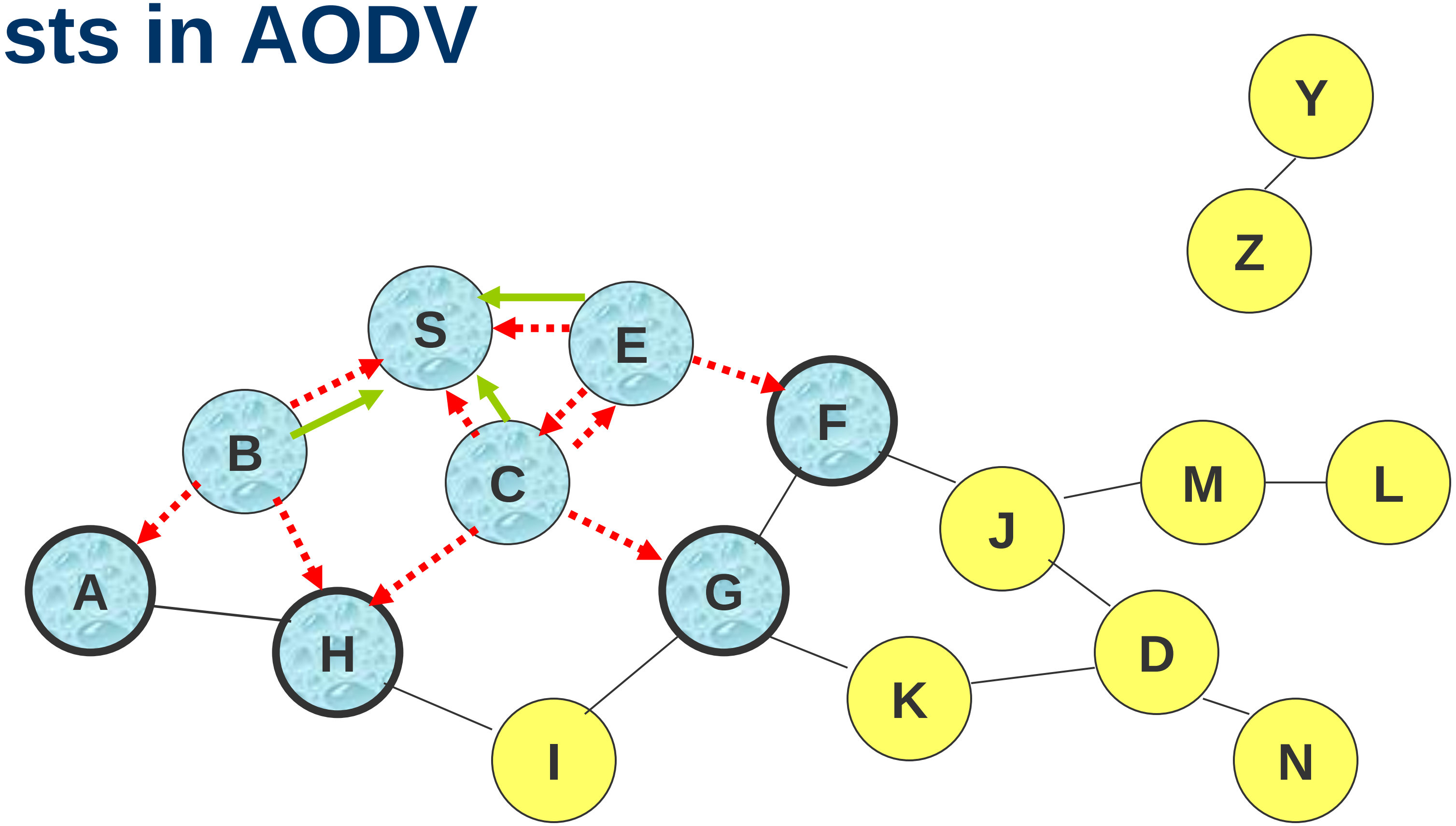


Represents a node that has received RREQ for D from S

Route Requests in AODV

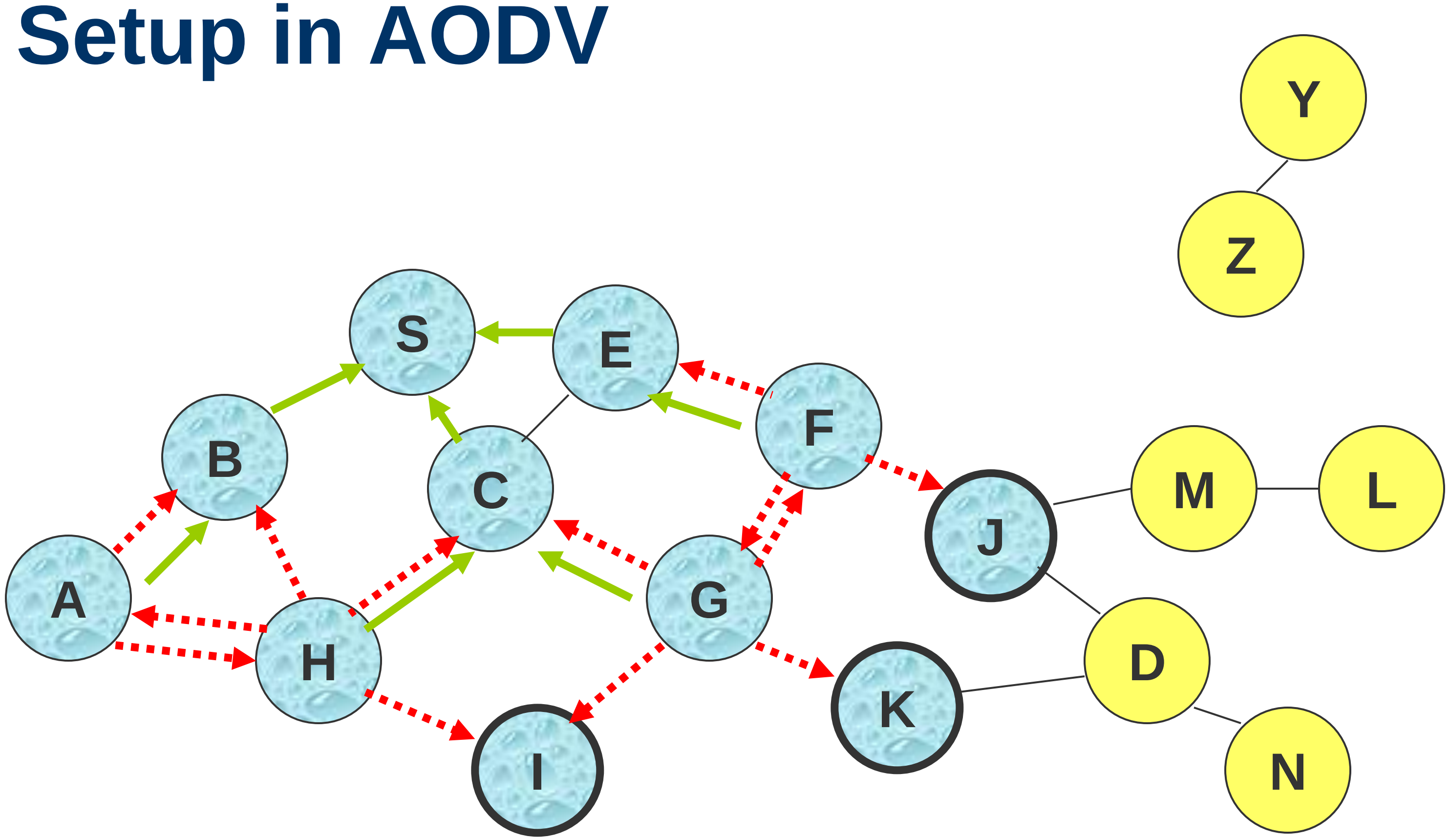


Route Requests in AODV



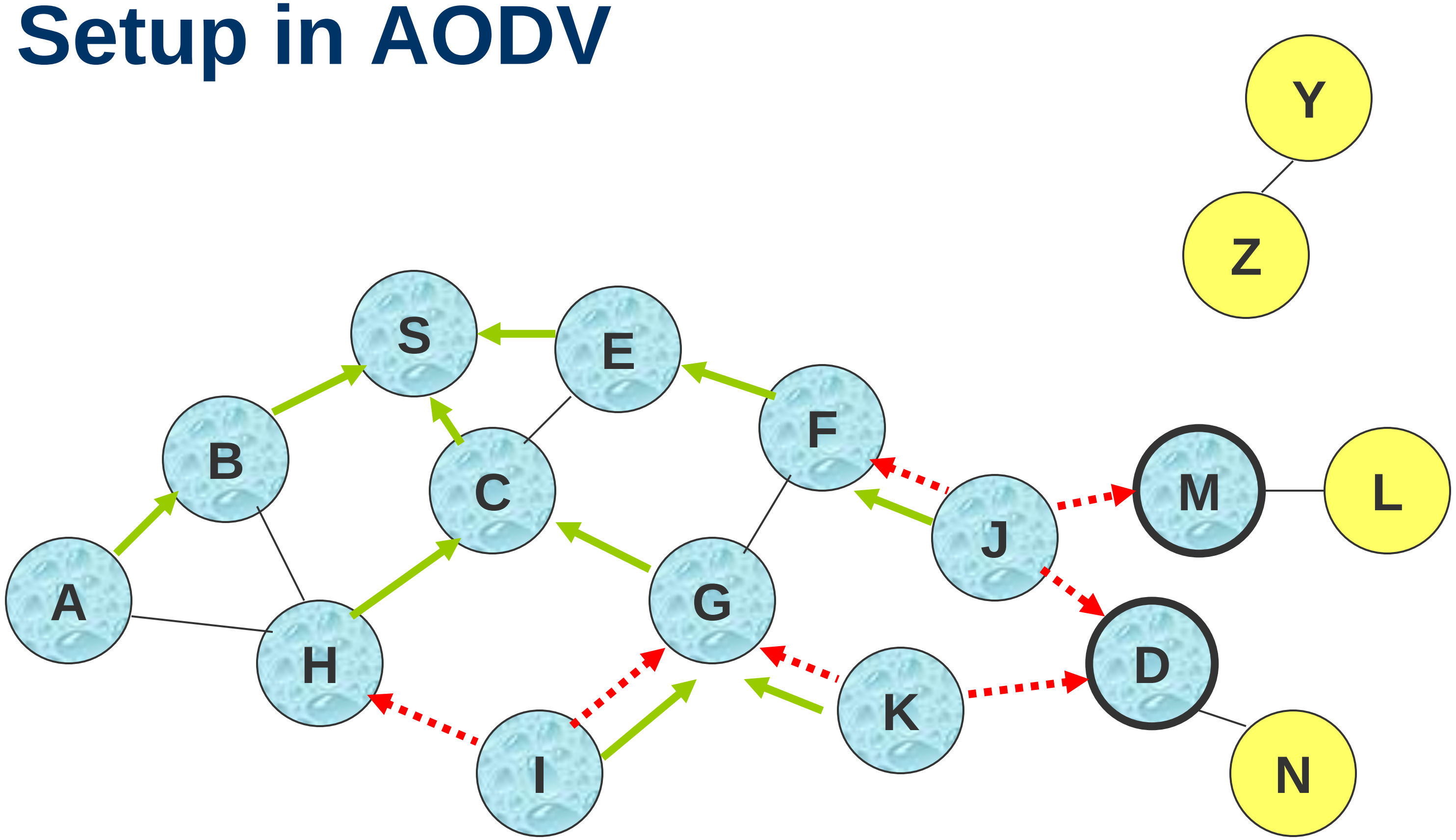
← Represents links on Reverse Path

Reverse Path Setup in AODV

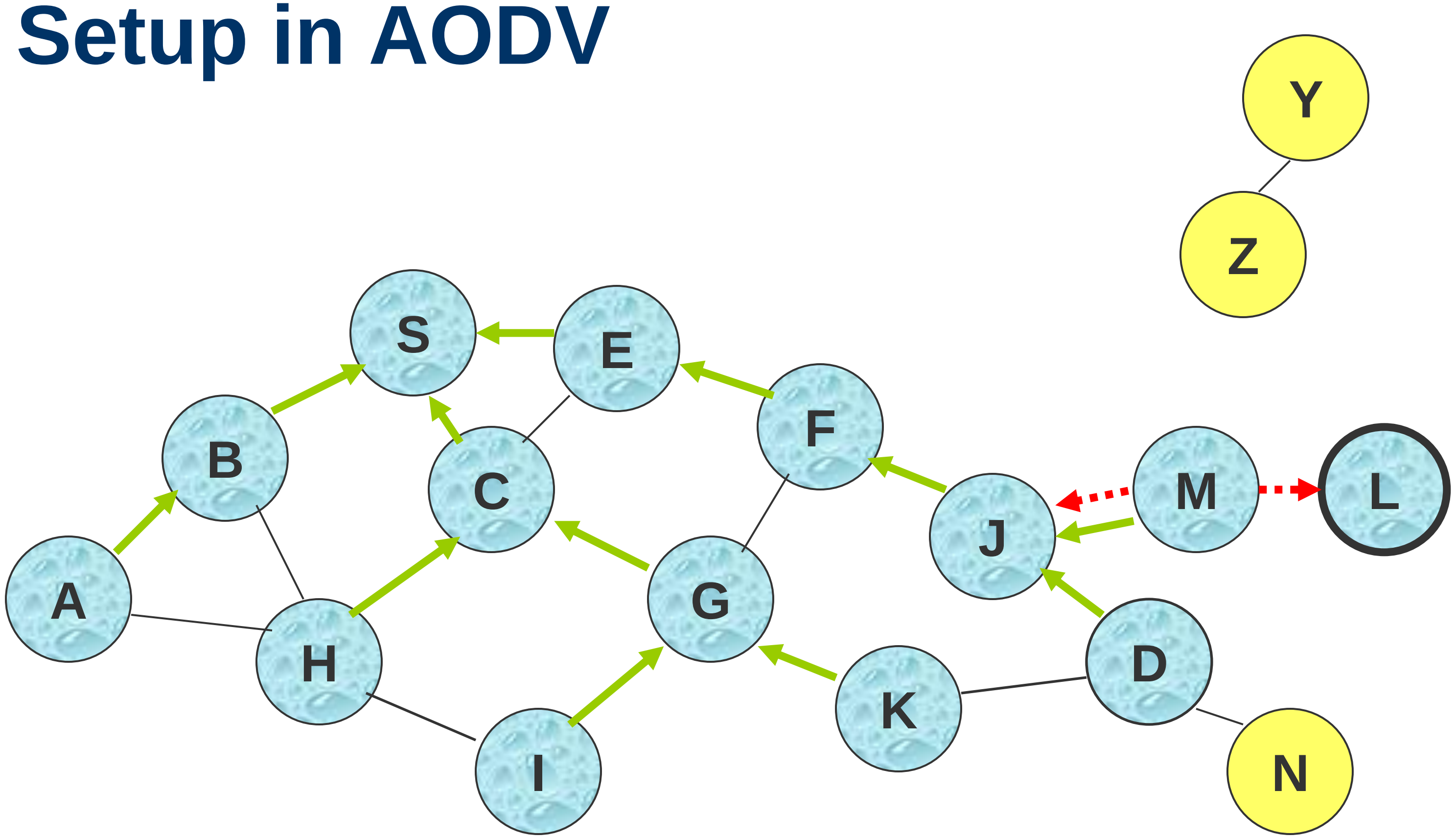


- Node C receives RREQ from G and H, but does not forward it again, because node C has <span>already forwarded RREQ</span> once

Reverse Path Setup in AODV

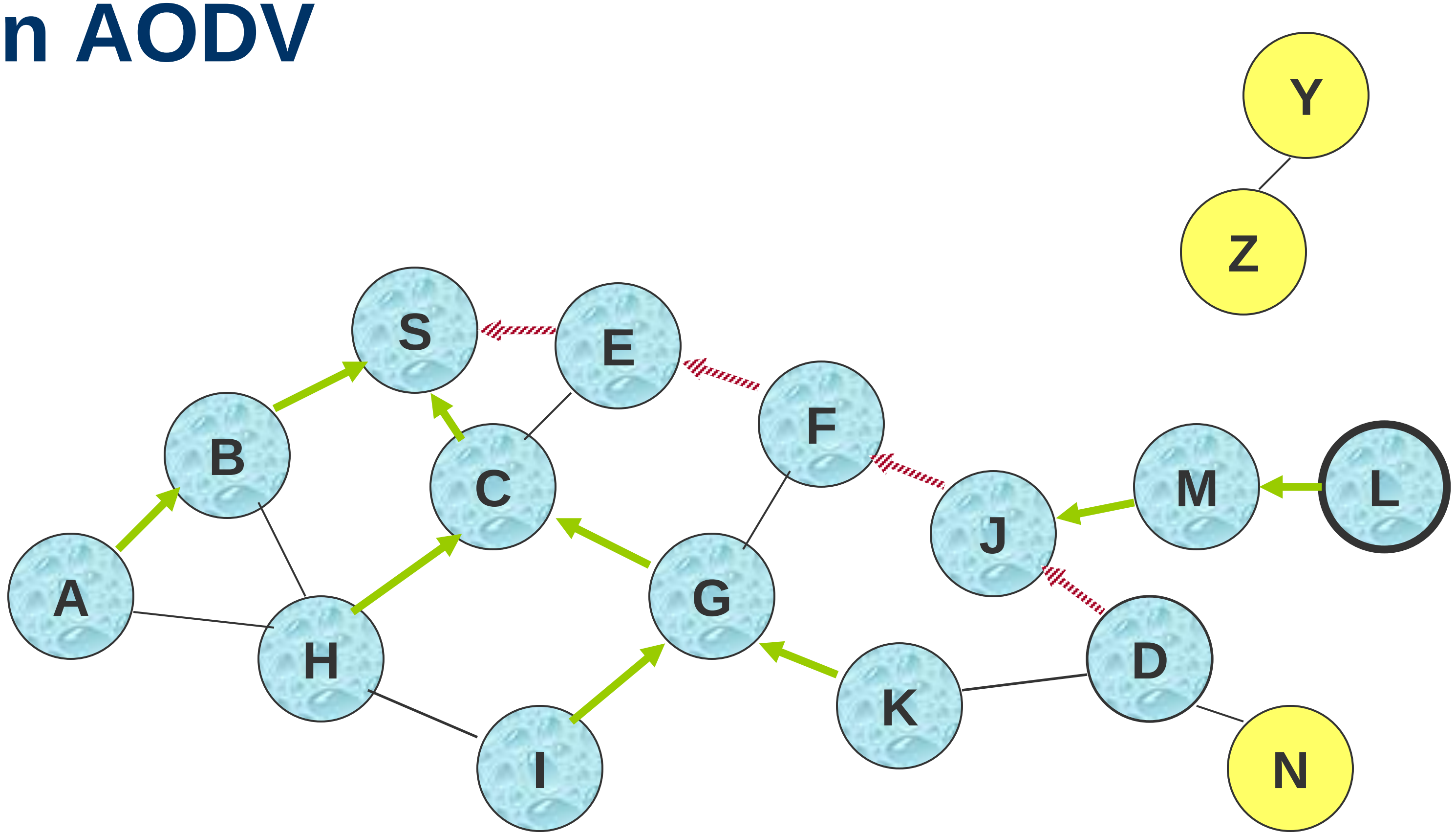


Reverse Path Setup in AODV



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

Route Reply in AODV



⇠ Represents links on path taken by RREP

Route Reply in AODV

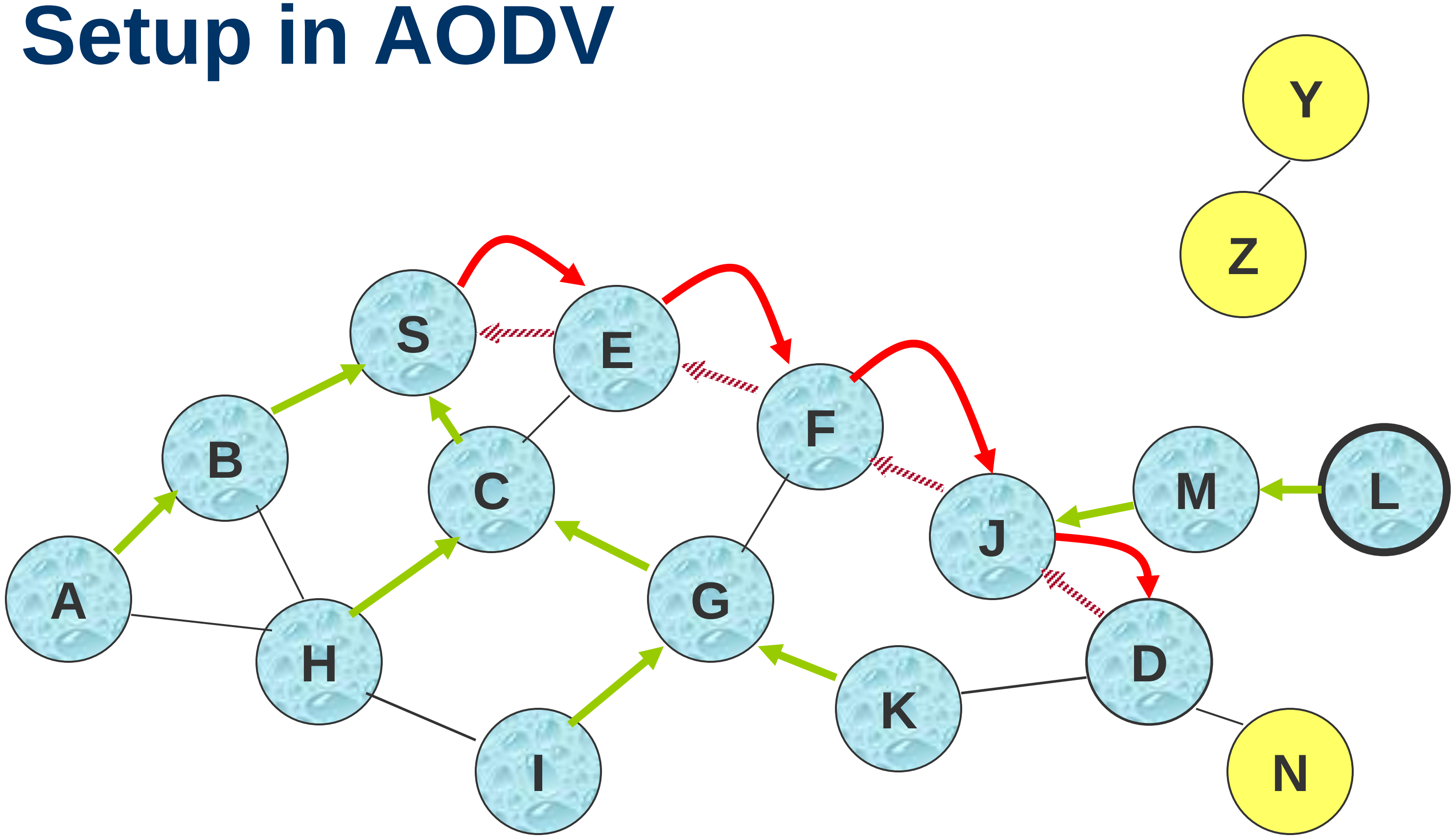
An intermediate node (not the destination) may also send a Route Reply (RREP) provided that it knows a more recent path than the one previously known to sender S

To determine whether the path known to an intermediate node is more recent, *destination sequence numbers* are used

The likelihood that an intermediate node will send a Route Reply when using AODV is not as high as DSR

- A new Route Request by node S for a destination is assigned a higher destination sequence number. An intermediate node, which knows a route, but with a smaller sequence number, cannot send Route Reply

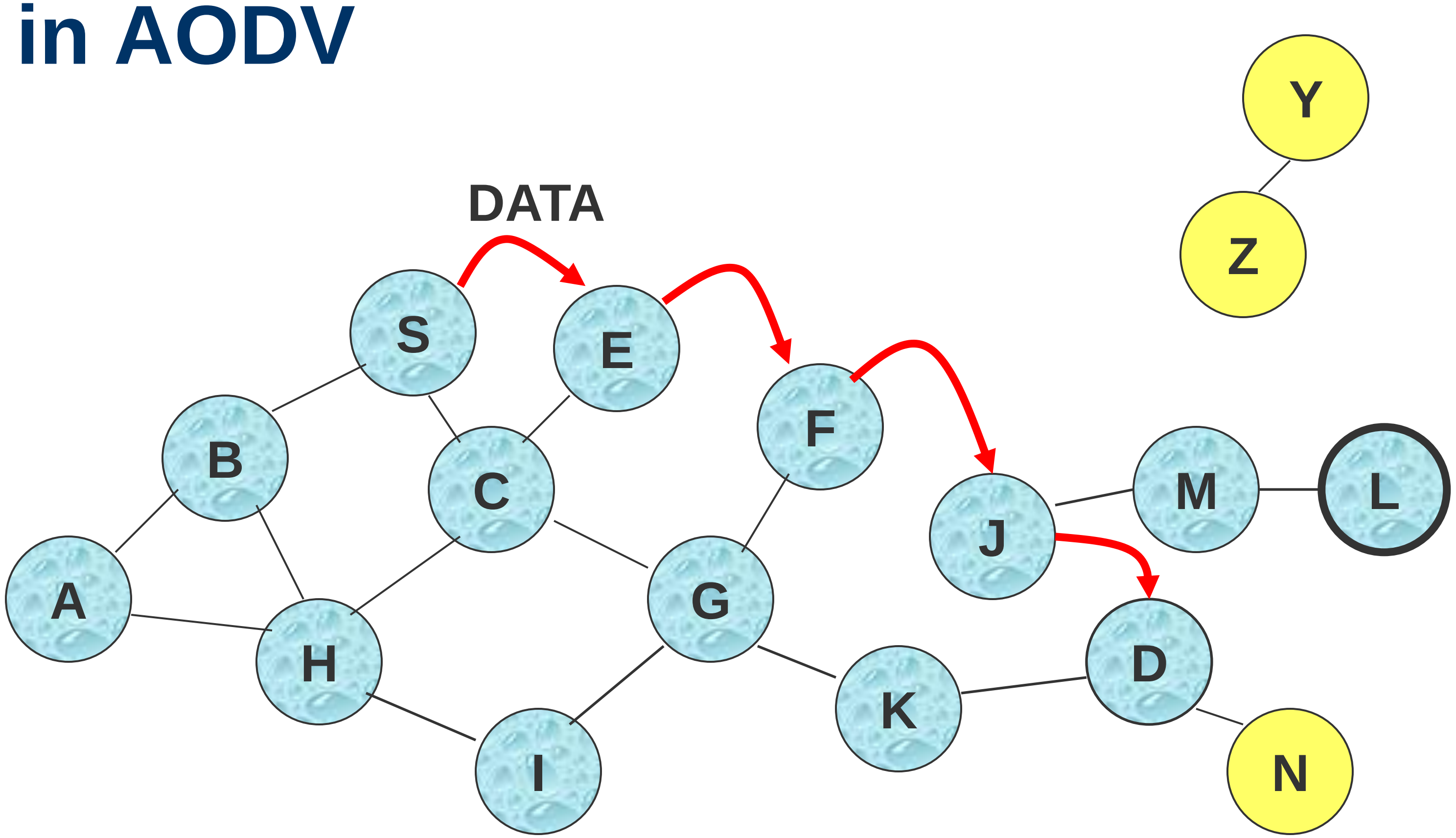
Forward Path Setup in AODV



Forward links are setup when RREP travels along the reverse path

↷ Represents a link on the forward path

Data Delivery in AODV



Routing table entries used to forward data packet.
Route is *not* included in packet header.

Summary: AODV

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

- Multi-path extensions can be designed
- DSR may maintain several routes for a single destination

Unused routes expire even if topology does not change

Link State Routing [Huitema95]

Each node periodically floods status of its links

Each node re-broadcasts link state information received from its neighbor

Each node keeps track of link state information received from other nodes

Each node uses above information to determine next hop to each destination

Optimized Link State Routing (OLSR)

The overhead of flooding link state information is reduced by requiring fewer nodes to forward the information

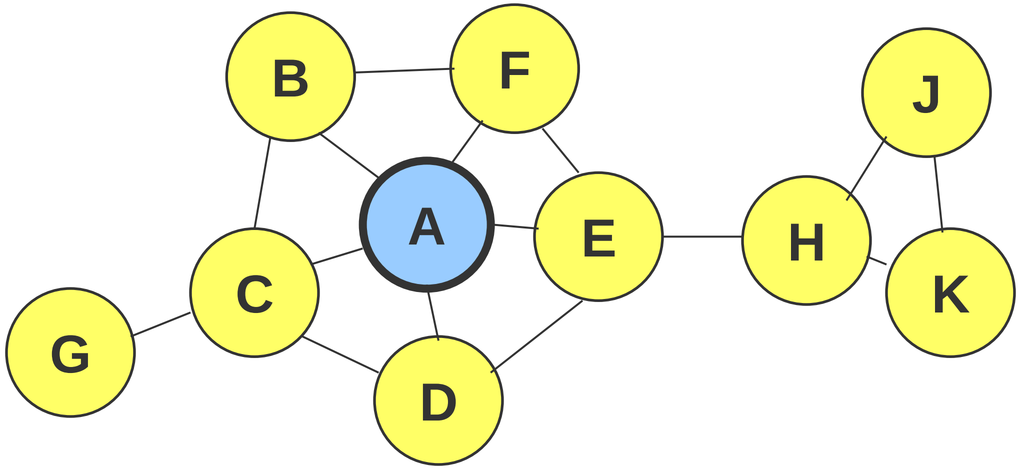
A broadcast from node X is only forwarded by its *multipoint relays*

Multipoint relays of node X are its neighbors such that each two-hop neighbor of X is a one-hop neighbor of at least one multipoint relay of X

- Each node transmits its neighbor list in periodic beacons, so that all nodes can know their 2-hop neighbors, in order to choose the multipoint relays

Optimized Link State Routing (OLSR)

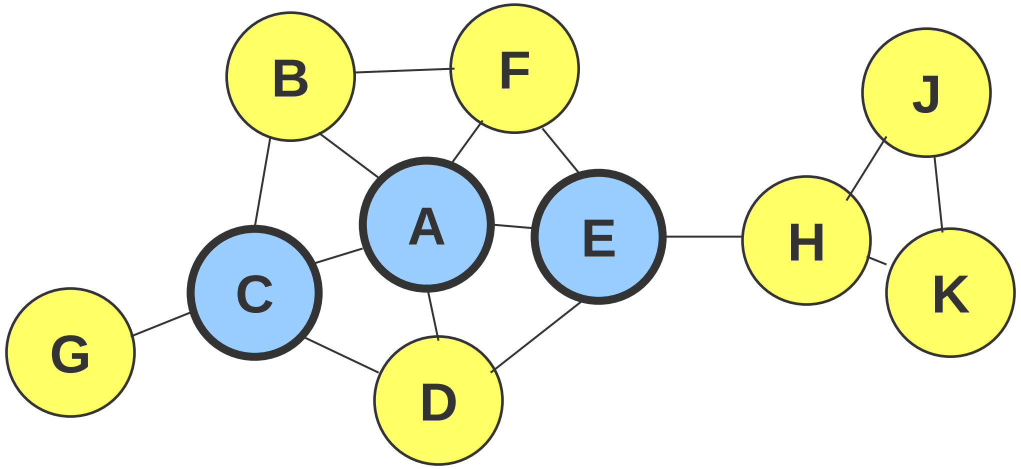
Nodes C and E are multipoint relays of node A



● Node that has broadcast state information from A

Optimized Link State Routing (OLSR)

Nodes C and E forward information received from A

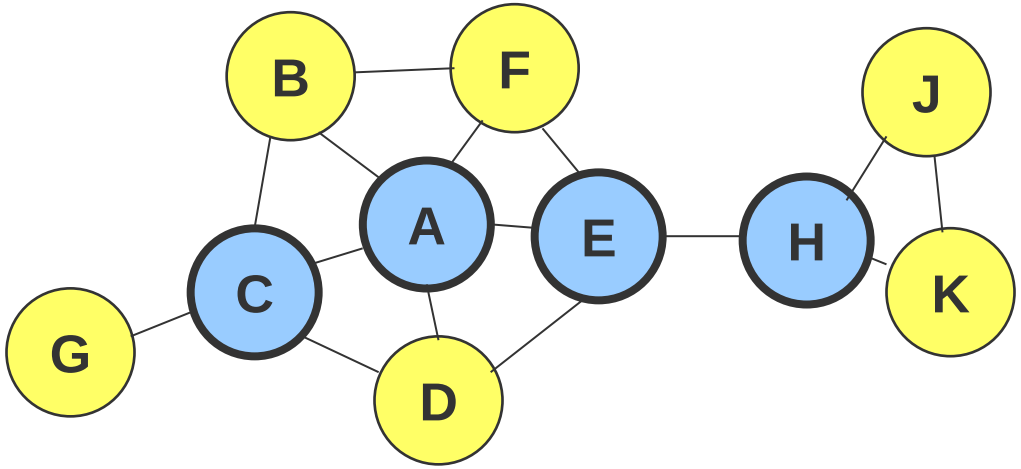


● Node that has broadcast state information from A

Optimized Link State Routing (OLSR)

Only node E is a multipoint relay for node H

E has already forwarded the same information once



● Node that has broadcast state information from A

Summary: OLSR

OLSR floods information through the multipoint relays

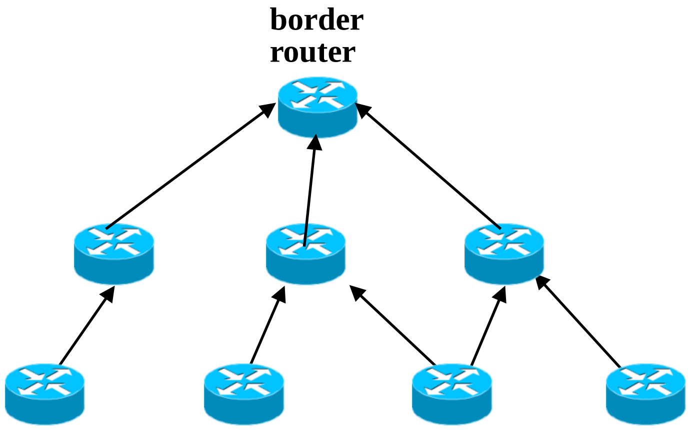
The flooded information itself is for links connecting nodes to respective multipoint relays

Nodes need to calculate routes (shortest path trees) based on link-state knowledge, typically using the Dijkstra algorithm

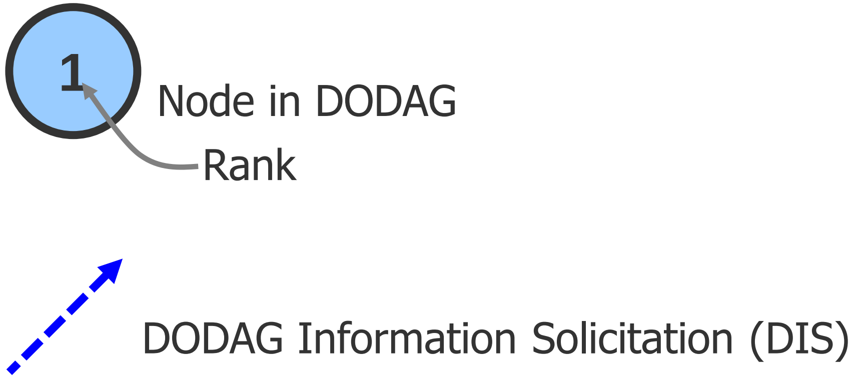
Routes used by OLSR only include multipoint relays as intermediate nodes

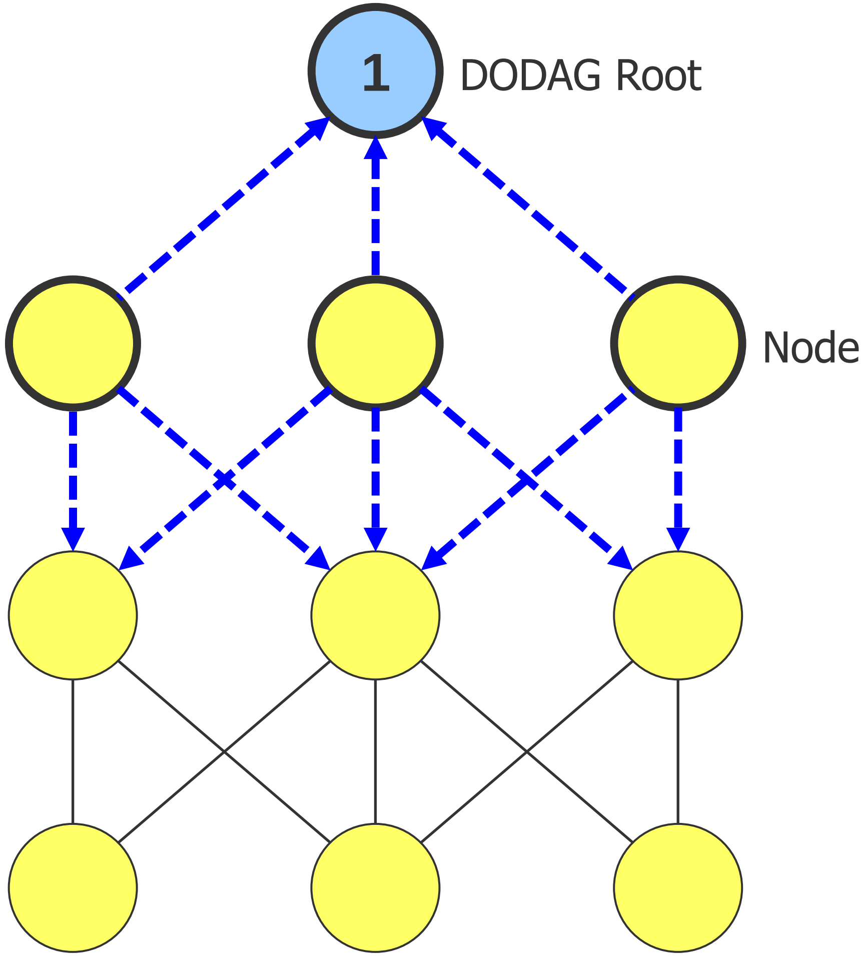
RPL - Routing Protocol for Low Power and Lossy Networks (LLN) – RFC 6550

- Optimized for low-energy networks (without mobility)
- Destination Oriented Directed Acyclic Graph (DODAG)
- Routing state propagation
  - Conventional:
    - Link-state: scoped flooding
    - Distance-vector: periodic routing beacons
  - Trickle (RFC 6202):
    - adaptive exchange rate
- Spatial diversity
  - A router maintains multiple potential parents
- Expressive link metrics
  - ETX: Estimated Number of Transmissions

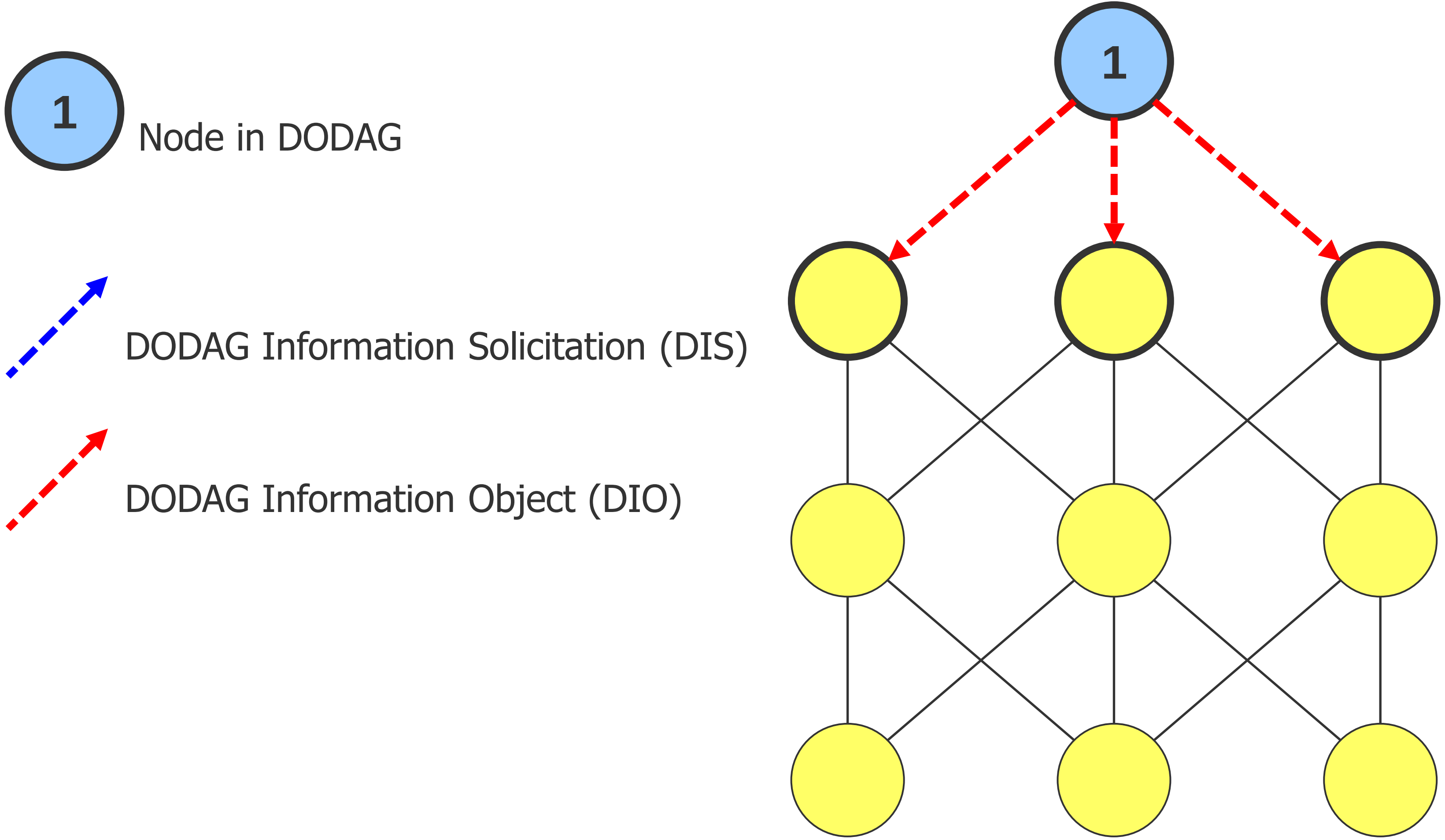


RPL Topology Creation - Upward





RPL Topology Creation - Upward

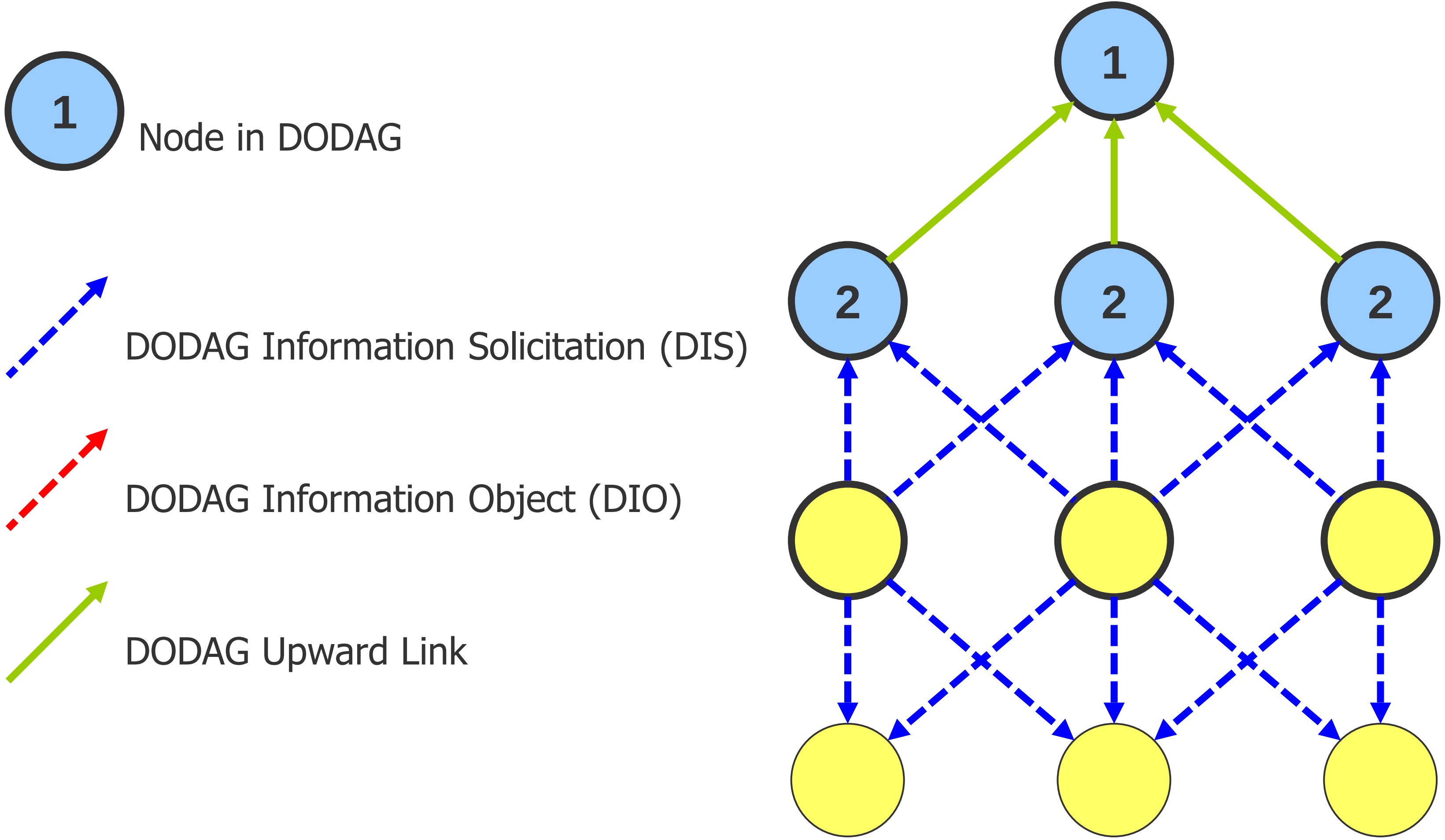


Node in DODAG

DODAG Information Solicitation (DIS)

DODAG Information Object (DIO)

RPL Topology Creation - Upward



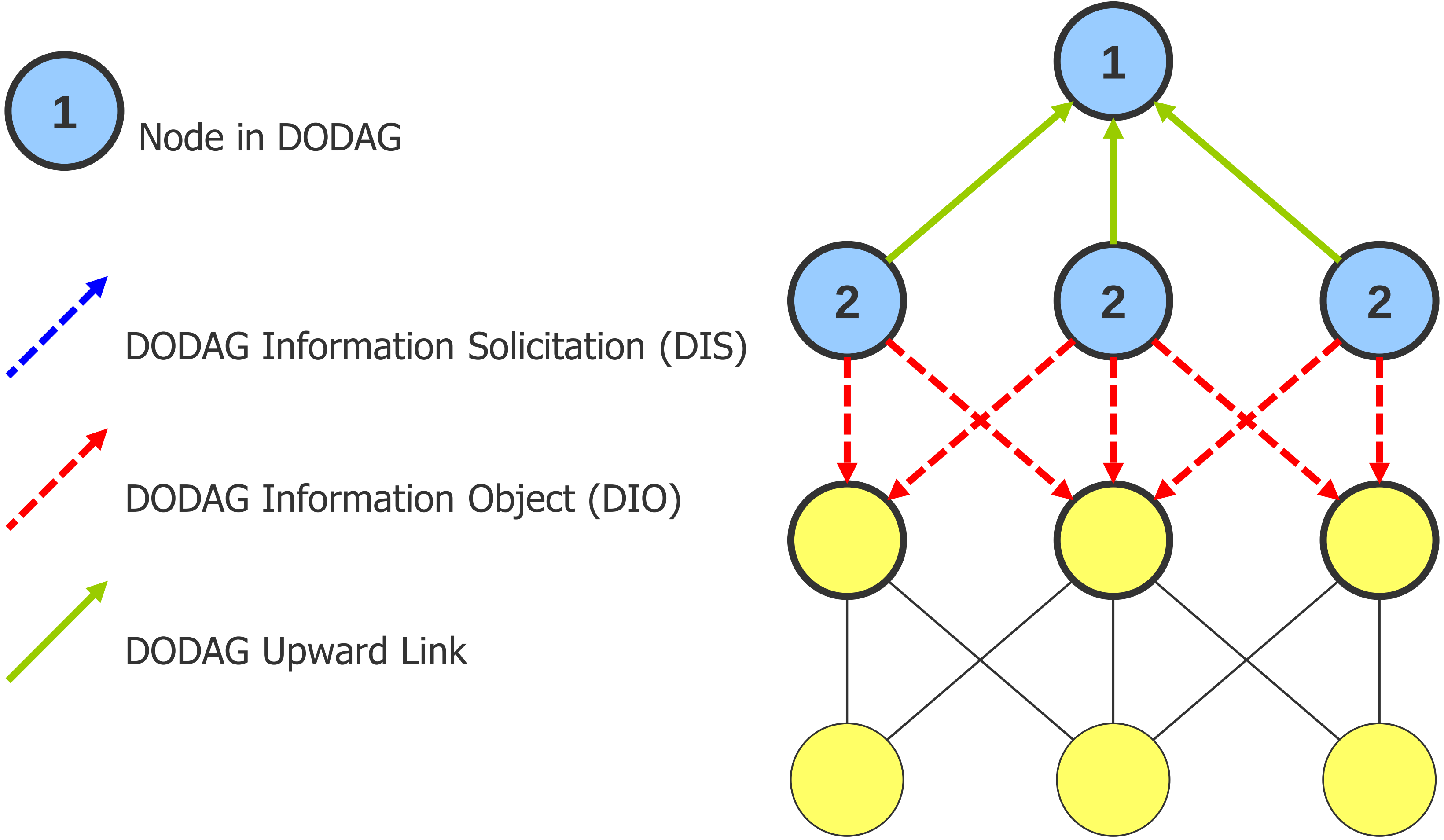
Node in DODAG

DODAG Information Solicitation (DIS)

DODAG Information Object (DIO)

DODAG Upward Link

RPL Topology Creation - Upward



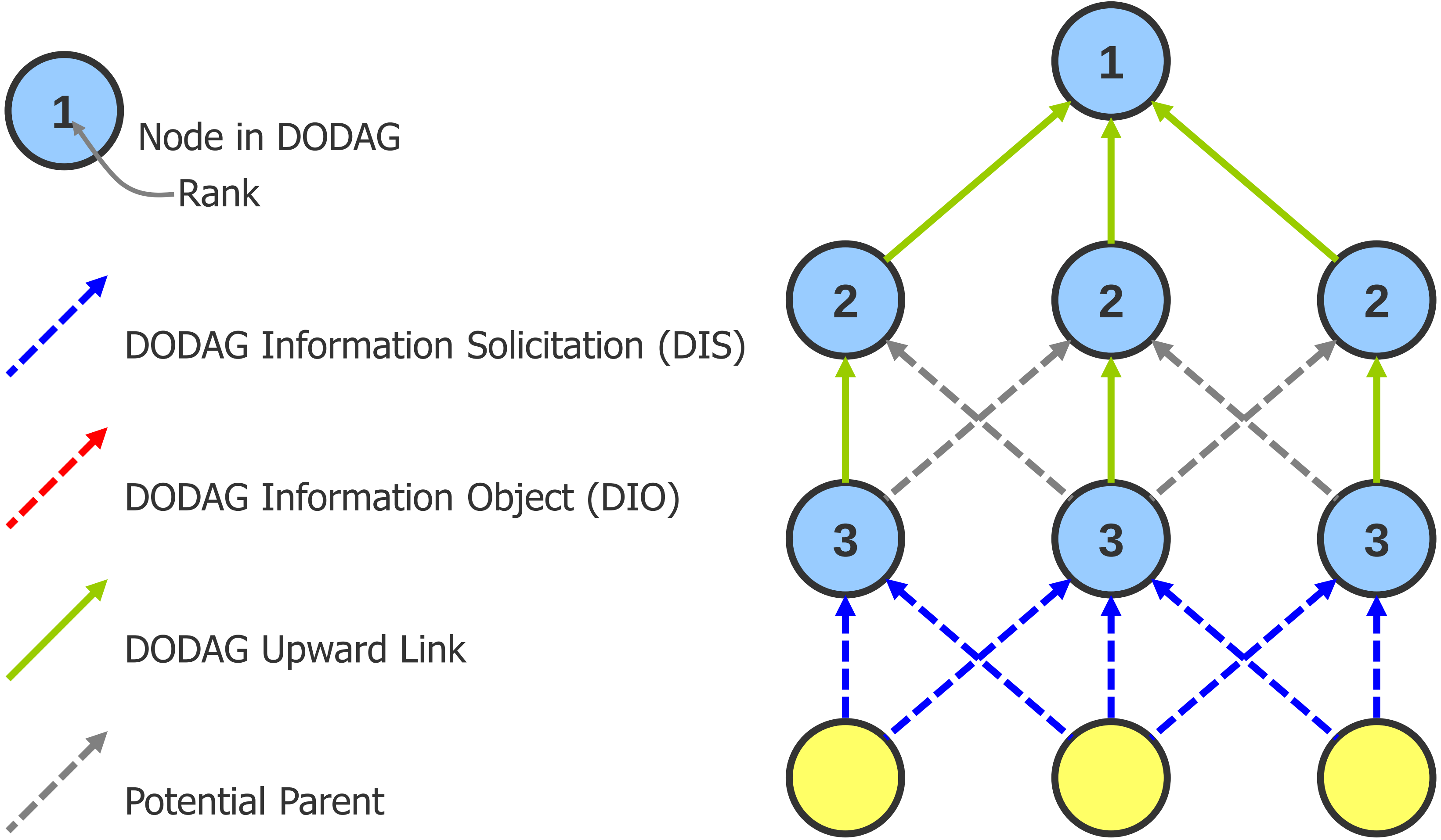
Node in DODAG

DODAG Information Solicitation (DIS)

DODAG Information Object (DIO)

DODAG Upward Link

RPL Topology Creation - Upward



Node in DODAG

Rank

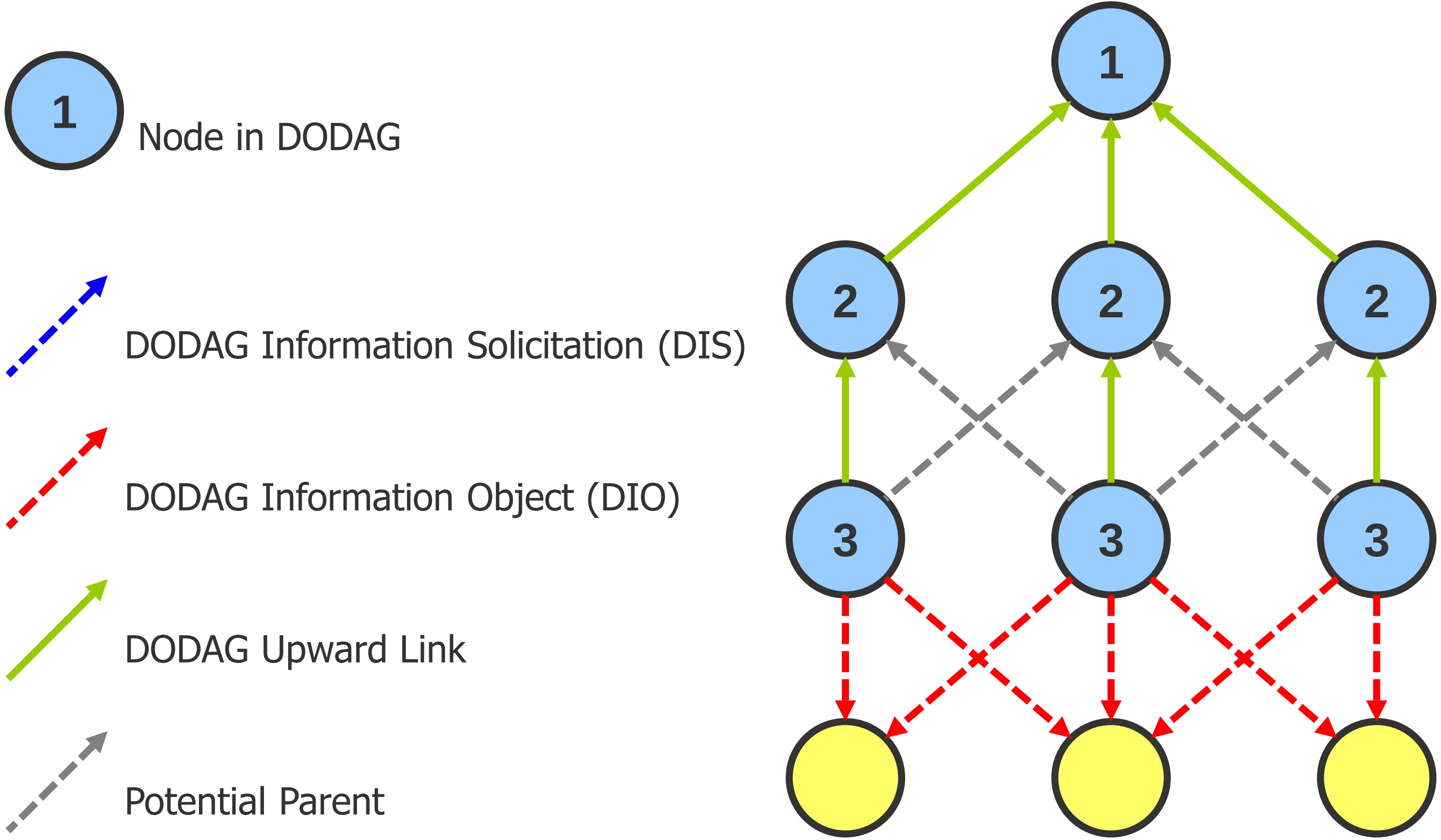
DODAG Information Solicitation (DIS)

DODAG Information Object (DIO)

DODAG Upward Link

Potential Parent

RPL Topology Creation - Upward



Node in DODAG

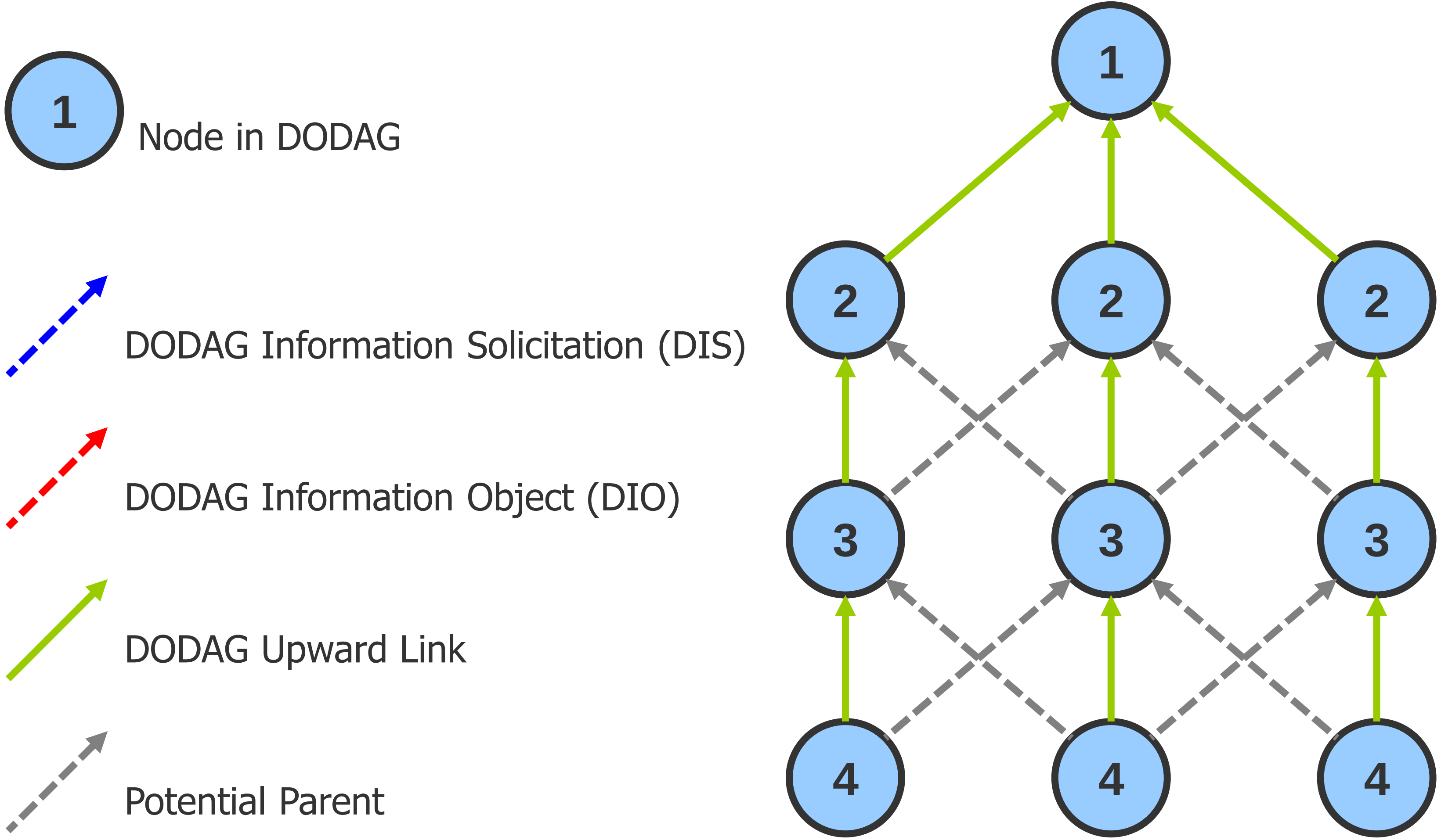
DODAG Information Solicitation (DIS)

DODAG Information Object (DIO)

DODAG Upward Link

Potential Parent

RPL Topology Creation - Upward



Node in DODAG

DODAG Information Solicitation (DIS)

DODAG Information Object (DIO)

DODAG Upward Link

Potential Parent

RPL Topology

Downward routes created analogously

Two routing modes

- Non-storing: without local routing tables
  - Local routing: Uptree (default) to root
  - Source routes issued at root
- Storing: with local routing tables
  - Local routing decisions forward directly into subtrees

Topology maintenance: New DAG version created on request

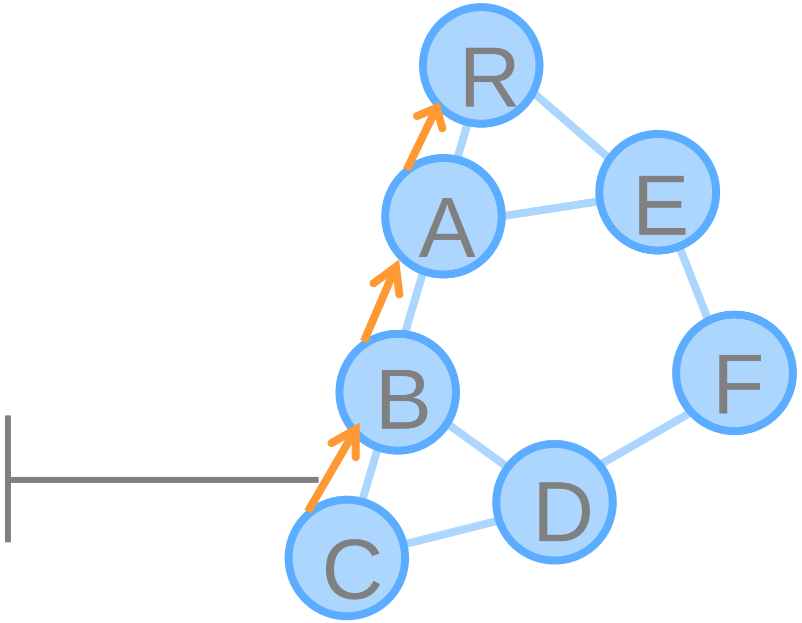
Creating Downward Routes
(Point-to-Multipoint, Point-to-Point)

Nodes announce DAOs (Destination Advertisement Objects)

Non-storing Mode

Node informs root about its parents via DAOs

Root builds Downward Routing Table



C tells R its address and B

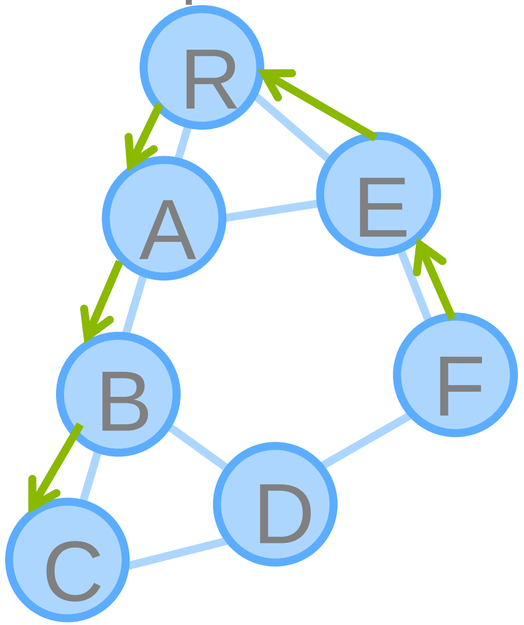
Creating Downward Routes
(Point-to-Multipoint, Point-to-Point)

Nodes announce DAOs (Destination Advertisement Objects)

Non-storing Mode

- Node informs root about its parents via DAOs
- Root builds Downward Routing Table
- Nodes send data via Default Routes to the root
- Root sends data via Source Routing to destination

```
C: A->B->C
B: A->B
A: A
…
```



Creating Downward Routes
(Point-to-Multipoint, Point-to-Point)

Nodes announce DAOs (Destination Advertisement Objects)

Non-storing Mode

Node informs root about its parents via DAOs

Root builds Downward Routing Table

Nodes send data via Default Routes to the root

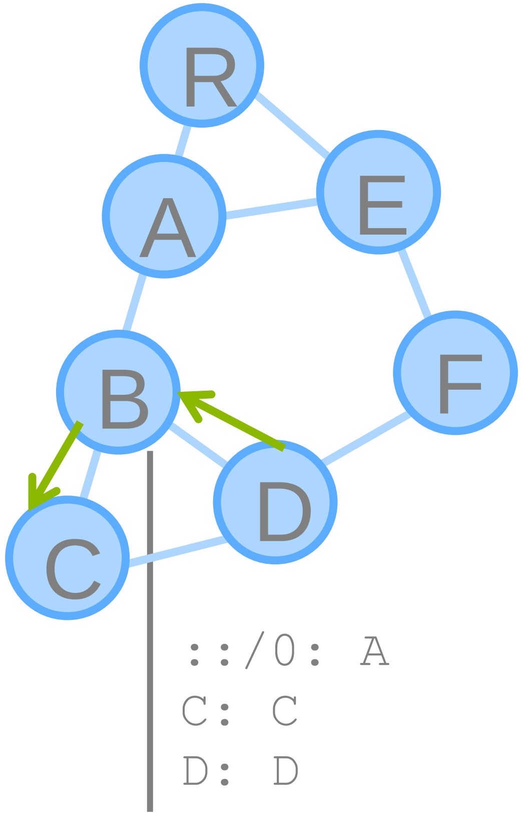
Root sends data via Source Routing to destination

Storing Mode

Node uses DAOs to inform parents about addresses

Each parent learns all addresses of the Sub-DODAG

Packets within the Sub-DODAGs arrive directly



```
::/0: A
C: C
D: D
```

Further Routing Approaches

Improvements & Optimisations of Previous Protocols

Location Aided Routing

Clustering after Landmarking

Hybrid Routing

Hierarchic / Anchored Routing

Power-Aware Routing

…

Bibliography

- Internet Society: *The Internet of Things: An Overview,* White Paper, Oct. 2015
  http://www.internetsociety.org/doc/iot-overview
- Zach Shelby, Carsten Bormann: *6LoWPAN: The Wireless Embedded Internet,* Wiley & Sons, 2009.
- C. Murthy and B. Manoj: *Ad Hoc Wireless Networks*, Pearson Prentice Hall, 2004.
- Charles Perkins: *Ad Hoc Networking*, Addison-Wesley, 2001.
- S. Sarkar, T. Basavaraju, C. Puttamadappa: *Ad Hoc Mobile Wireless Networks*, Auerbach Publications, 2008.
- Nitin H. Vaidya: *Mobile Ad Hoc Networks,* Tutorial at InfoCom 2006*,* http://www.crhc.uiuc.edu/wireless/talks/2006.Infocom.ppt.
- P. Gupta and P. R. Kumar, “The capacity of wireless networks,” IEEE Transactions on Information Theory, vol. 46, no. 2, pp. 388–404, 2000.
- Drafts, RFCs: tools.ietf.org, http://www.rfc-editor.org