Routing in Ad-hoc networks

PRESENTED BY-

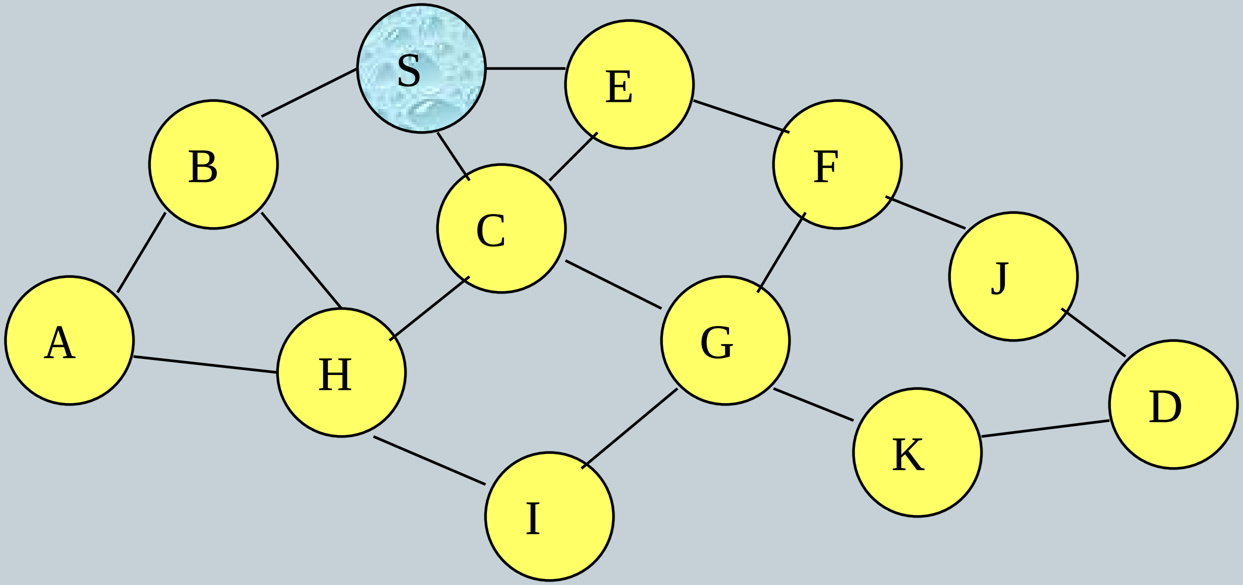
LEWIS TSENG

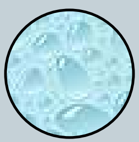
RACHIT AGARWAL

Ad-hoc networks

- Infrastructure-less networks
  - No fixed routers
  - (potentially) mobile nodes
  - Dynamically and arbitrarily located
- Desired routing requirements
  - High connectivity
  - Low overhead (how to characterize overhead?)

Flooding at the Data-plane



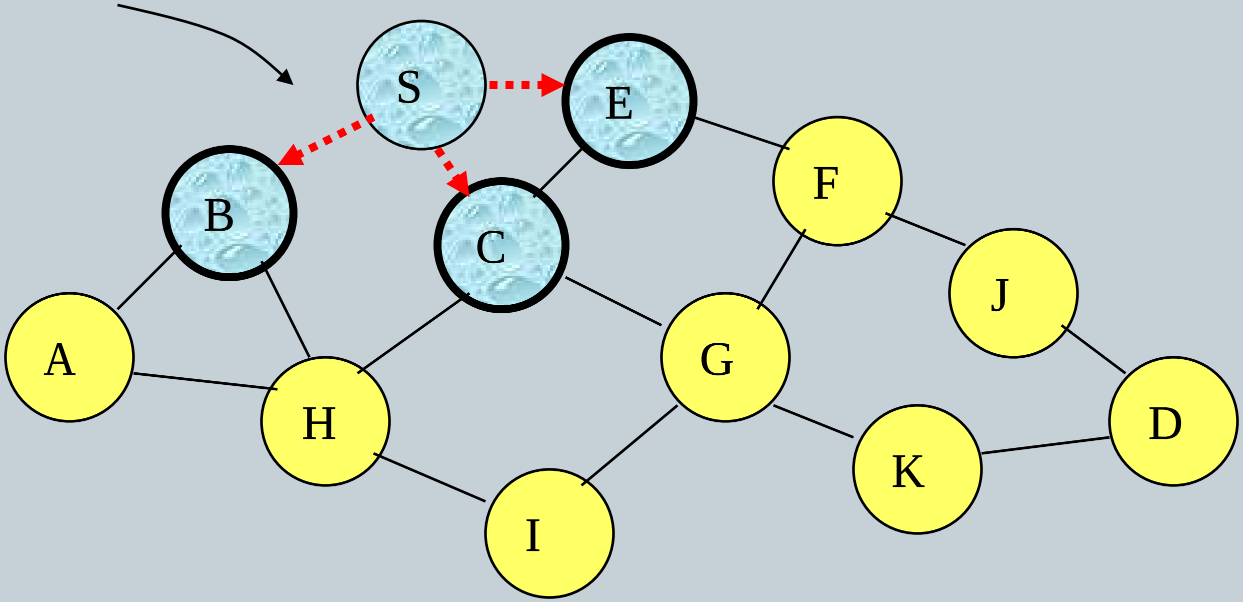


Represents a node that has received packet P

Represents that connected nodes are within each other's transmission range

Flooding at the Data-plane

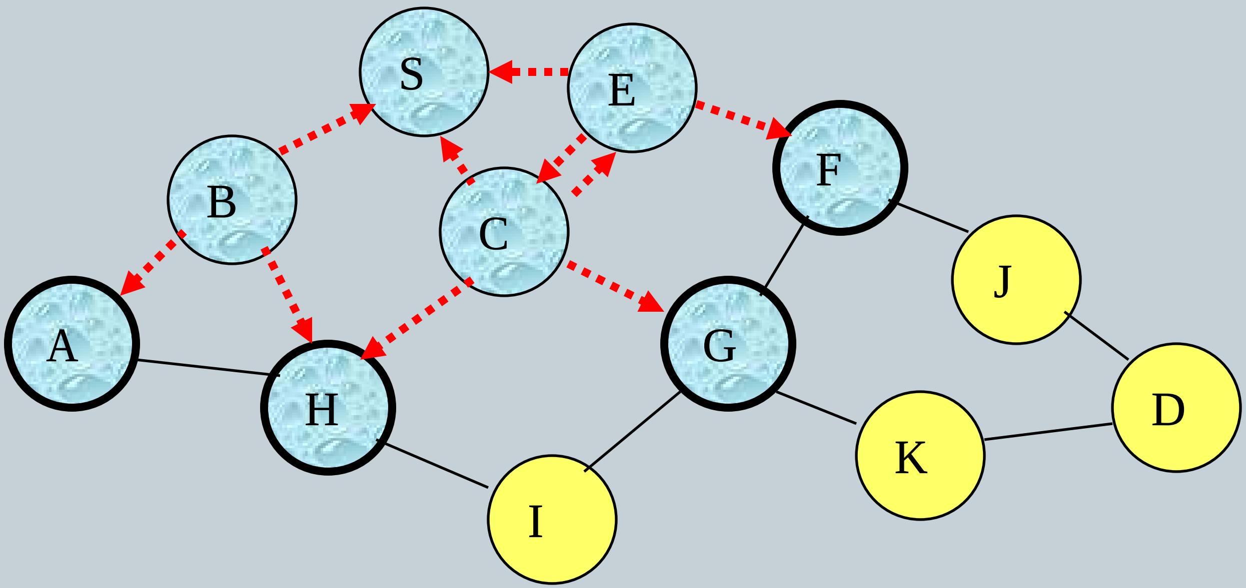
Broadcast transmission



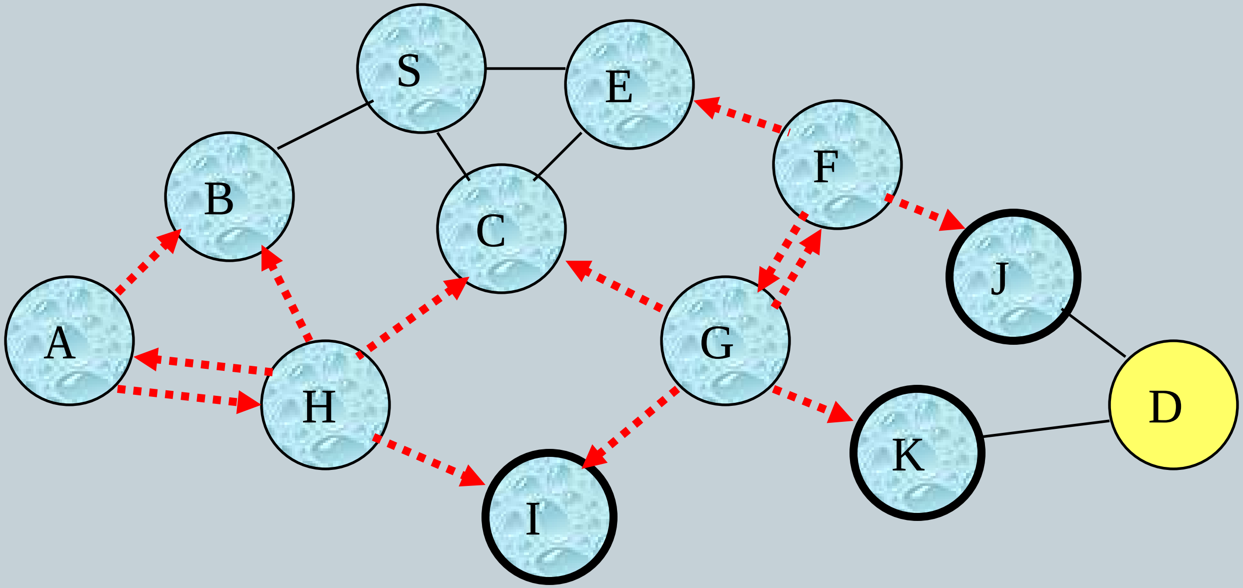
Represents a node that receives packet P for the first time

Represents transmission of packet P

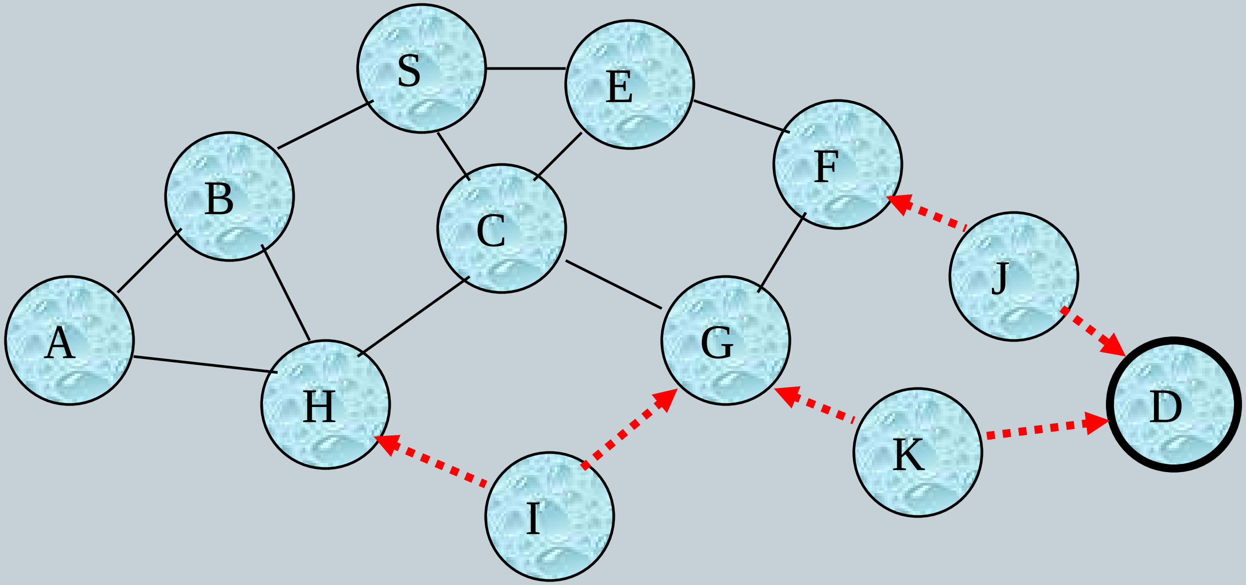
Flooding at the Data-plane



Flooding at the Data-plane



Flooding at the Data-plane



- 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
- Welcome to the world of wireless networks

Advantages of flooding at the data-plane

- Simplicity
- Potentially higher reliability of data delivery
- No routing tables – just need to store neighbors

Disadvantages of flooding at the data-plane

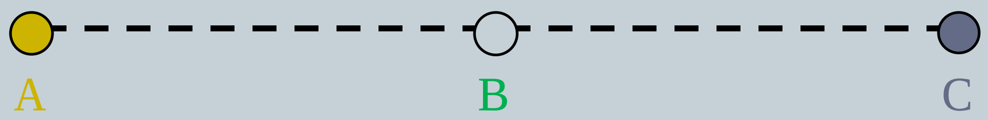
- Potentially, very high overhead

- Potentially lower reliability of data delivery
  - hard to implement reliable broadcast
  - Packet collisions

Destination-Sequenced Distance-Vector (DSDV)

- Routing tables:
  - Each node stores, for each destination:
    - next-hop
    - cost
    - sequence number

- Control plane:
  - periodically broadcast routing tables to neighbors



A:

Dest.	Next	Metric	Seq.
A	A	0	A-550
B	B	1	B-104
C	B	2	C-590

B:

Dest.	Next	Metric	Seq.
A	A	1	A-550
B	B	0	B-104
C	C	1	C-590

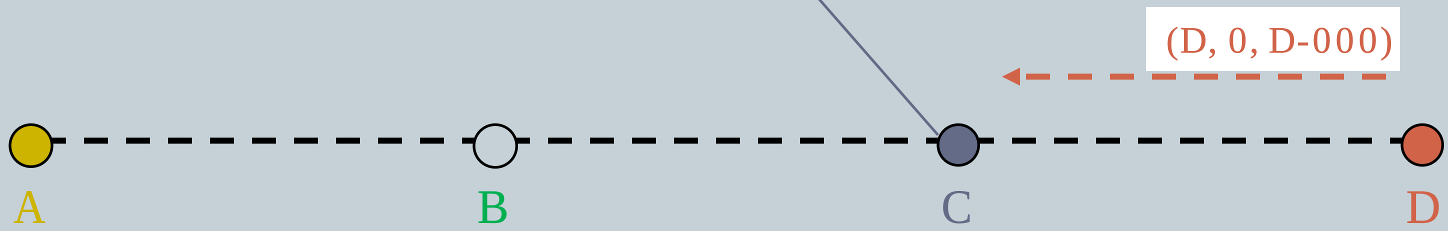
C:

Dest.	Next	Metric	Seq.
A	B	2	A-550
B	B	1	B-104
C	C	0	C-590

DSDV Routing tables

2. Insert entry for D with sequence number D-000
3. Immediately broadcast own table

1. D broadcast for first time – sends sequence number D-000



Dest.	Next	Metric	Seq.
A	A	0	A-550
B	B	1	B-104
C	B	2	C-590

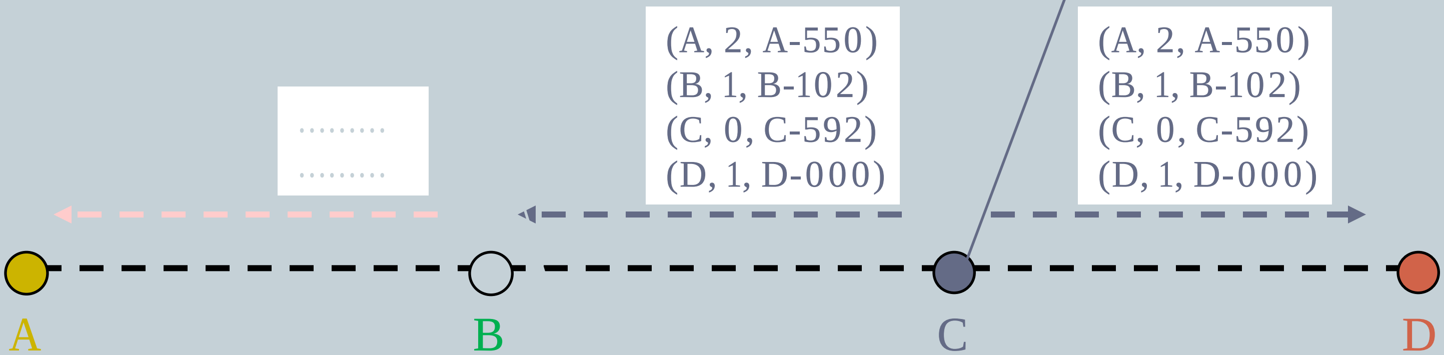
Dest.	Next	Metric	Seq.
A	A	1	A-550
B	B	0	B-104
C	C	1	C-590

Dest.	Next	Metric	Seq.
A	B	2	A-550
B	B	1	B-104
C	C	0	C-590
D	D	1	D-000

DSDV Routing Tables

3. C increases its sequence number to C-592 and broadcasts its new table.

4. B gets this new information and updates its table.......



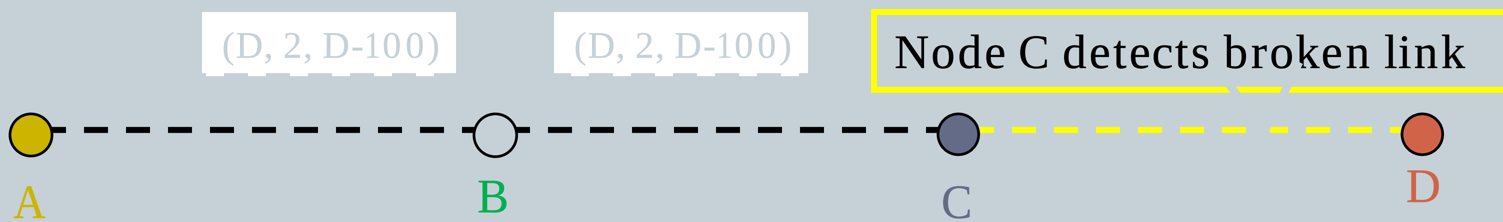
Dest.	Next	Metric	Seq.
A	A	0	A-550
B	B	1	B-104
C	B	2	C-590

Dest.	Next	Metric	Seq.
A	A	1	A-550
B	B	0	B-102
C	C	1	C-592
D	C	2	D-000

Dest.	Next	Metric	Seq.
A	B	2	A-550
B	B	1	B-102
C	C	0	C-592
D	D	1	D-000

DSDV Link Failures

2. B does its broadcast – no affect on C (old sequence number)

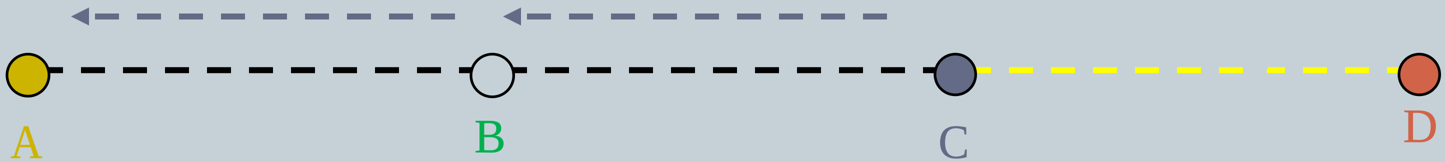


Dest.	Next	Metric	Seq.
...	...	...	
D	B	3	D-100

Dest.c	Next	Metric	Seq.
...	...	...	
D	C	2	D-100

Dest.	Next	Metric	Seq.
...	...	...	
D	D	∞	D-101

DSDV Link Failures



Dest.	Next	Metric	Seq.
...	...	...	...
D	B	3	D-100
D	B	∞	D-101

Dest.c	Next	Metric	Seq.
...	...	...	...
D	C	2	D-100
D	C	∞	D-101

Dest.	Next	Metric	Seq.
...	...	...	
D	D	1	D-100
D	D	∞	D-101

Advantages of flooding at control plane

- Overhead due to data plane flooding avoided
- Nodes maintain (almost) consistent network map
  - If the network is stable, loop-free routing very easy
  - Resulting paths are shortest paths

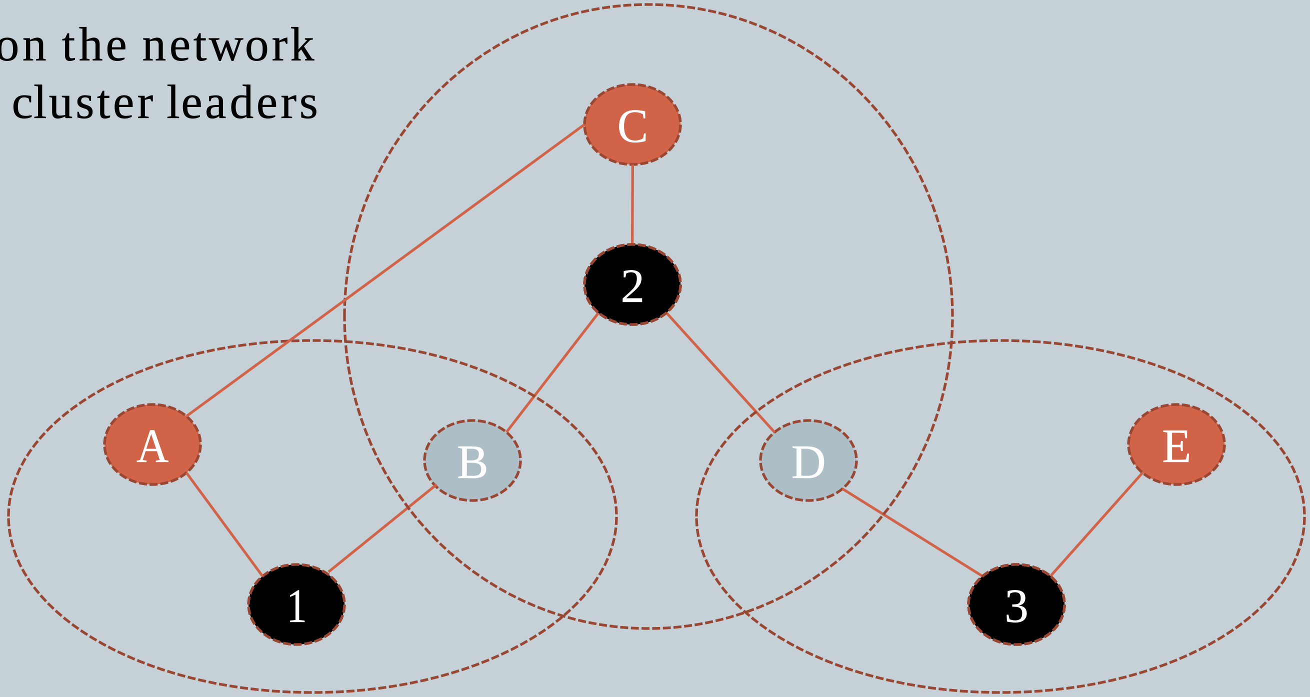
Disadvantages of flooding at control plane

- Scalability
  - does not scale to large networks
  - Even for small networks, large overhead if network is dynamic

- #Data packets versus #control packets?

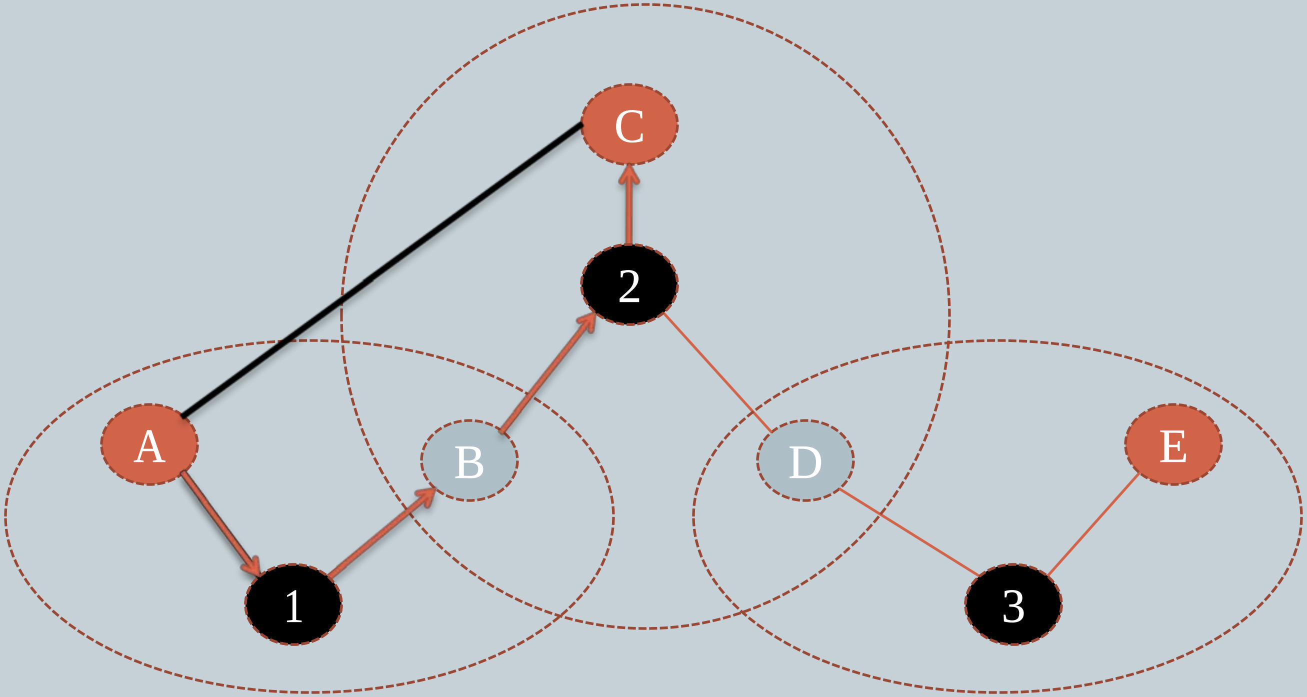
Clusterhead Gateway Switch Routing (CGSR)

1. Partition the network
2. Assign cluster leaders



- Flood the control plane within a cluster
- Flood the control plane among the cluster leaders

Clusterhead Gateway Switch Routing (CGSR)



Potentially longer paths

Advantages of CGSR

- Improved Scalability
  - Scales for large networks
  - Scales even for small, highly dynamic networks

- Failure reaction is more localized compared to DSDV

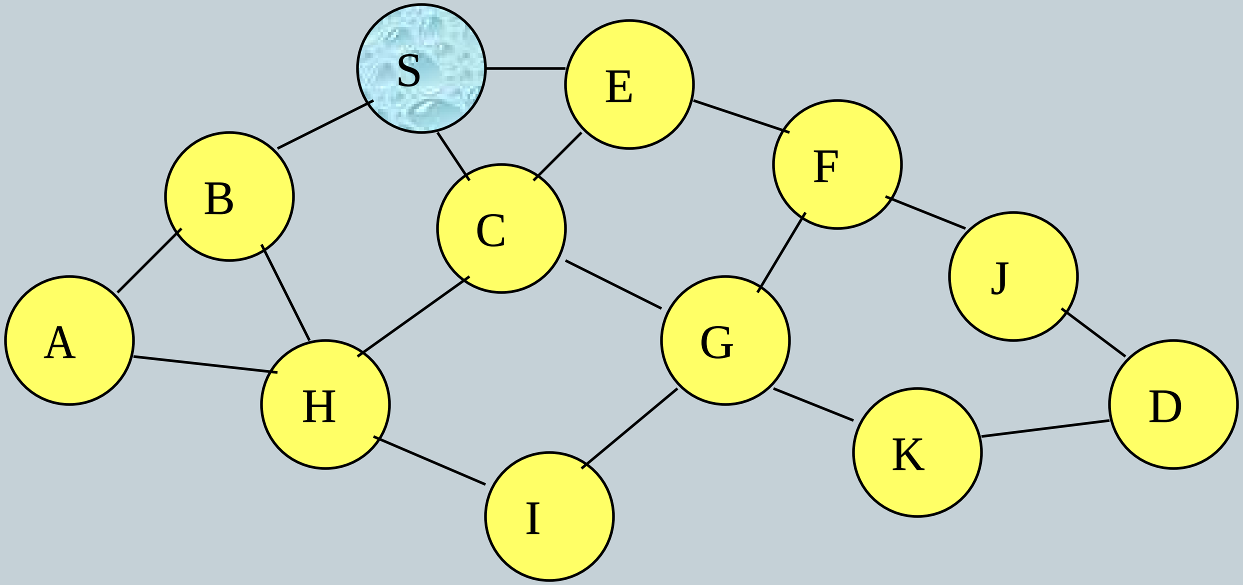
Disadvantages of CGSR

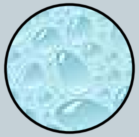
- Inflated Path lengths
  - May not route along shortest possible paths
  - (Price for improved scalability?)
- Failures adversely effect CGSR
- #Data packets versus #control packets?
  - If #data packets per unit time << 1 ?

Dynamic Source Routing (DSR)

- 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

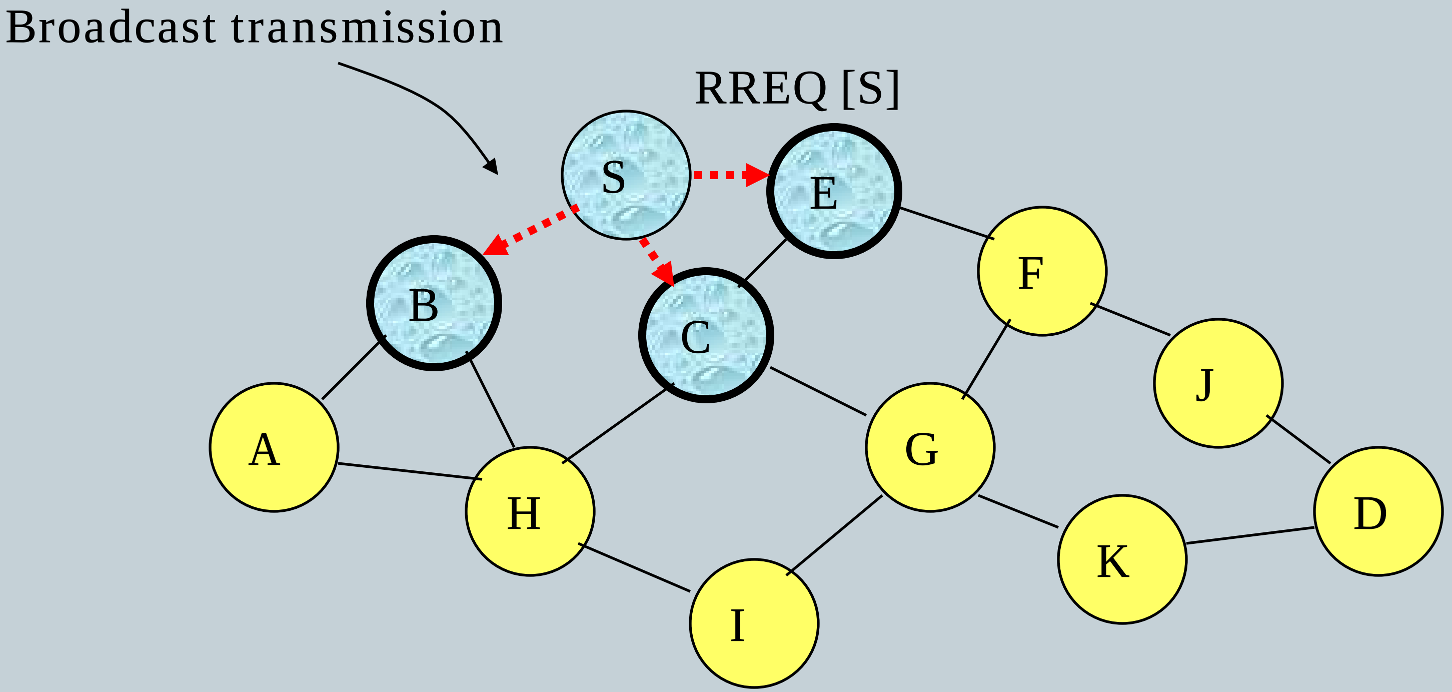
Route Discovery in DSR





Represents a node that has received RREQ for D from S

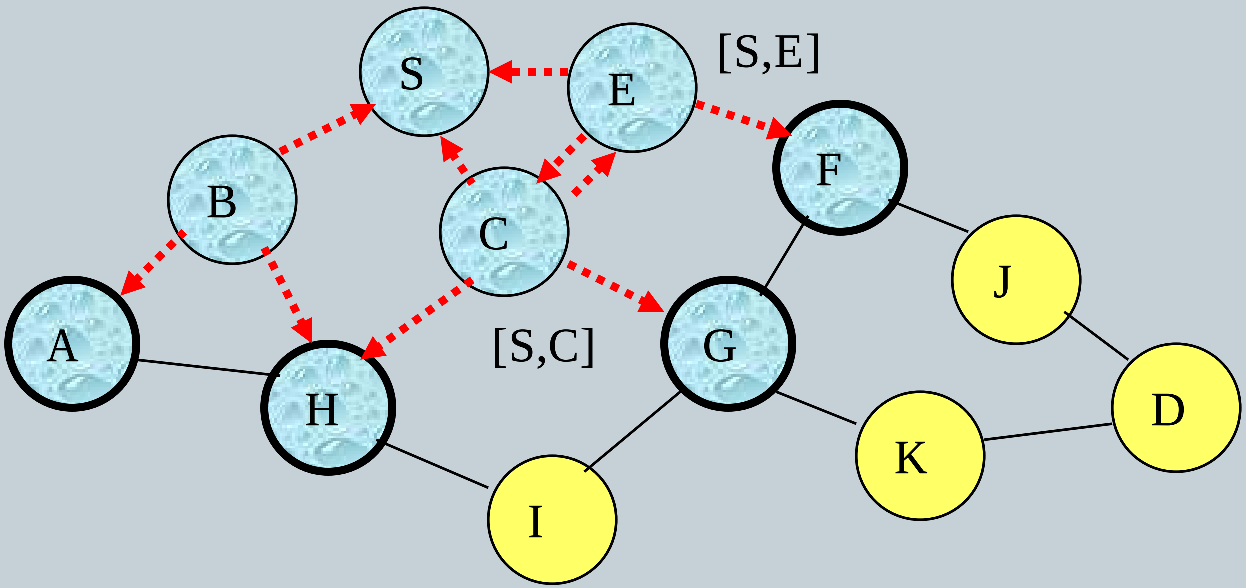
Route Discovery in DSR



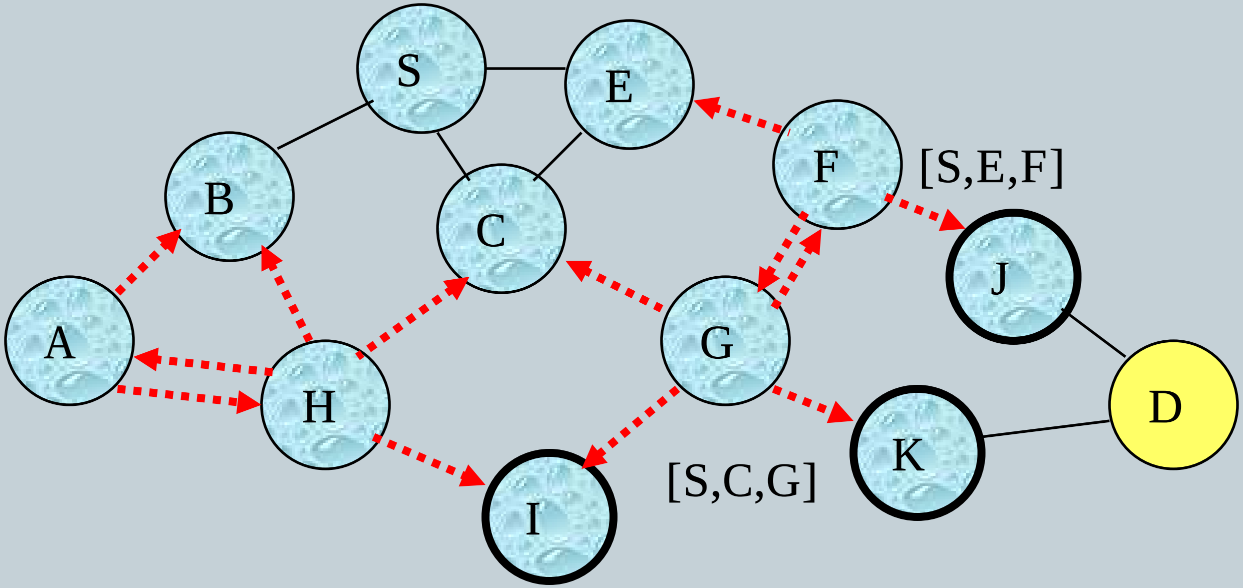
Represents transmission of RREQ

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

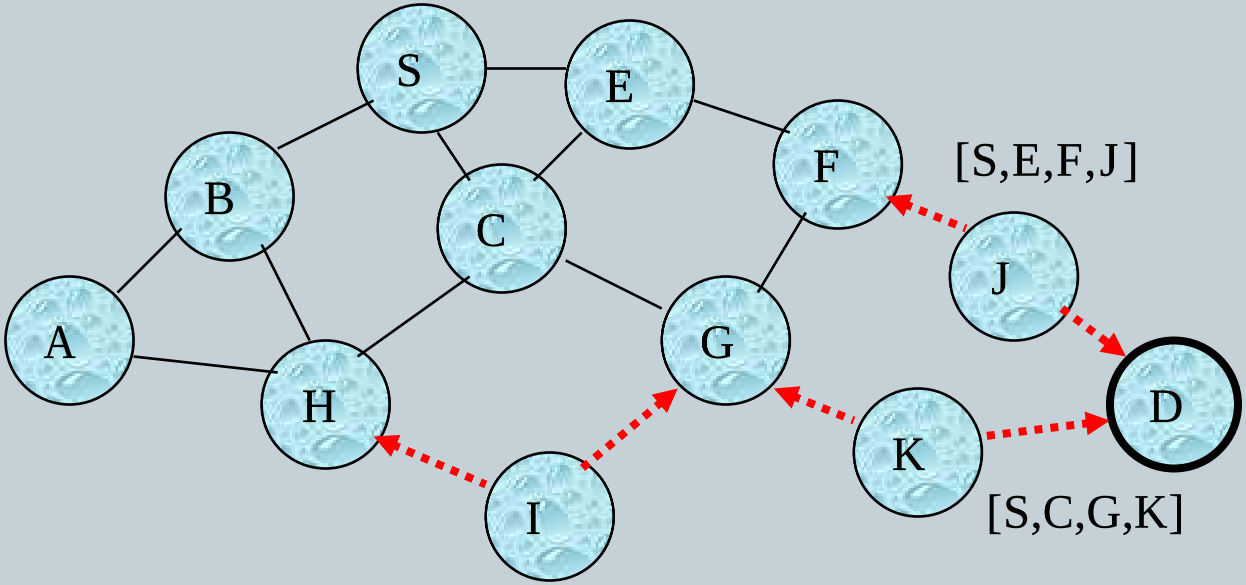
Route Discovery in DSR



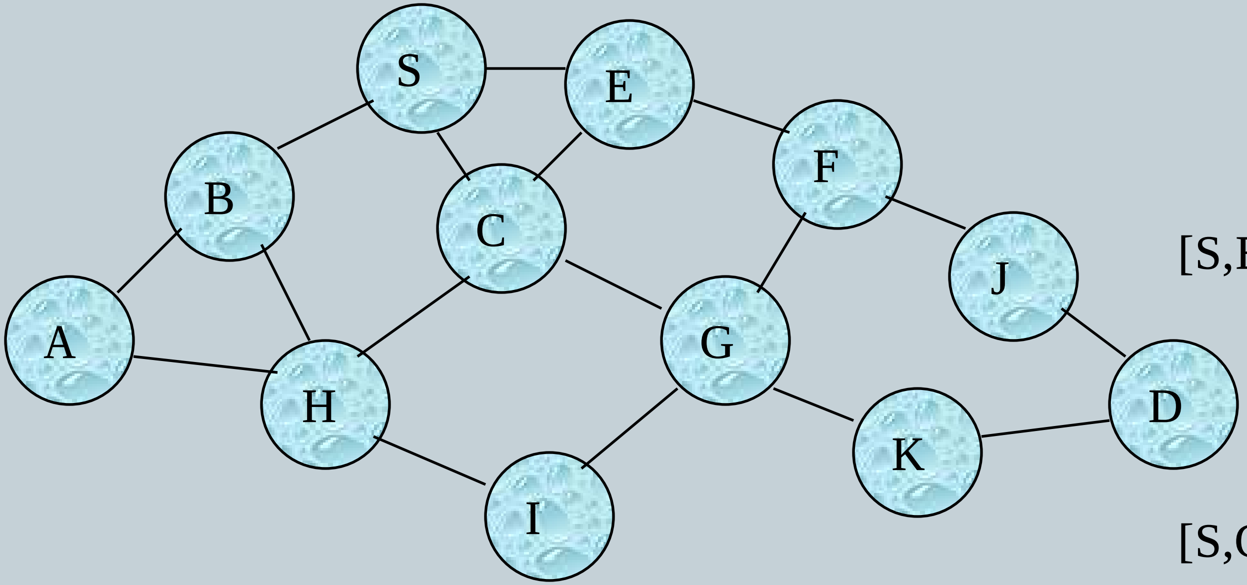
Route Discovery in DSR



Route Discovery in DSR



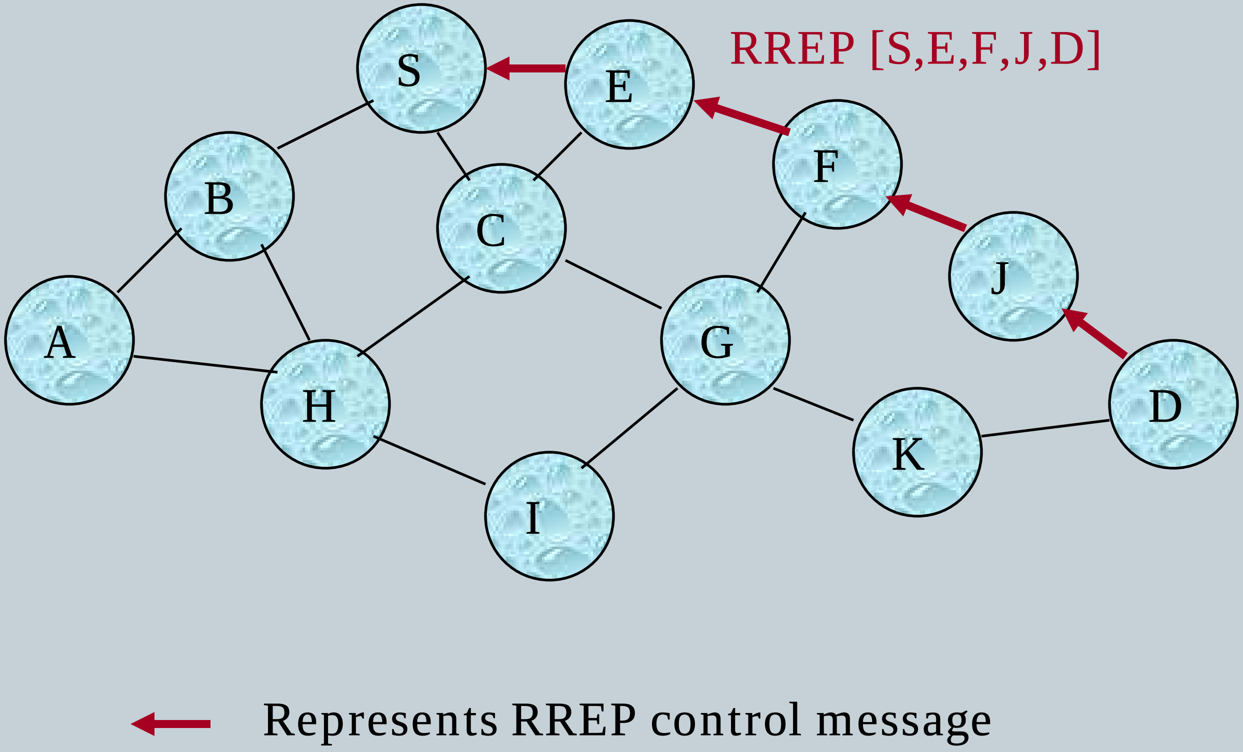
Route Discovery in DSR



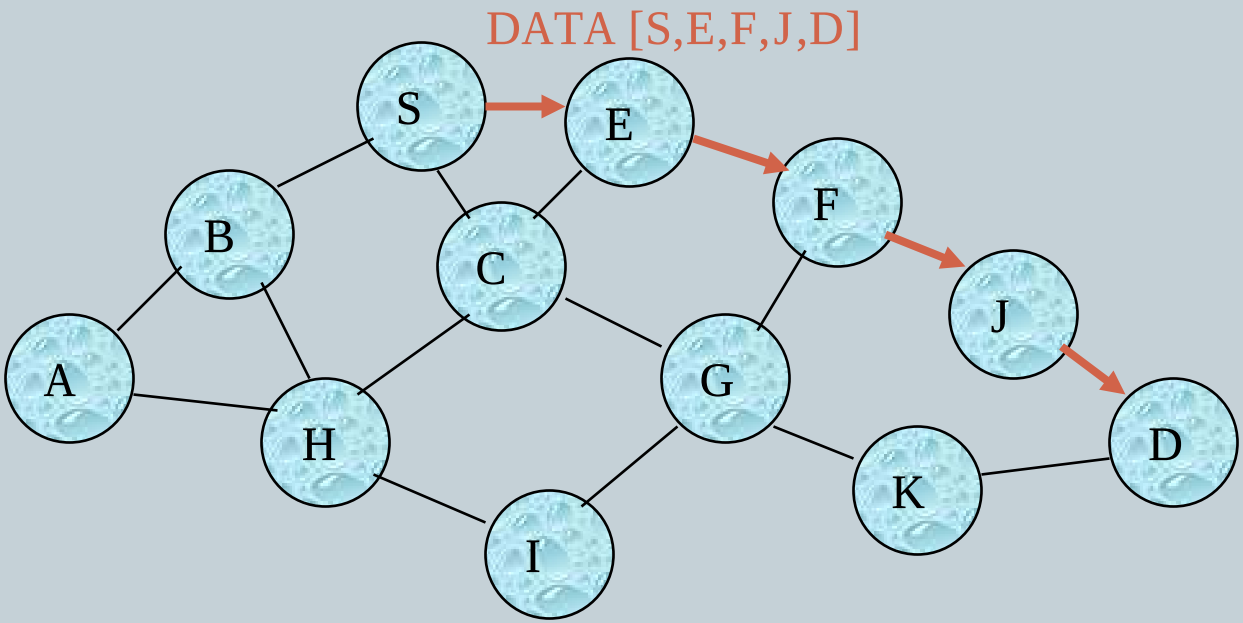
[S,E,F,J, D]

[S,C,G,K, D]

Route Reply in DSR



Data Delivery in DSR



- Packet header includes the entire route
- Intermediate nodes do a "packet header" look-up

Advantages of DSR

- Routes maintained only between nodes who need to communicate
  - reduces overhead of route maintenance

- Allows multi-path routing

- No routing tables

- Shortest, loop-free paths

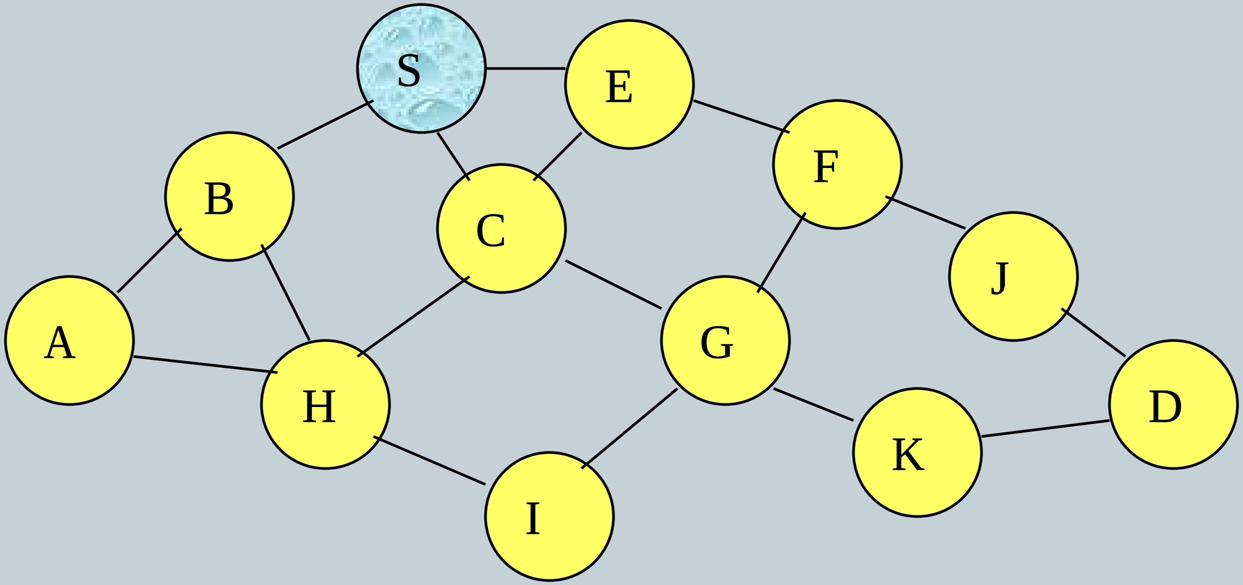
Disadvantages of DSR

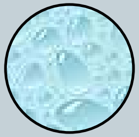
- Packet header size grows with route length
  - Large overhead if data size is small
- Flood of route requests may potentially reach all nodes in the network
  - Even if the network is stable

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
- 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

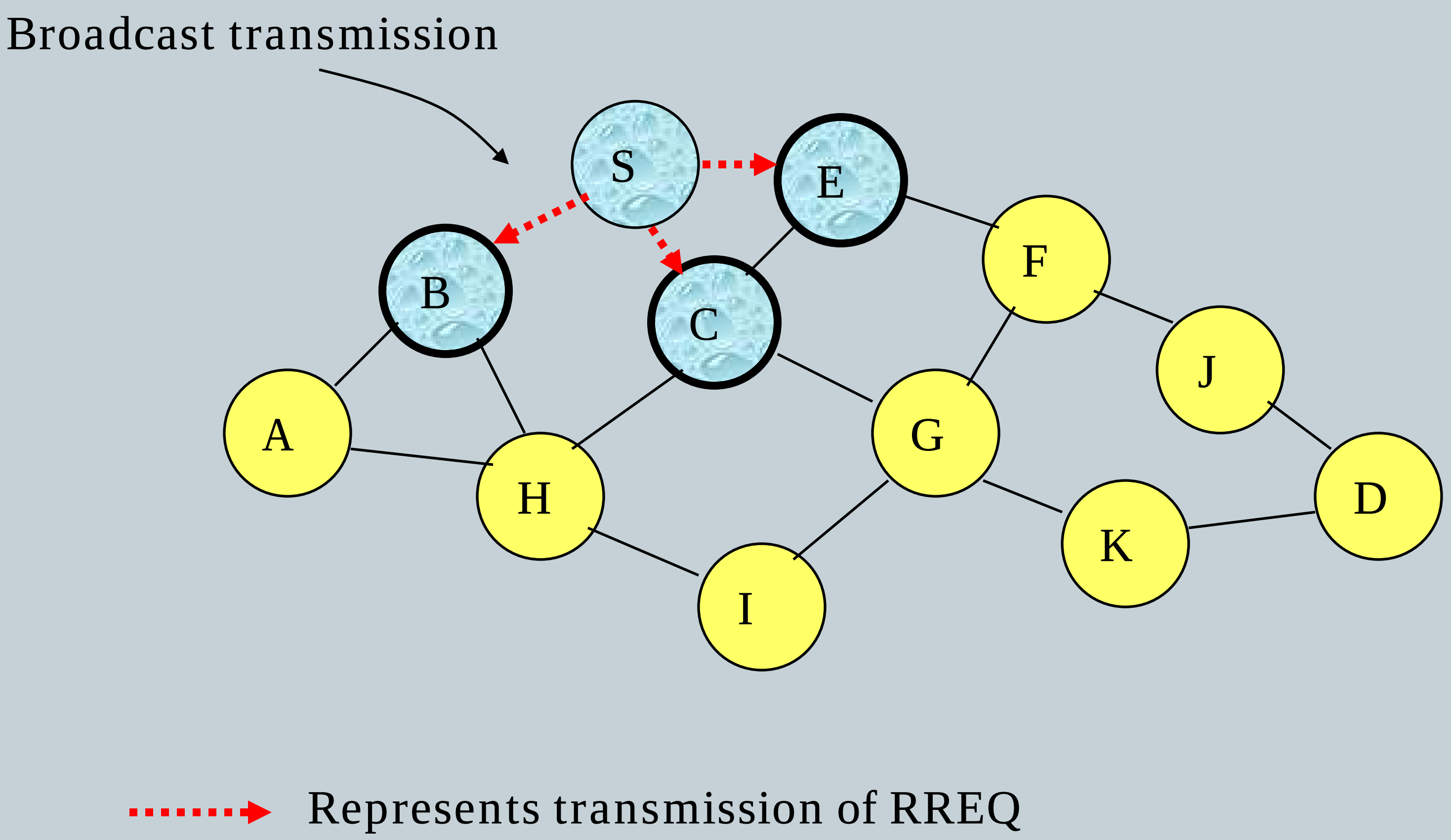
Route Requests in AODV



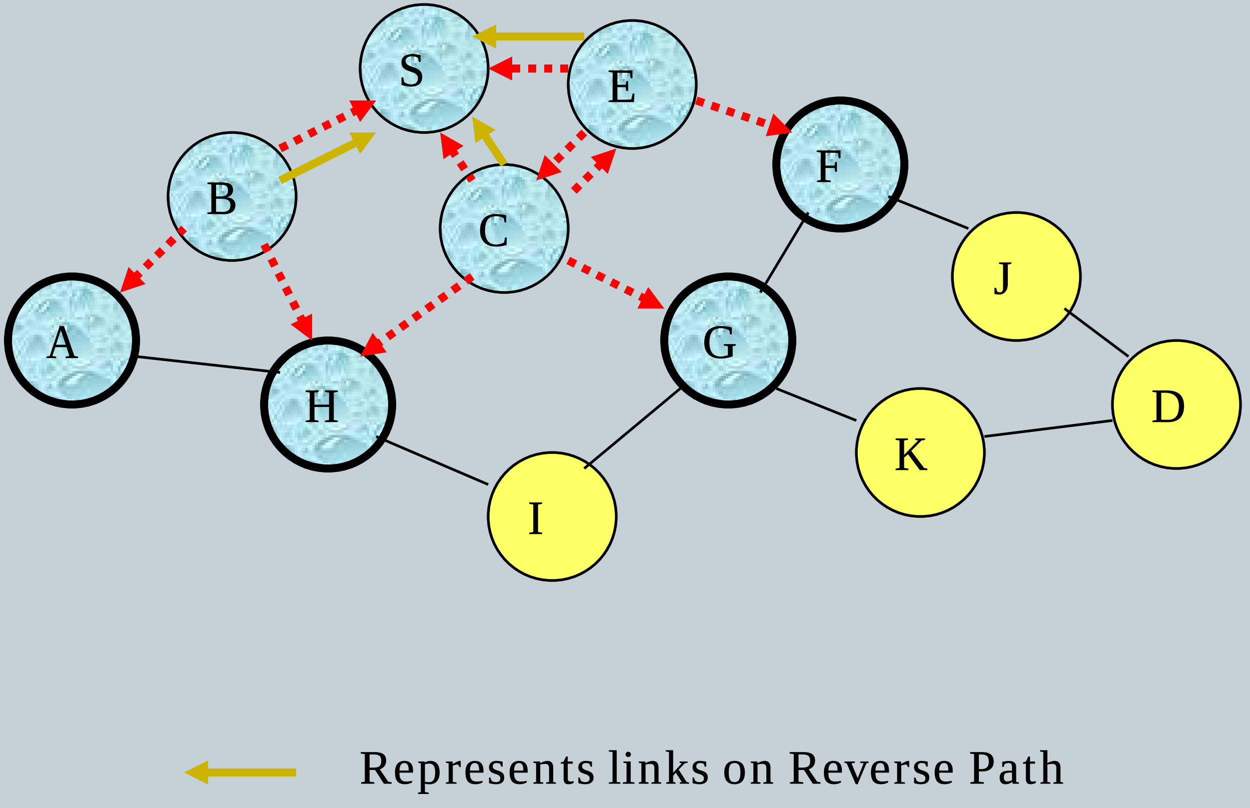


Represents a node that has received RREQ for D from S

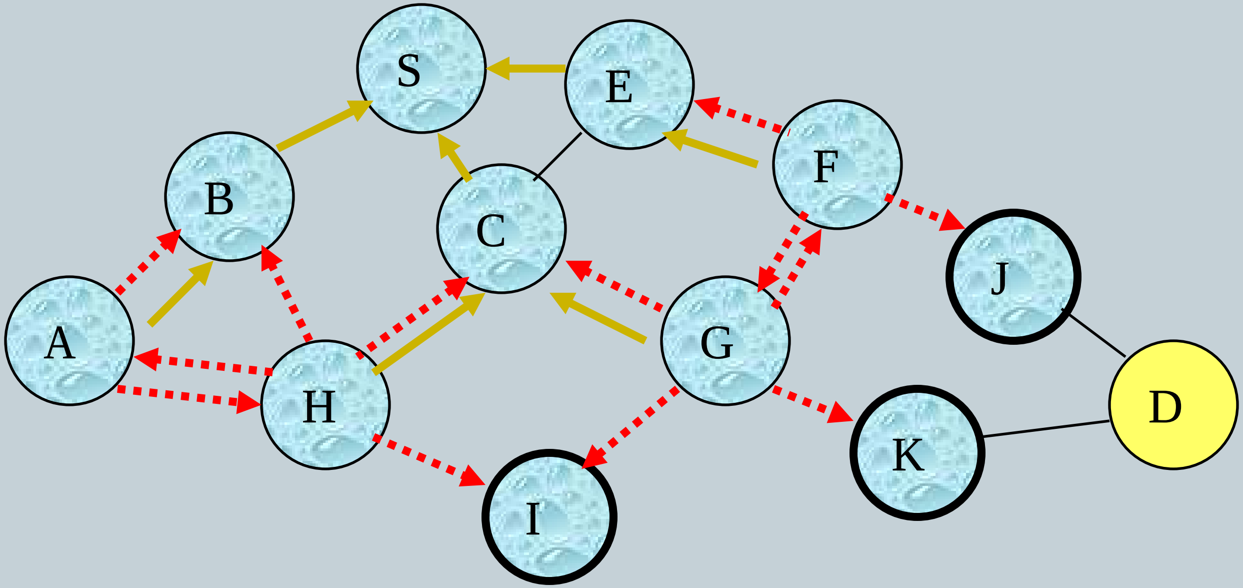
Route Requests in AODV



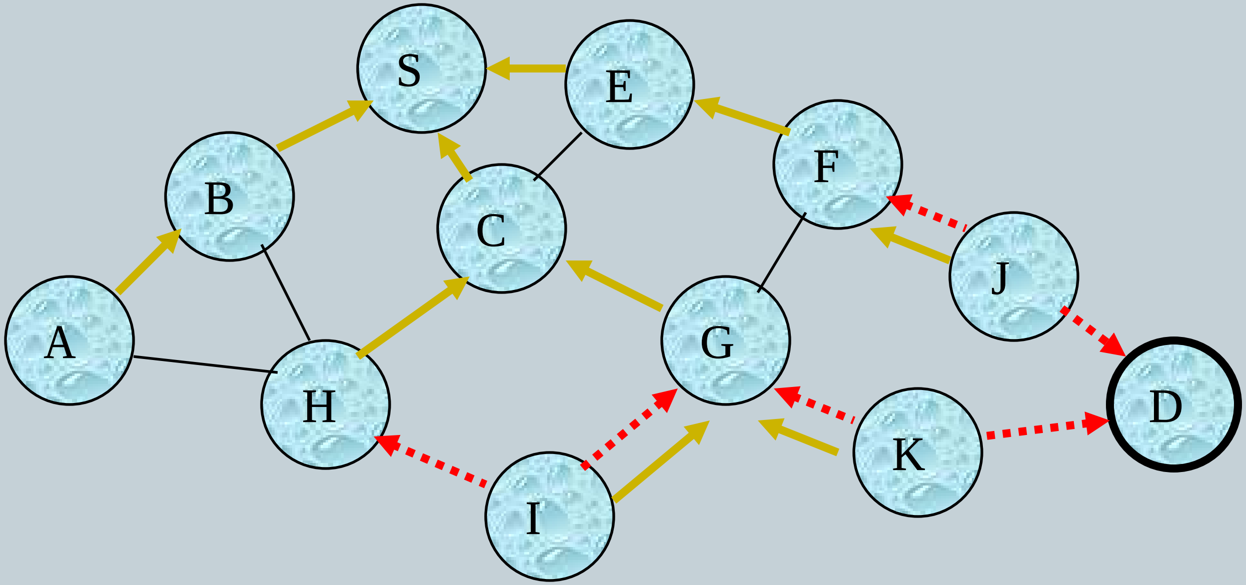
Route Requests in AODV



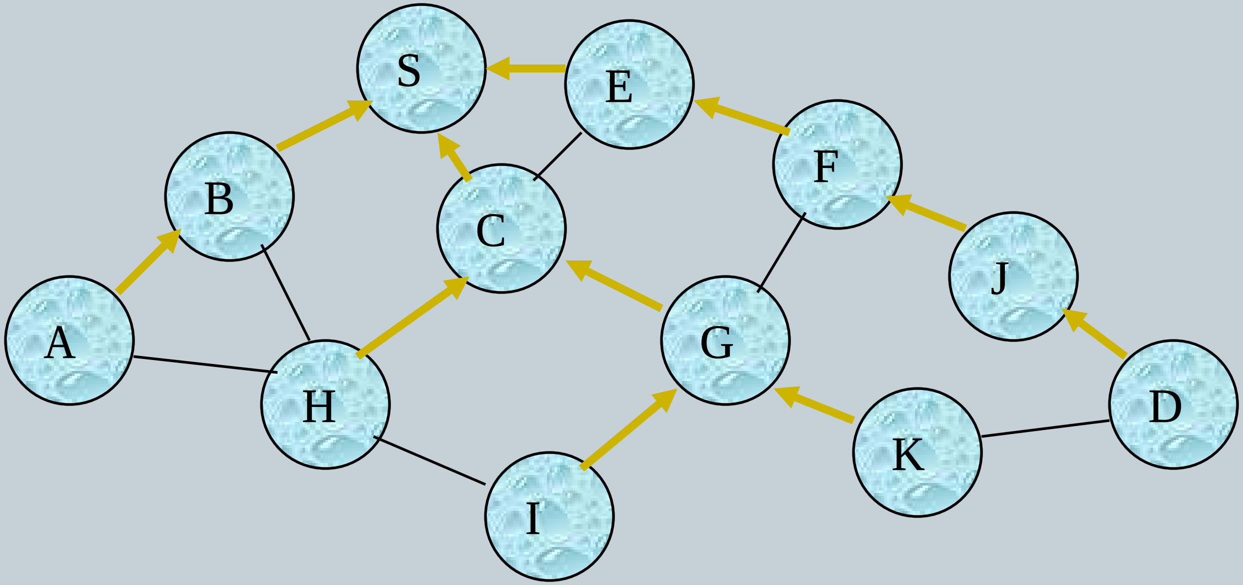
Reverse Path Setup in AODV



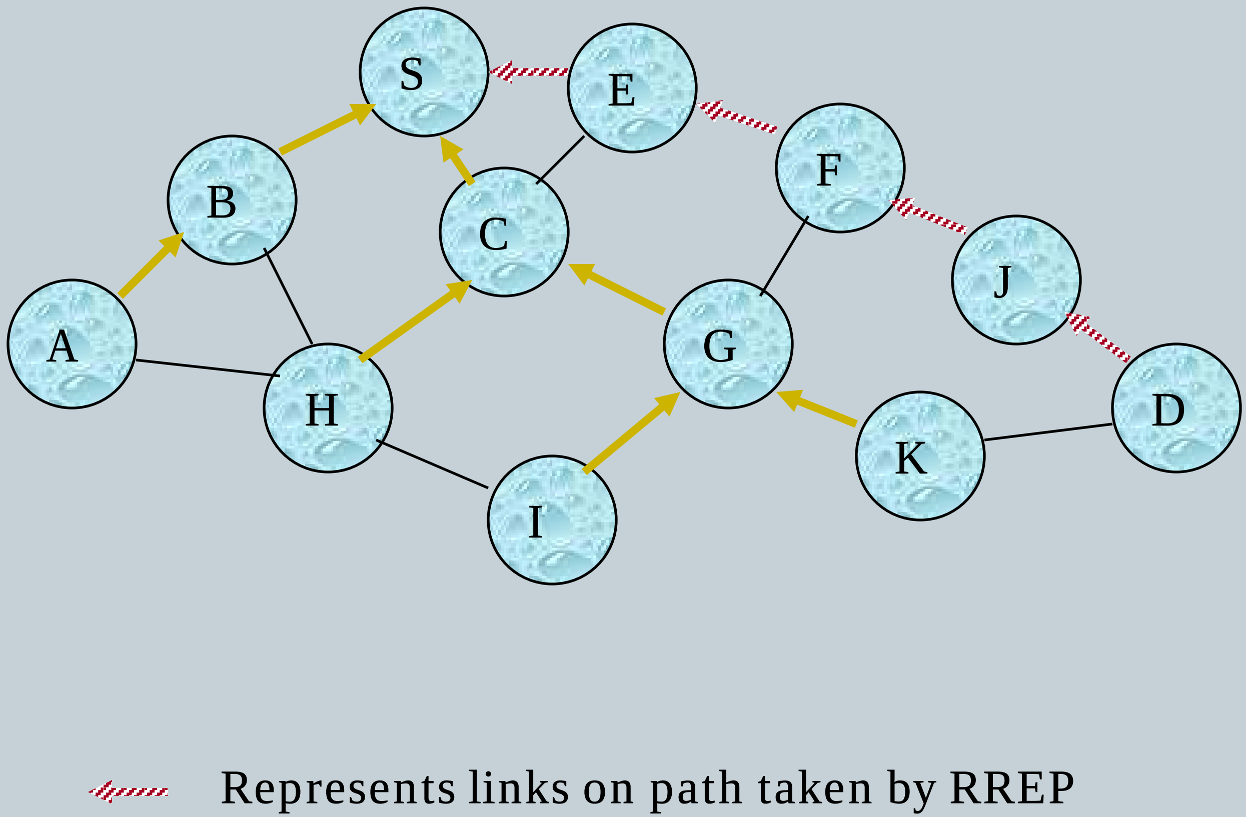
Reverse Path Setup in AODV



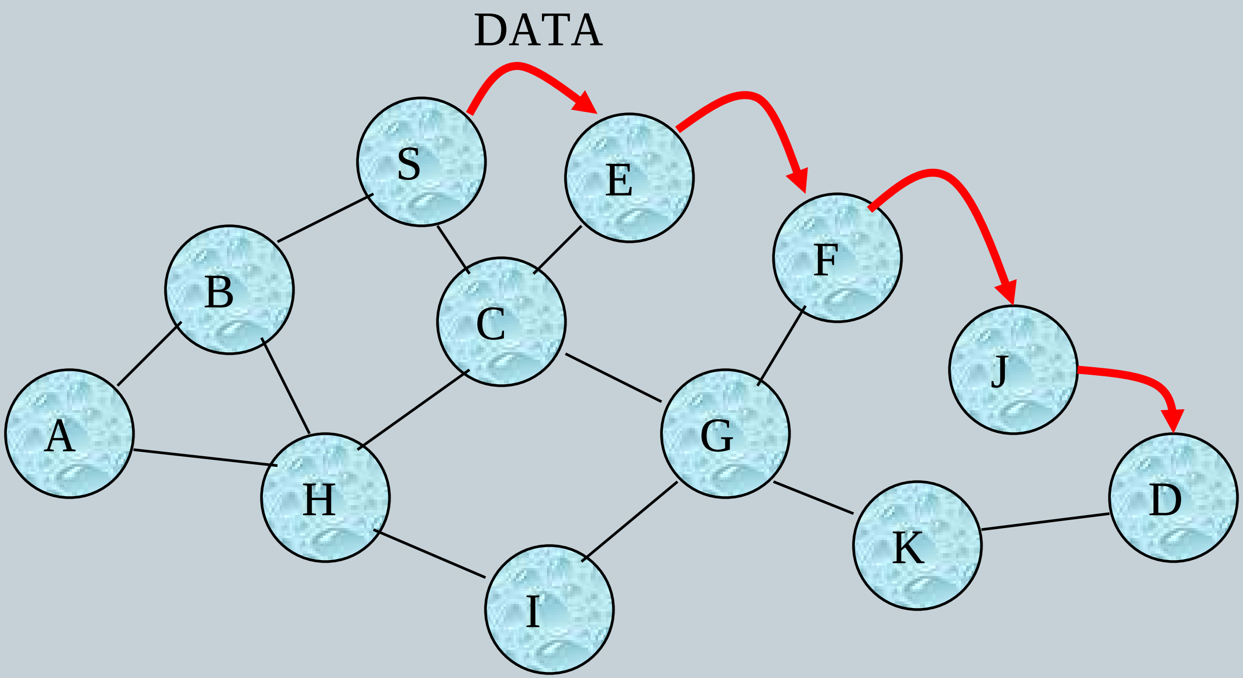
Reverse Path Setup in AODV



Route Reply in AODV



Data Delivery in AODV



Routing table entries used to forward data packet.

Route is *not* included in packet header.

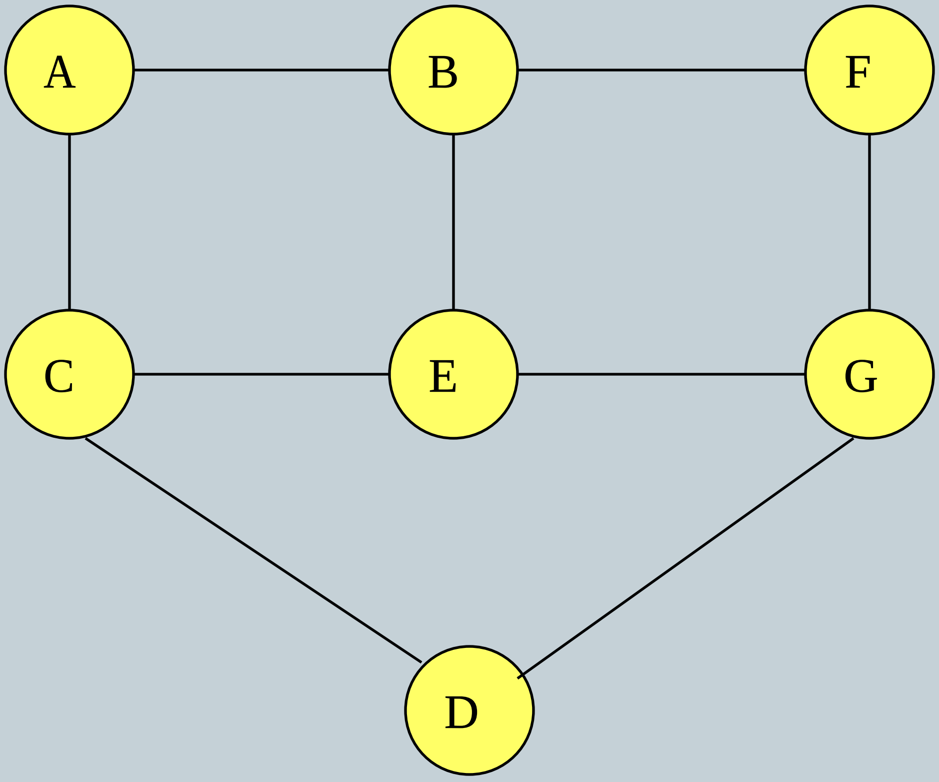
Advantages of AODV

- Routes maintained only between communicating nodes
  - reduces overhead of route maintenance
- No Packet header overhead as in DSR
  - but now we need (small?) routing tables
- Shortest, loop-free paths

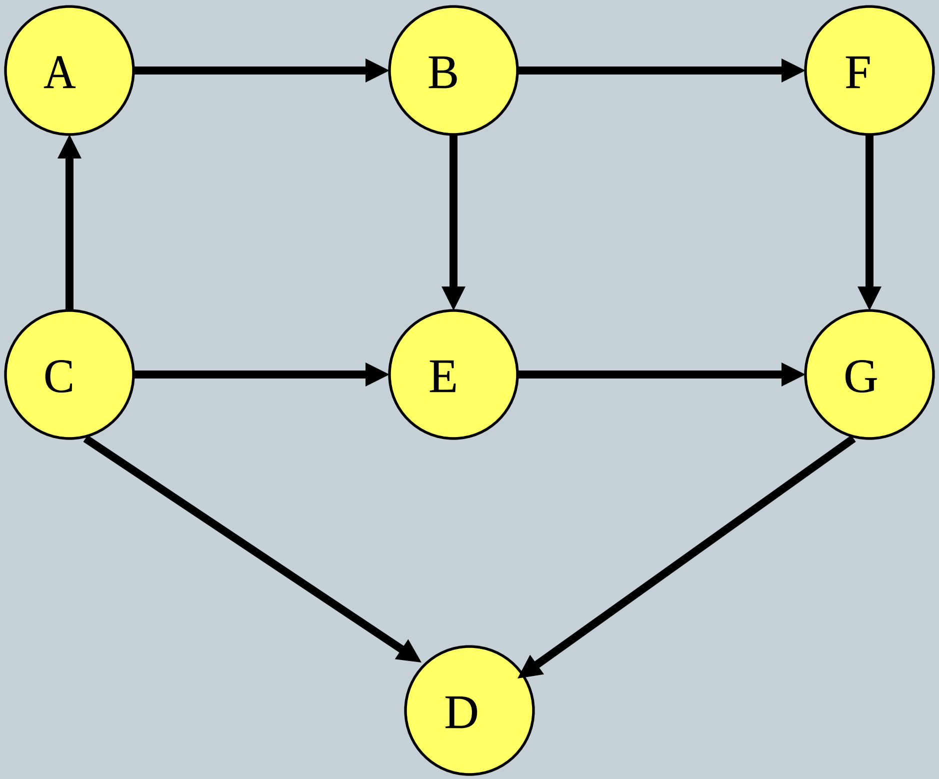
Disadvantages of AOADV

- Does not work if links are not bidirectional
- Does not allow multipath routing
- Flood of route requests may potentially reach all nodes in the network
  - Even if the network is stable

Link Reversal Algorithm (Simplified TORA)



Link Reversal Algorithm



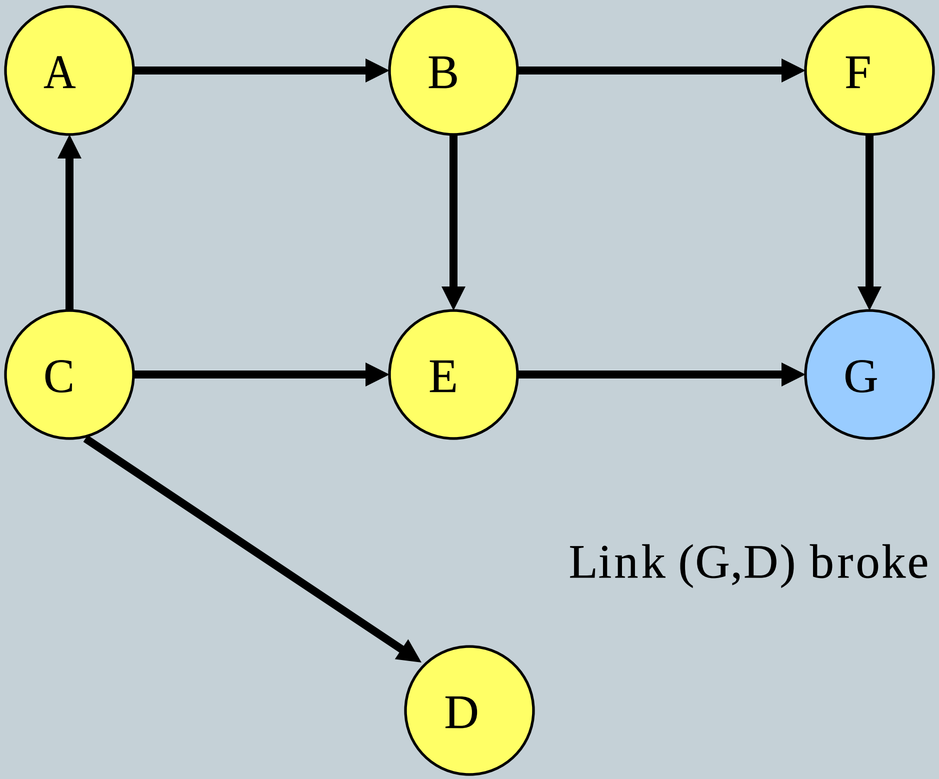
Links are bi-directional

But algorithm imposes logical directions on them

Maintain a directed acyclic graph (DAG) for each destination, with the destination being the *only sink*

This DAG is for *destination node D*

Link Reversal Algorithm

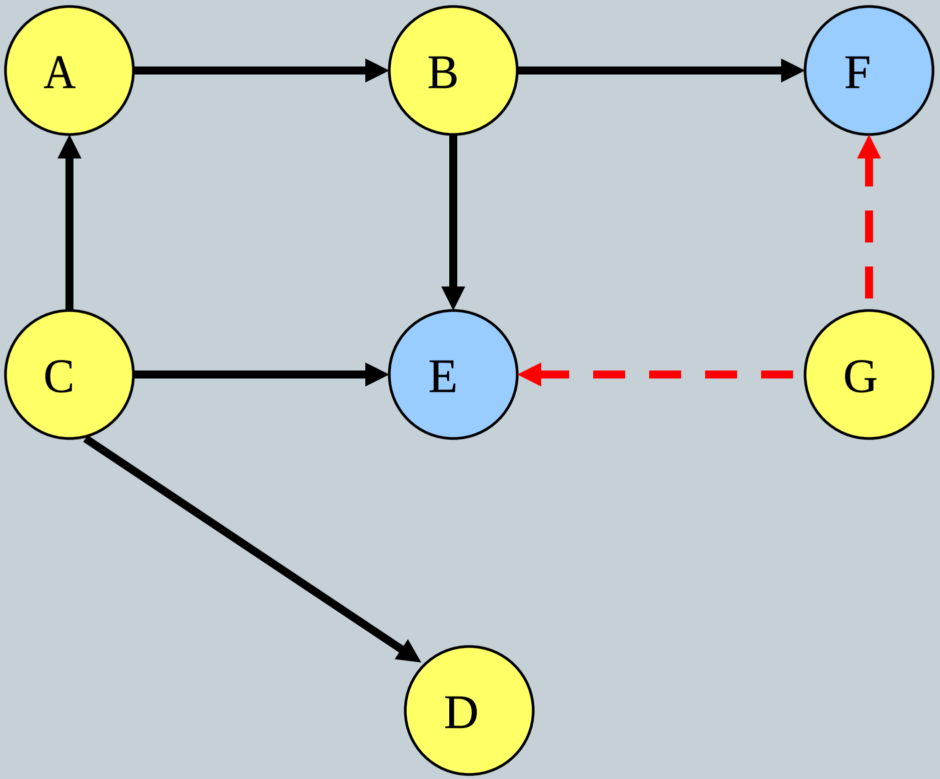


Link (G,D) broke

Any node, other than the destination, that has no outgoing links reverses all its incoming links.

Node G has no outgoing links

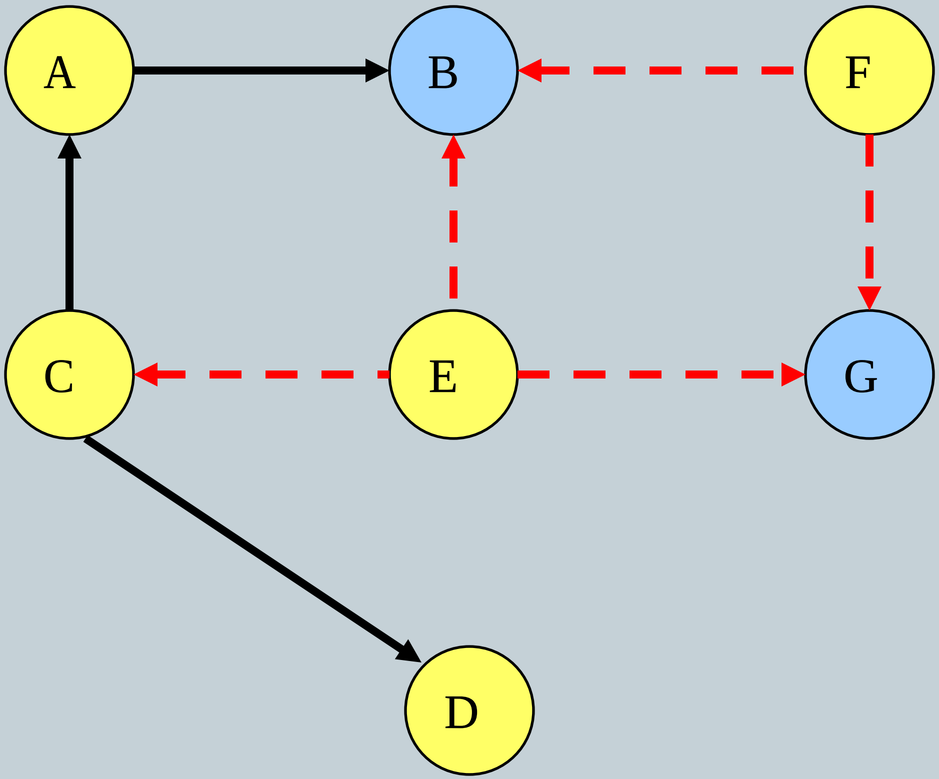
Link Reversal Algorithm



Represents a link that was reversed recently

Now nodes E and F have no outgoing links

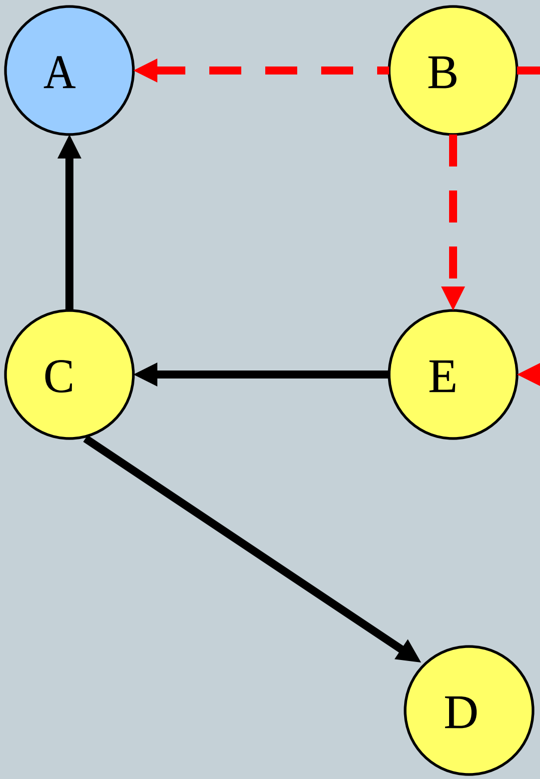
Link Reversal Algorithm



Represents a link that was reversed recently

Now nodes B and G have no outgoing links

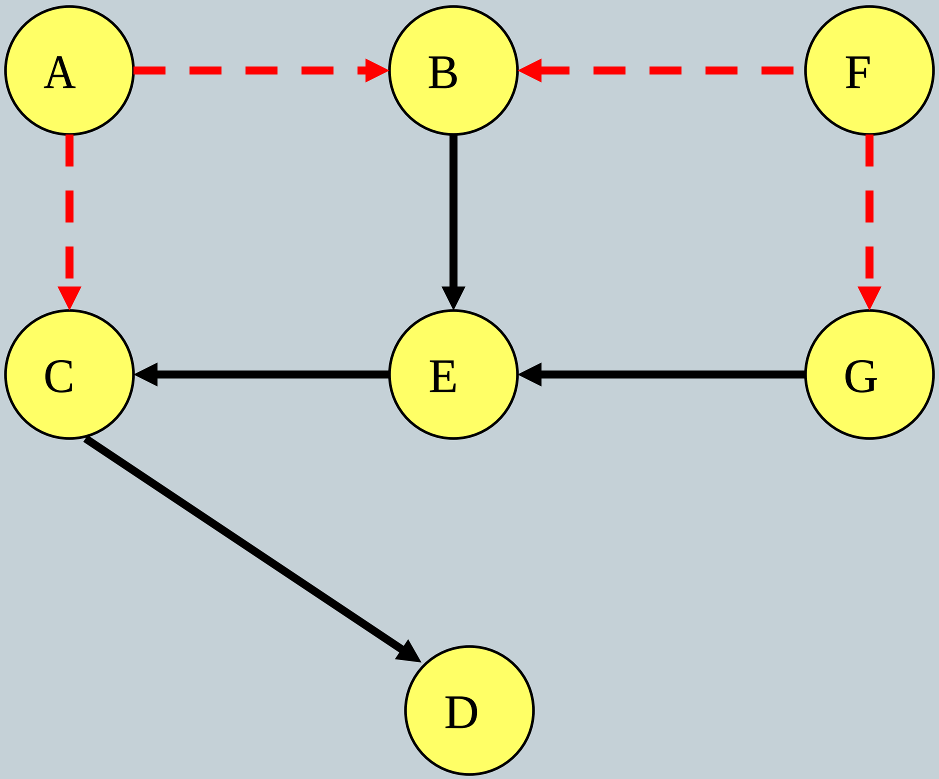
Link Reversal Algorithm



Represents a link that was reversed recently

Now nodes A and F have no outgoing links

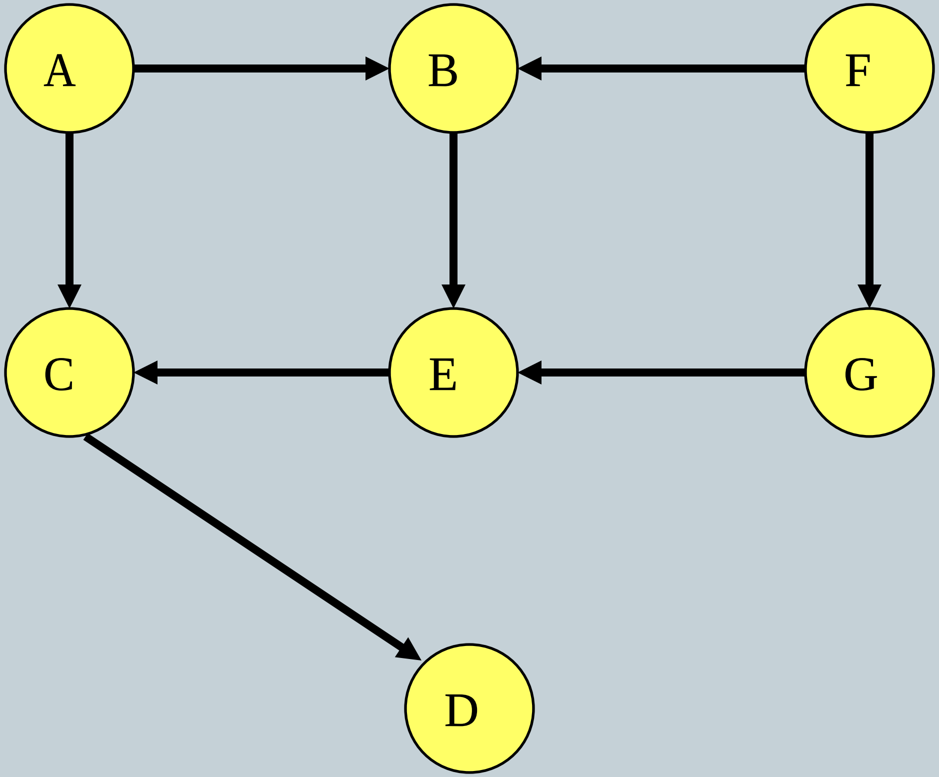
Link Reversal Algorithm



Represents a link that was reversed recently

Now all nodes (other than the destination D) have outgoing links

Link Reversal Algorithm



DAG has been restored with only the destination as a sink

Advantages of Link Reversal Algorithm

- No flooding of control packets
  - The initial construction does result in flooding of control packets
- Purely local failure recovery

Disadvantages of Link Reversal Algorithm

- Does not work if the links are not bidirectional
- Requires synchronization
- High overhead of route maintenance
  - Routes maintained between nodes even if they do not communicate

What we did not cover?

- Wireless routing protocol
  - Nothing interesting in particular
- Specific design of TORA
  - It is good to know the fundamentals (link reversal routing)
  - Link reversal has strong lower bounds – too much overhead
- Associativity- and Signal-stability- based routing
  - DSR/DSDV with mobility (or the lack of)/signal strength as a performance metric
  - What is a performance metric?

Looking forward ....

- The discussion so far assumed that nodes communicate with each other
  - Today, networks are information-oriented
    - Do not care about the location of the information

- Directed diffusion
    - Interested in information rather than the end-host
    - route on “flat” identifiers

Looking forward ….

- What are links in wireless networks?
  - Who are my neighbors?
  - How to assign weights to these links?
    - hop-count could be a really bad metric – why?
  - LOF – how (not) to assign link weights!

- Who are my neighbors?
  - Fundamental trade-off:
    - Transmit at higher power: more neighbors, more interference
    - Transmit at lower power: fewer neighbors, less interference

Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks

Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin @USC/UCLA

Mobicomm 2000

Presented by Rachit Agarwal & Lewis Tseng

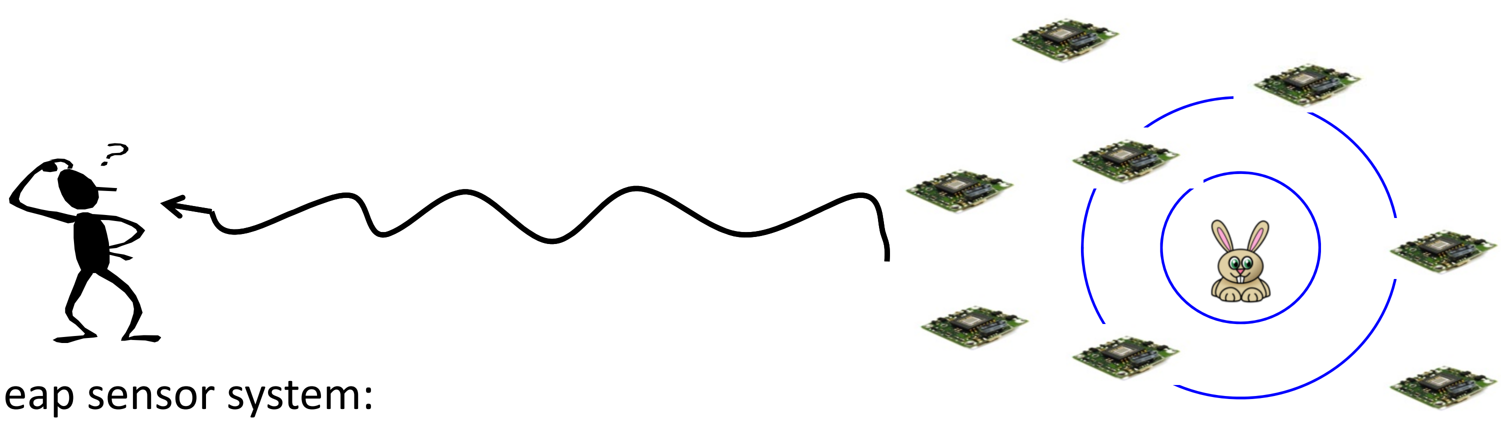
Some of the slide figures based on the paper and the following presentation:
<http://snslab.kangwon.ac.kr/home/page/semFile/2004sum/Directed%20Diffusion%20for%20Wireless%20Sensor%20Networks(dkmoon).ppt >

Outline

- Motivation
- Core Design
- Main Contribution
- Evaluation
- Discussion

Motivation

- What if sensor do not have global knowledge?
- "How many four-legged animal do you observe in the geographical region X?"



Cheap sensor system:
- Simple
- Spatial dense → Close to object → High SNR
- Energy efficient
- Able to route through holes

Motivation

- Need a new set of communication primitives that is energy efficient and considers the following:

- Task-specific
- Data-centric
- Based on only local information
- Coordination

Naming & Interest

- Task is known to every node in advance
- Task descriptions contains some attribute-value pairs
- Query/Interest:

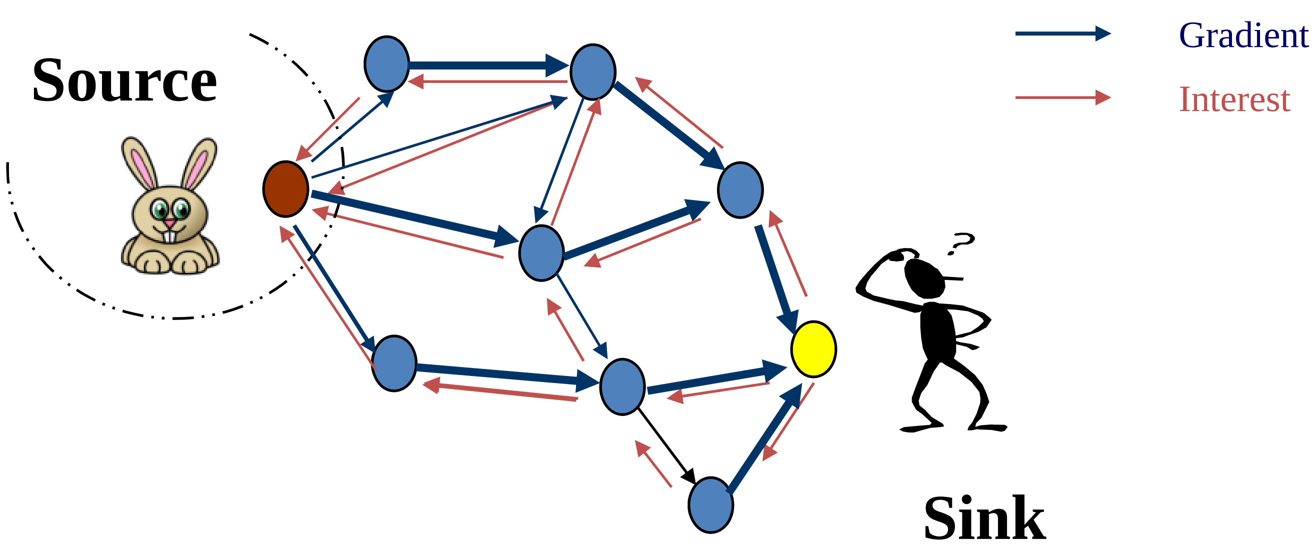
- Type = four-legged animal
- Interval = 20 ms
- Duration = 10 sec.
- Rect = [-100,100,200,400]

Directed Diffusion

- Sink broadcasts interest to neighbors
- Any node receiving a new interest first caches it and then sets up gradients towards the neighbor sending (or forwarding) the interest
- When source detects something, it checks its cache; if it finds match, sends reply using gradient
- Any node receiving a reply checks its cache, and forwards using gradient
- Sink then reinforce the "best" route

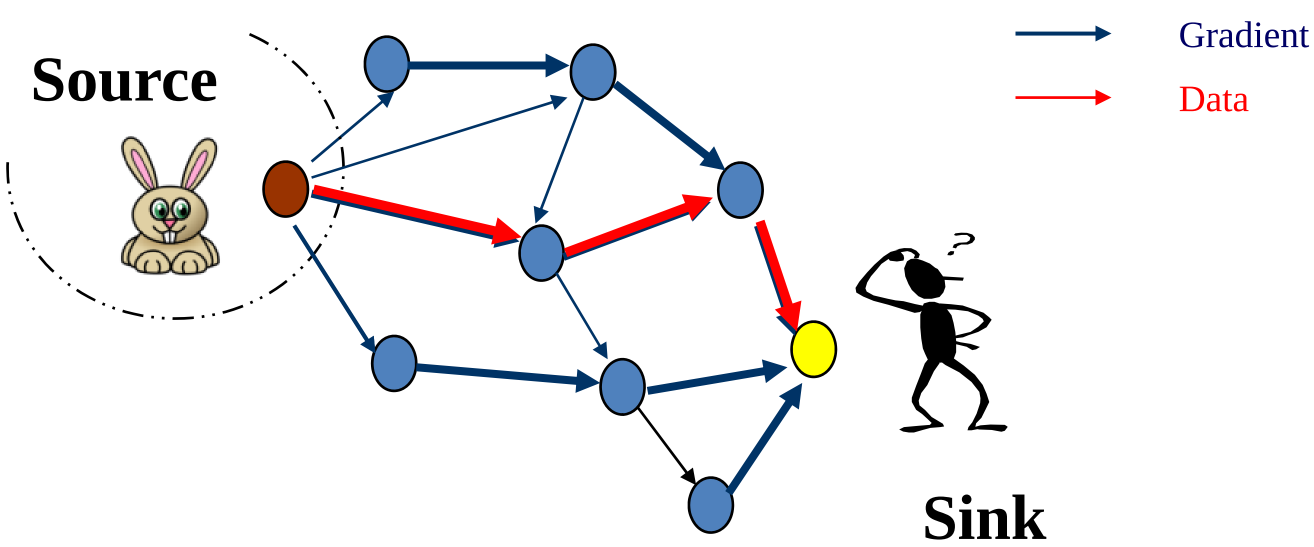
Interests & Gradient

- Sink broadcasts interest to neighbors
- Node sets up gradient



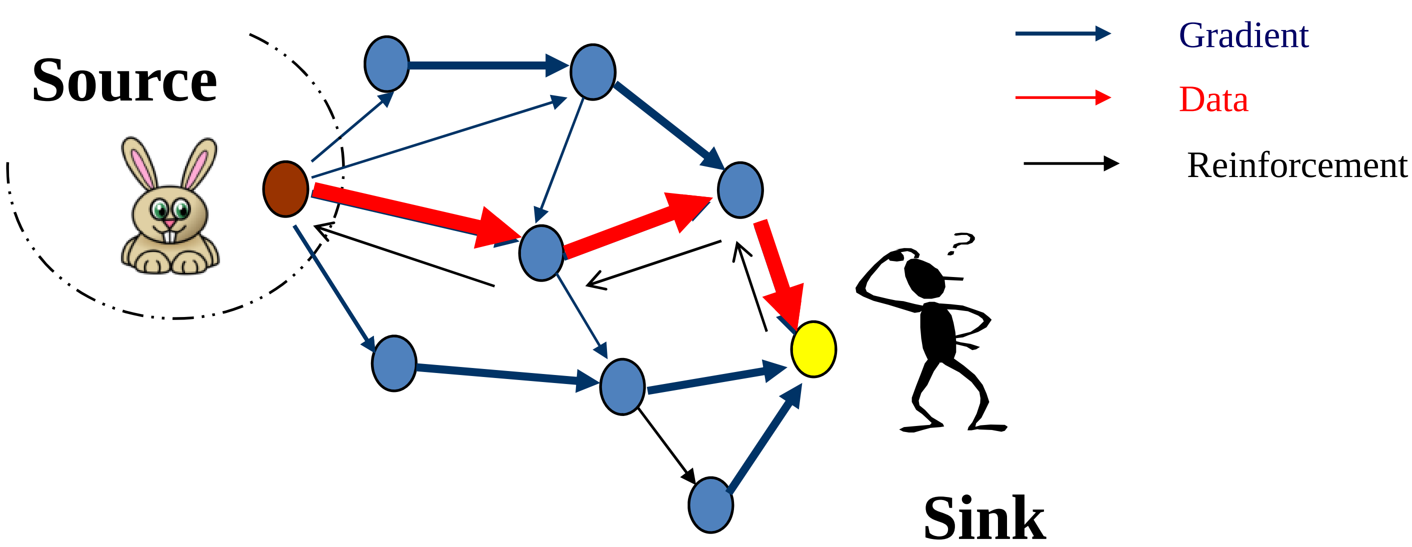
Data Propagation

- If source finds matched interest in the cache, it unicasts to neighbor using gradient
- Node forwards accordingly



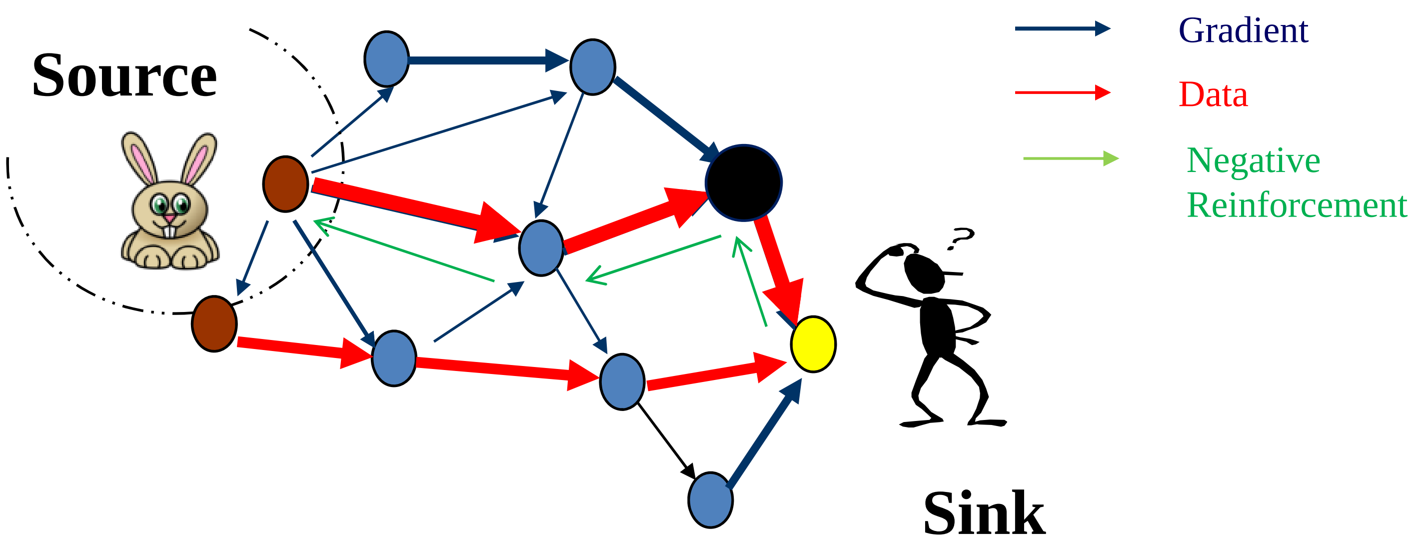
(Positive) Reinforcement

- Sink reinforces one particular neighbor in order to pull higher quality observations by some local rules or to perform local repair



(Negative) Reinforcement

- Negative reinforcement can be used to perform route truncation, loop removal or reinforce a consistently better route



Cache

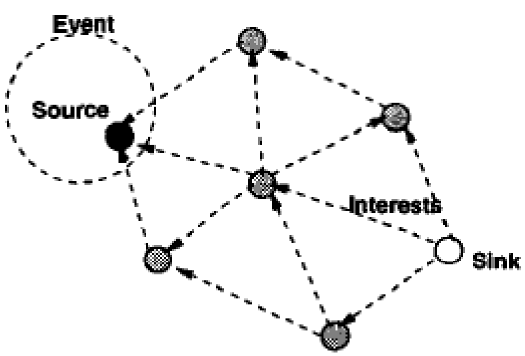
- Interest Cache
- Stores

  Interest

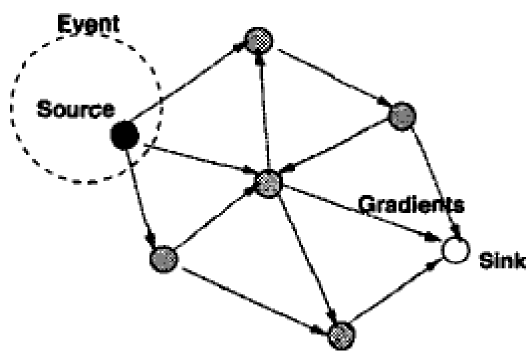
  Corresponding timestamp and gradient
- Contains no information about sink
- Data Cache
- Stores

  reply message (Type, Instance, Location, Intensity, Confidence, Timestamp)
- To prevent from loop and to perform aggregation

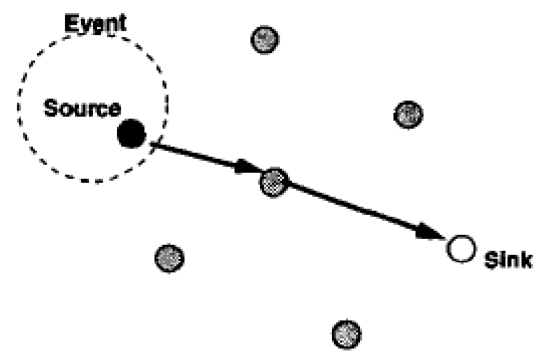
Summary



(a) Interest propagation



(b) Initial gradients set up



(c) Data delivery along reinforced path

- A reactive routing scheme:
- Broadcast: multiplicity of routes
- Gradient: data-centric routing
- Reinforcement: empirically best route
- Cache: loop avoidance, aggregation

Contribution

- A new set of communication primitives:
  - Task-specific
  - Data-centric
  - Based on only local information
  - Coordination

Every node can interpret data & Interest; Simple naming scheme

Only neighbor-to-neighbor comm.; Usage of Interest & Gradient

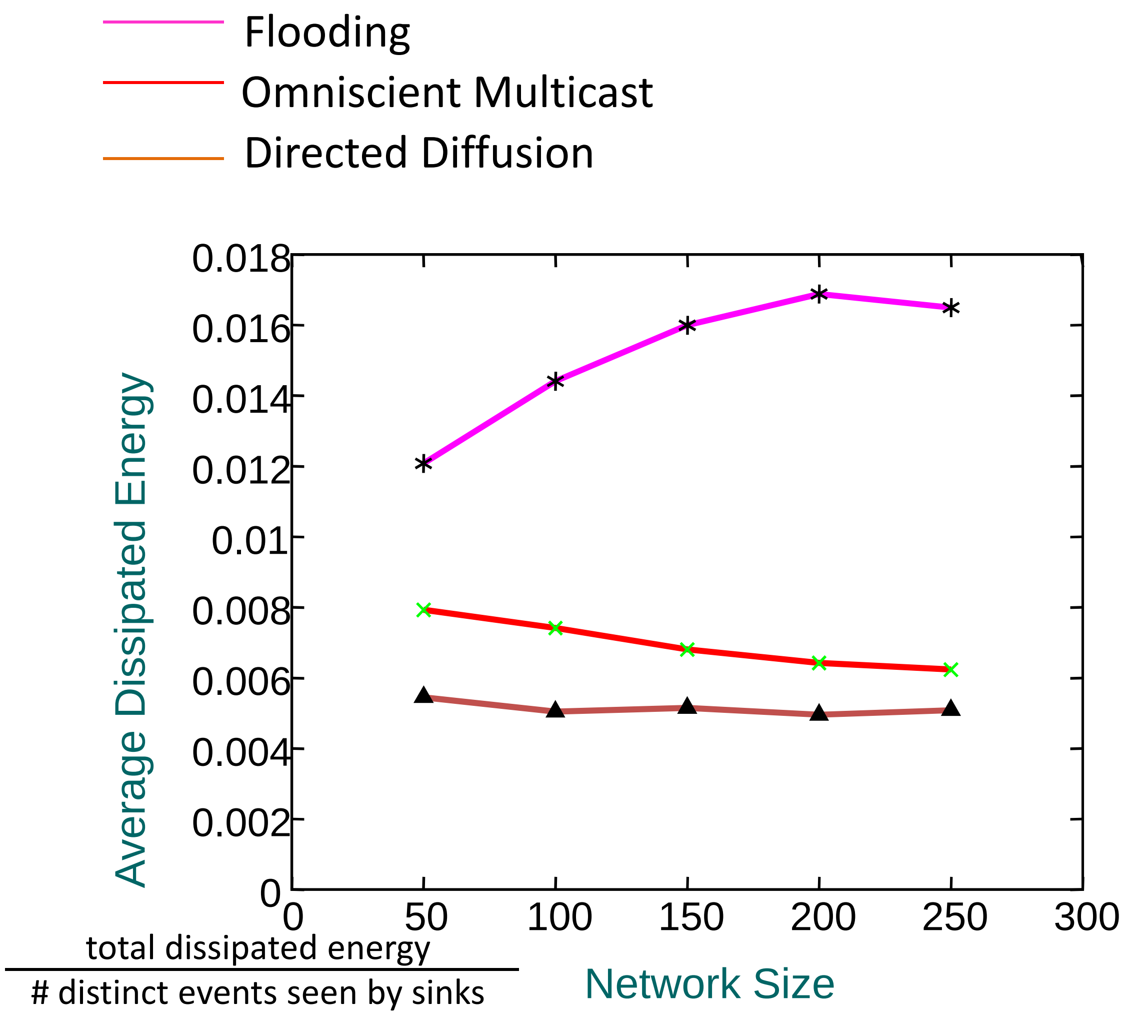
No globally unique ID & knowledge; Usage of gradient & reinforcement

Every node can cache, aggregation and process message

Evaluation - Methodology

- Result on average of 3 runs of ns2 simulation
- 50-250 sensors with roughly same density
- 5 sources (randomly chosen) and 5 sinks (uniformly scattered)
- Congestion-free communication

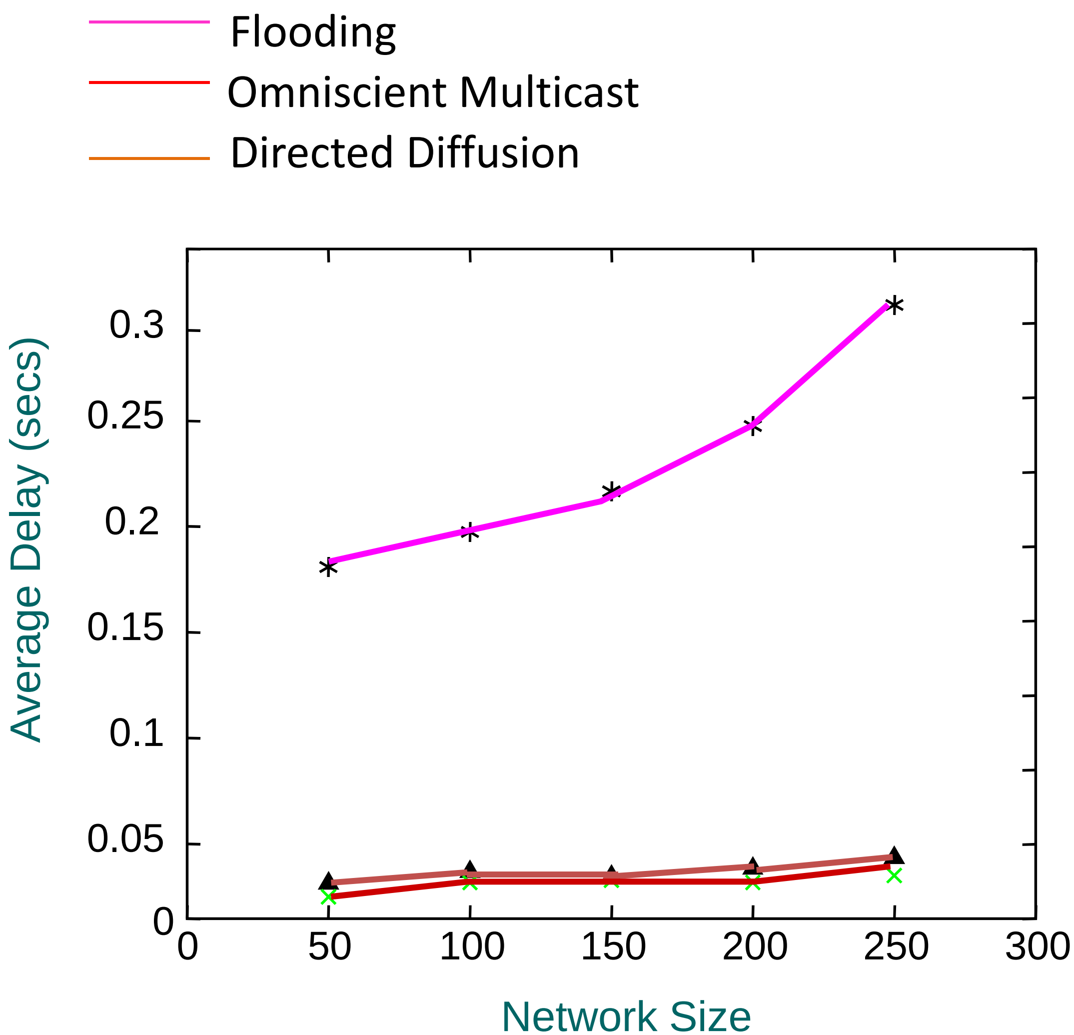
Comparative Evaluation



Omniscient Multicast has lower Average Dissipated Energy due to shortest-path multicast tree

Directed Diffusion has lowest Average Dissipated Energy due to in-network aggregation

Comparative Evaluation

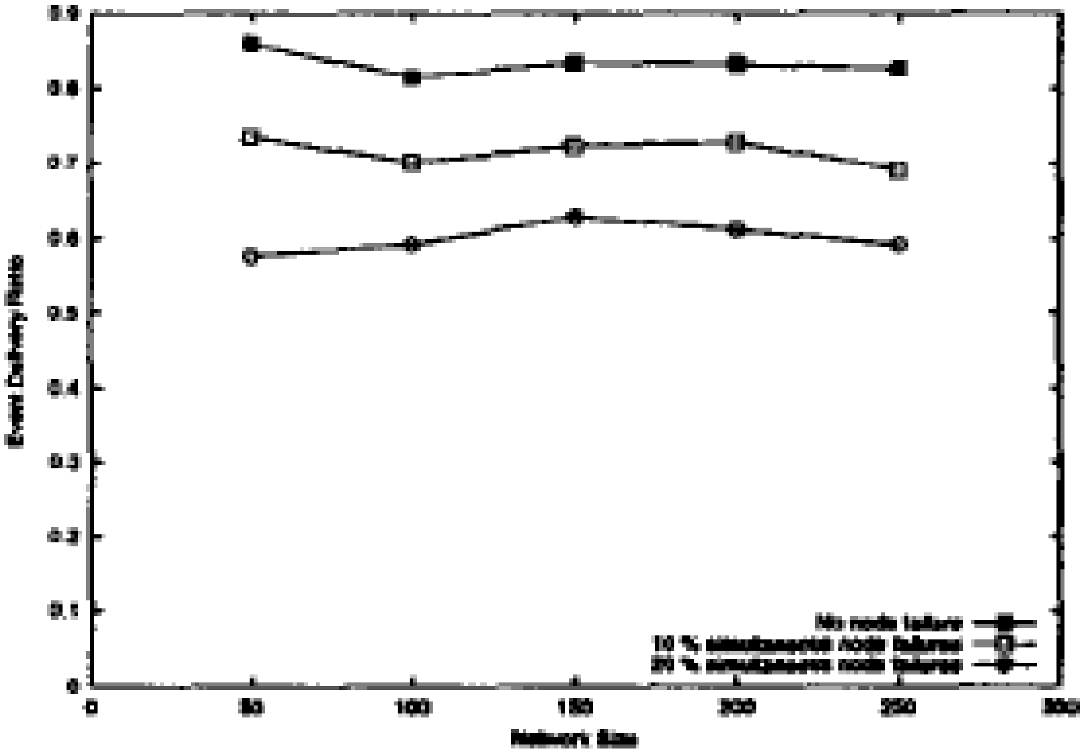


Omniscient Multicast and directed diffusion have roughly the same delay

Flooding is an order of magnitude higher due to artifact of the MAC layer broadcast

Impact of node failure

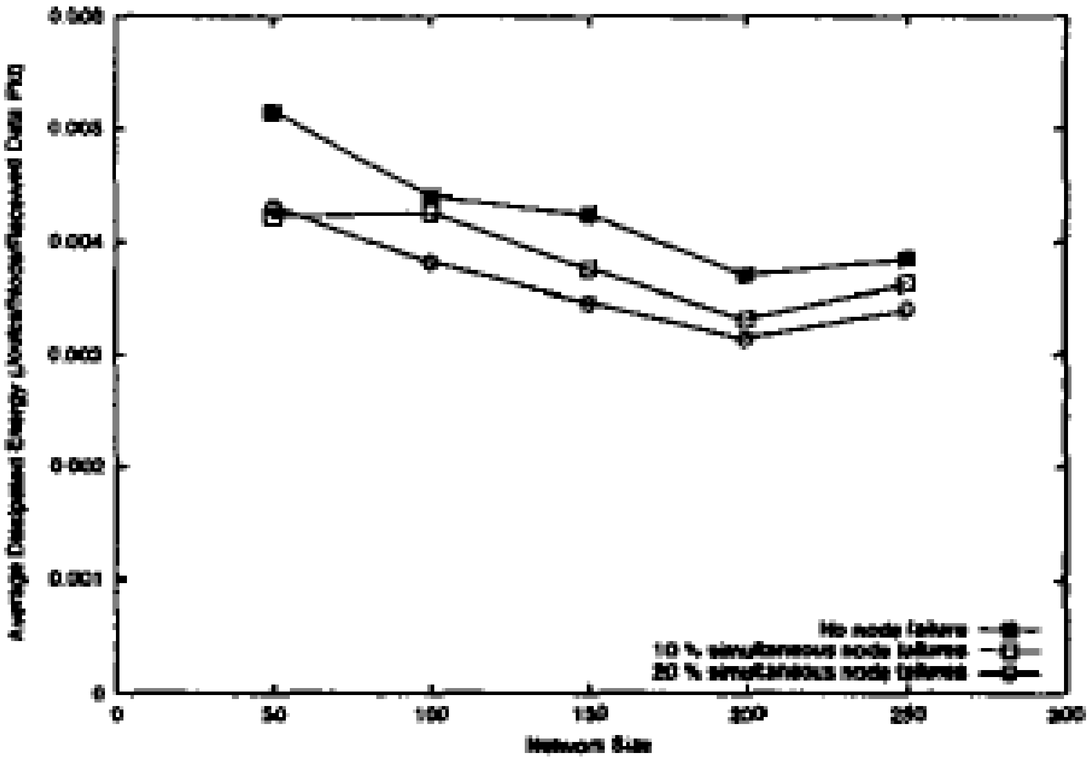
Directed diffusion is somewhat robust to node failure



(c) Event Delivery Ratio

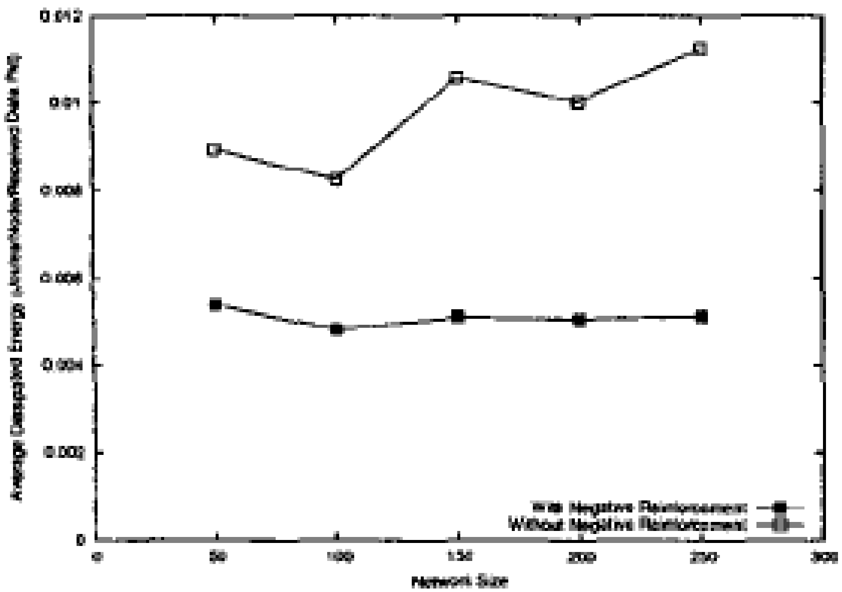
Impact of node failure

Not much overhead used to overcome node failure

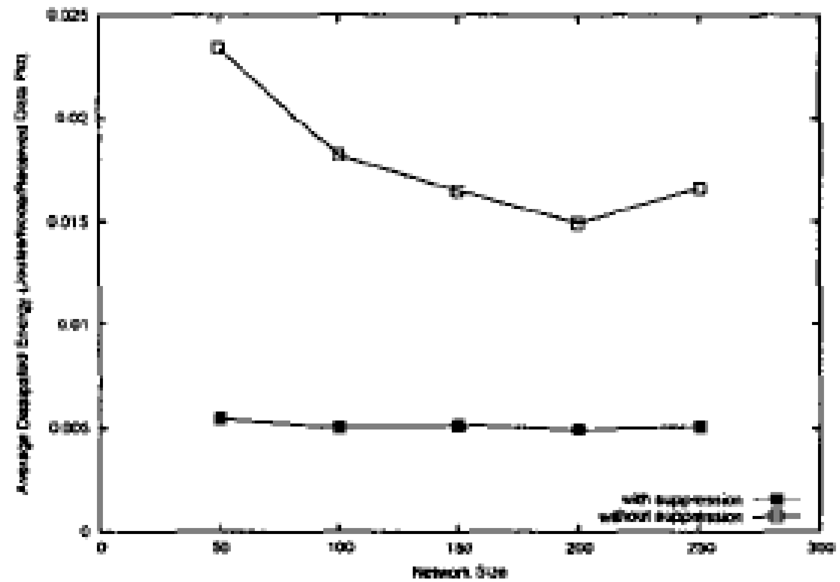


(a) Average dissipated energy

Sensitivity



(a) Negative reinforcement



(b) Duplicate suppression

Negative reinforcement really contributes

In-network aggregation really contributes

- Other factors (more realistic energy model, # sinks, #sources…etc.) in the tech. report

Discussion

- They list naming scheme as a possible future work. This will certainly affect the expressivity of tasks. But will naming affect performance of directed diffusion by much?

  - General attribute-based v.s. hierarchical naming

- Query/Interest:

- Type = four-legged animal
- Interval = 20 ms
- Duration = 10 sec.
- Rect = [-100,100,200,400]

Discussion

- How do you think about the idea of purely data-centric routing? What other types of scheme can be adopted?
- Though we are not aware of any practical usage of directed diffusion, the core idea of this paper can be utilized in other fields. Could you think of any usage?

Thank you for your attention!

Learn on the Fly: Data-driven Link Estimation and Routing in Sensor Network Backbones

PRESENTED BY:
- RACHIT AGARWAL
- LEWIS TSENG

Sensor Network Routing

- Sensor Network Routing requirements
  - energy efficiency
  - low latency
  - data reliability

- High-volume data traffic in a batch

- Large scale (possibly long route)
  - Directed diffusion is not suitable

Fundamental Questions

- Which next-hop should I forward the packet to?
- How to estimate link quality?

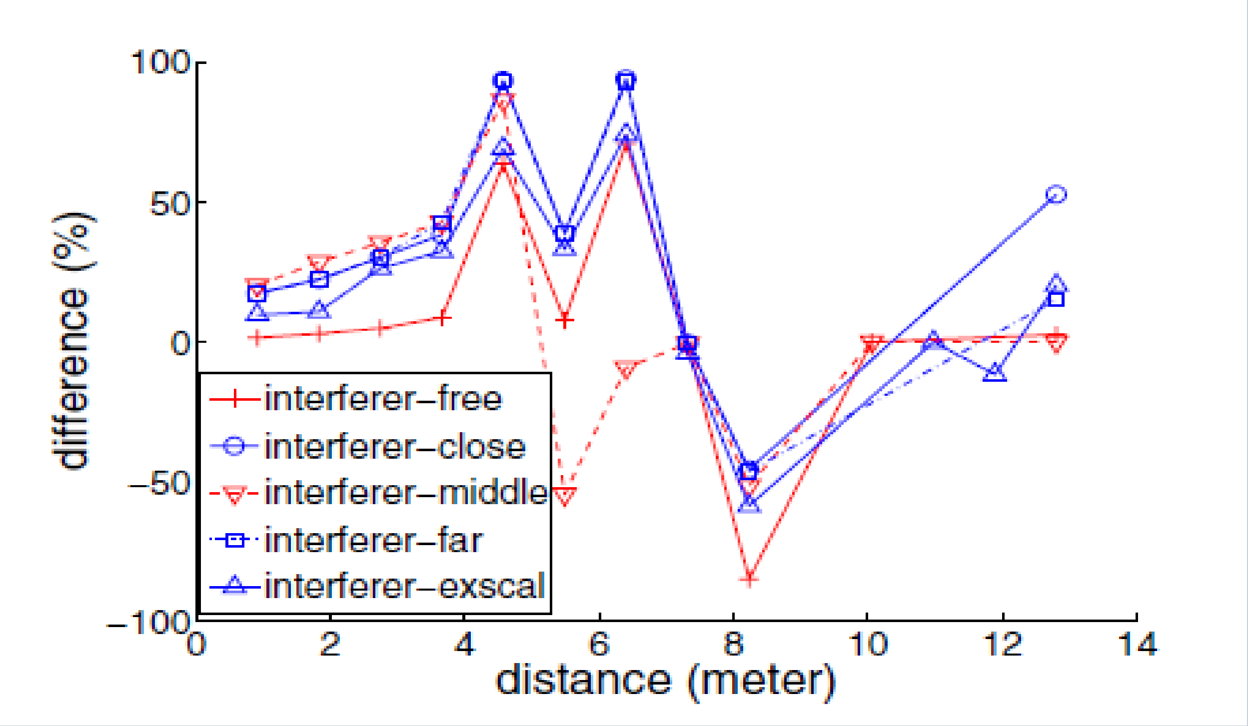
Traditional Approach

- Use control-plane beacon packets
  - Broadcast a "small" beacon packet to all your neighbors
    - Small beacons to avoid high overhead
  - Estimate link properties based on the broadcast results

- Unicast the (potentially much larger) data packet to the "best" neighbor

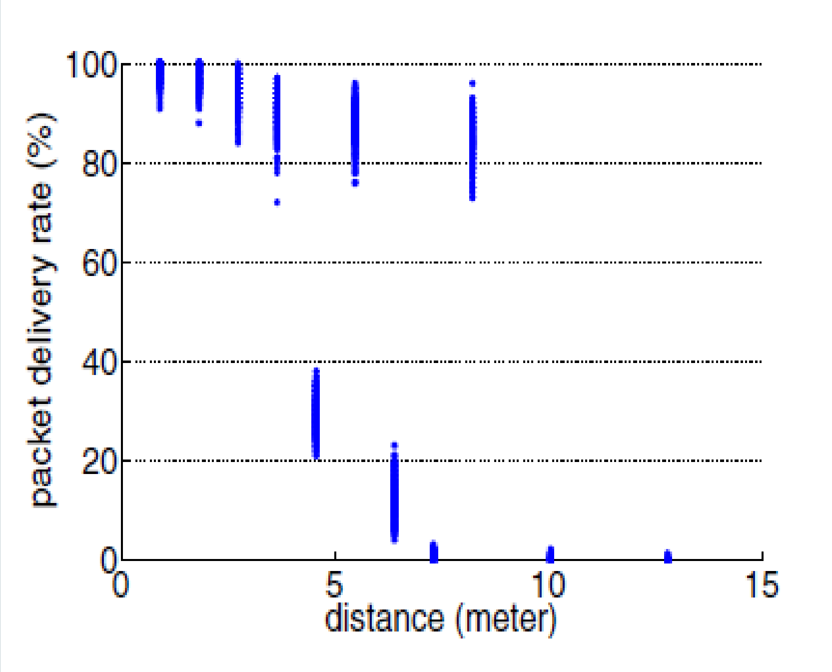
Problems with traditional approach (I)

Difference in packet delivery rate between broadcast and unicast

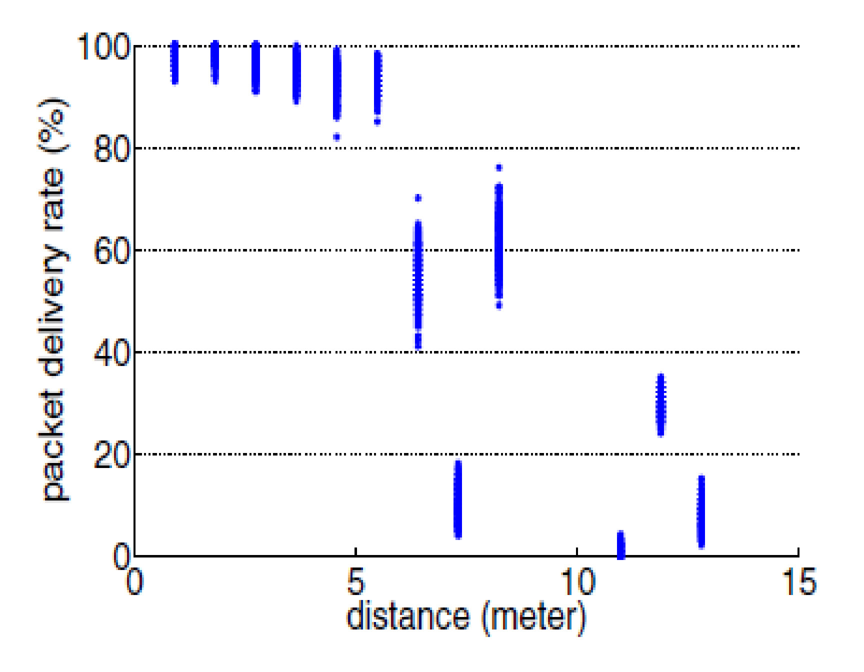


Problems in traditional approach (II)

Difference in packet delivery rate (broadcast) for packets of varying sizes



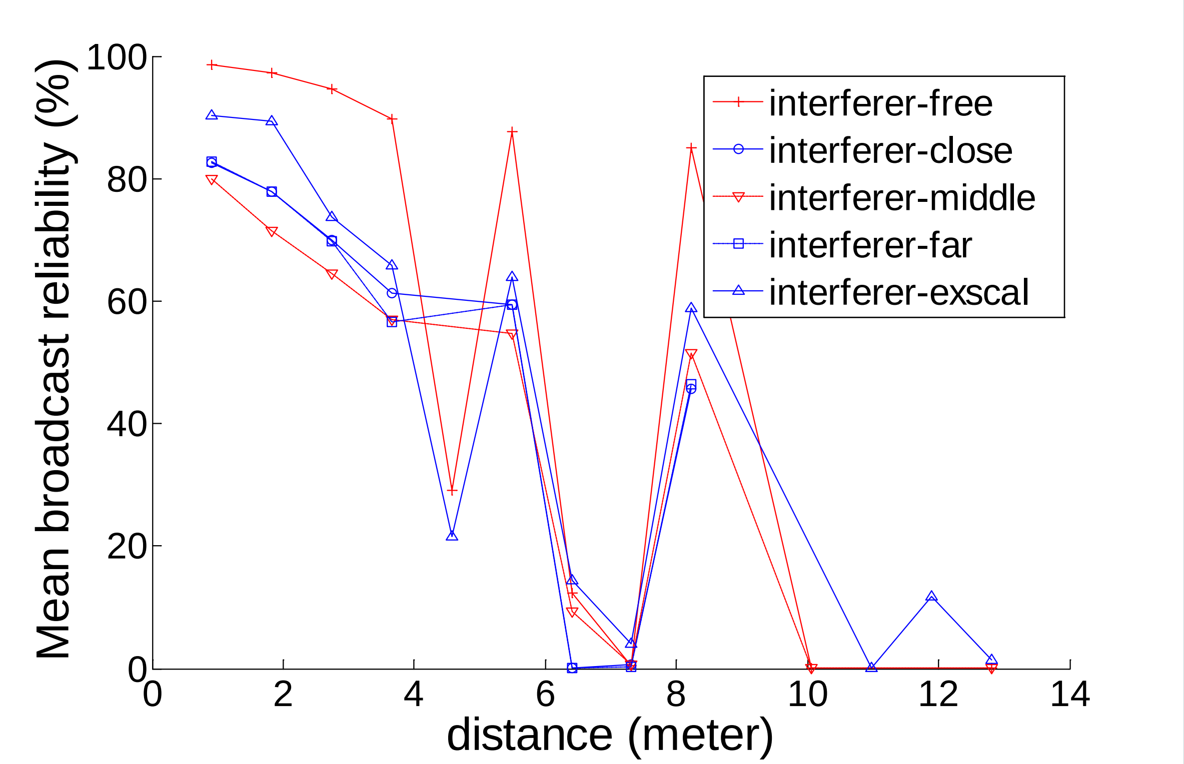
1200 bytes



300 bytes

Problems with traditional approach (III)

Variation in packet delivery rate due to
change in traffic pattern (interference)



Problems with traditional approach (IV)

- Temporal variations
- Spatial variations
  - Different coordination methods at the MAC layer
- Broadcast and unicast have different transmission rates
- Temporal correlations between link quality

Idea 1. Data-plane link estimation

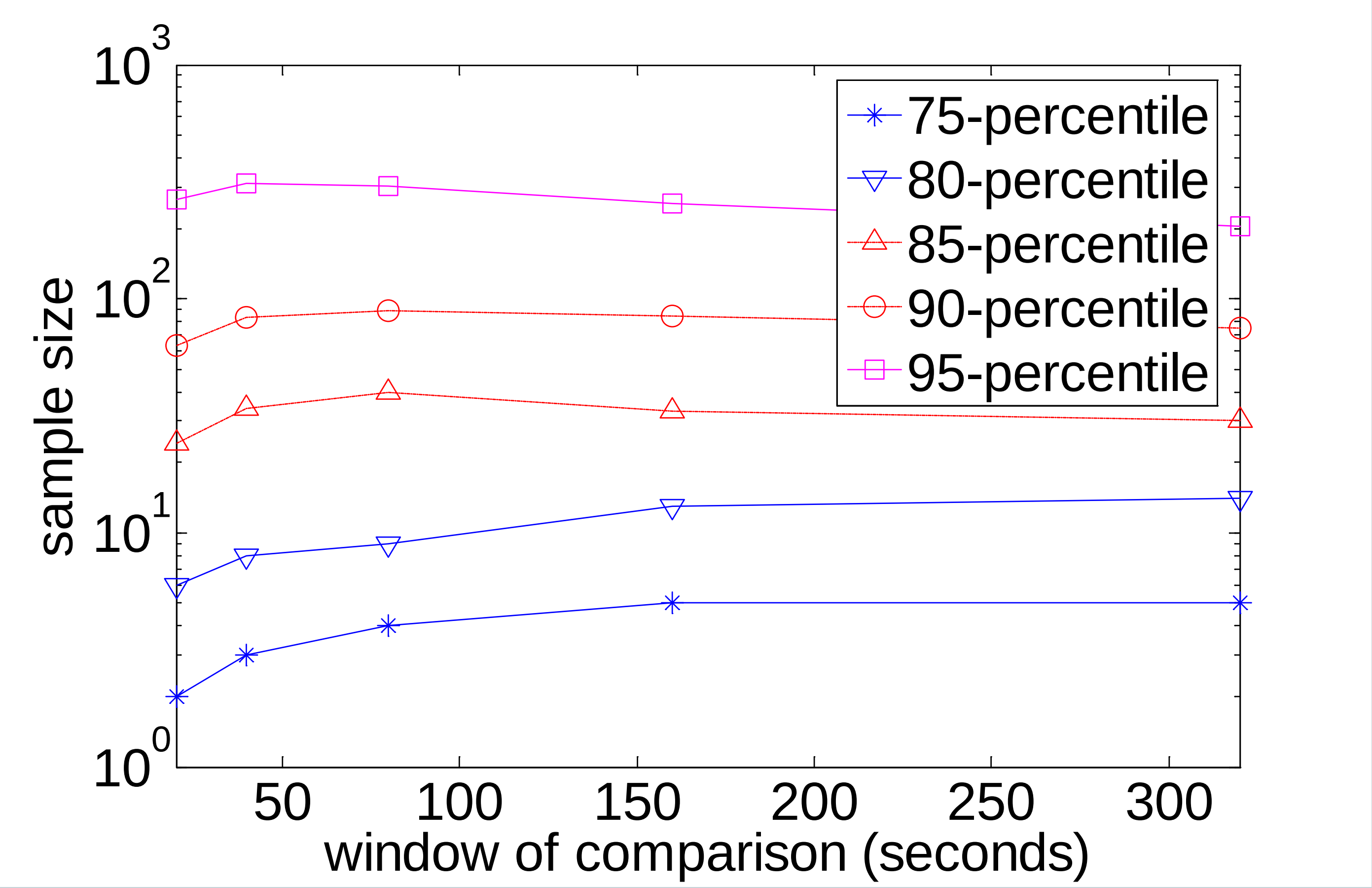
- Main idea: link estimation using the data packets
  - Requires no and very few beacon packets
    - Further reduces the energy consumption (?)
- Exploit MAC feedback mechanism –
  - Success or failure
  - MAC latency
    - time spent in transmitting a packet (including retries)

Idea 2. ELD metric

- Expected MAC latency per unit distance to the destination

- MAC latency reflects link reliability (number of MAC layer retries)
  - Routes of lower MAC latency tend to be more reliable
  - Reducing end-to-end MAC latency also improves network throughput

data packets required for selecting next-hop

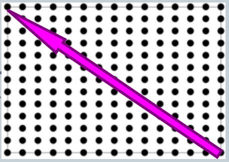


Experiment design: protocols studied

- Beacon-based routing
  - ETX: expected transmission count; geography unaware (Alec Woo et al. 2003, Douglas Couto et al. 2003)
  - PRD: product of link reliability and distance progress; geography based (Karim Seada et al., 2004)
- Several versions of LOF

Experiment design: evaluation method

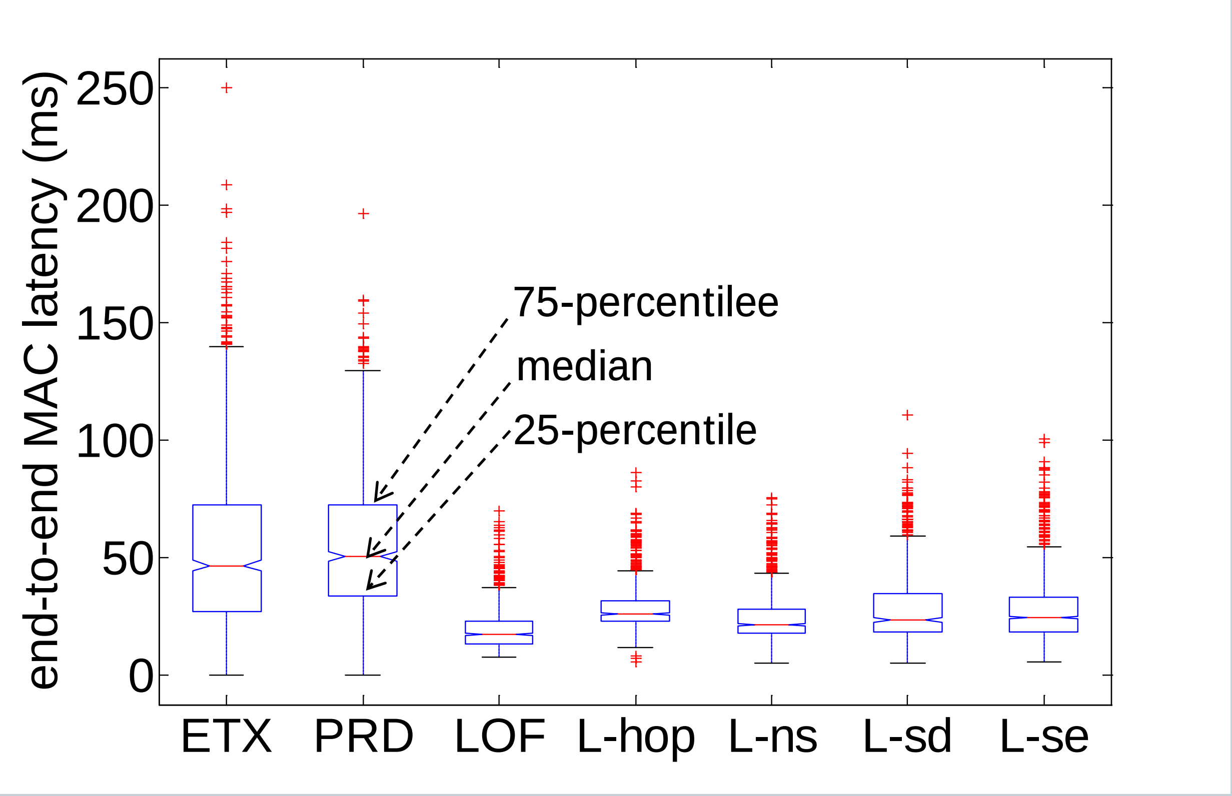
- 802.11b testbed of Kansei
  - 15 × 13 grid
- Traffic flow
  - from the right-bottom corner to the upper-left corn



  - *ExScal* traffic trace
  - 50 runs for each protocol (50 × 19 = 950 packets)
- Evaluation criteria
  - End-to-end MAC latency
  - Energy efficiency
  - Links used in routing

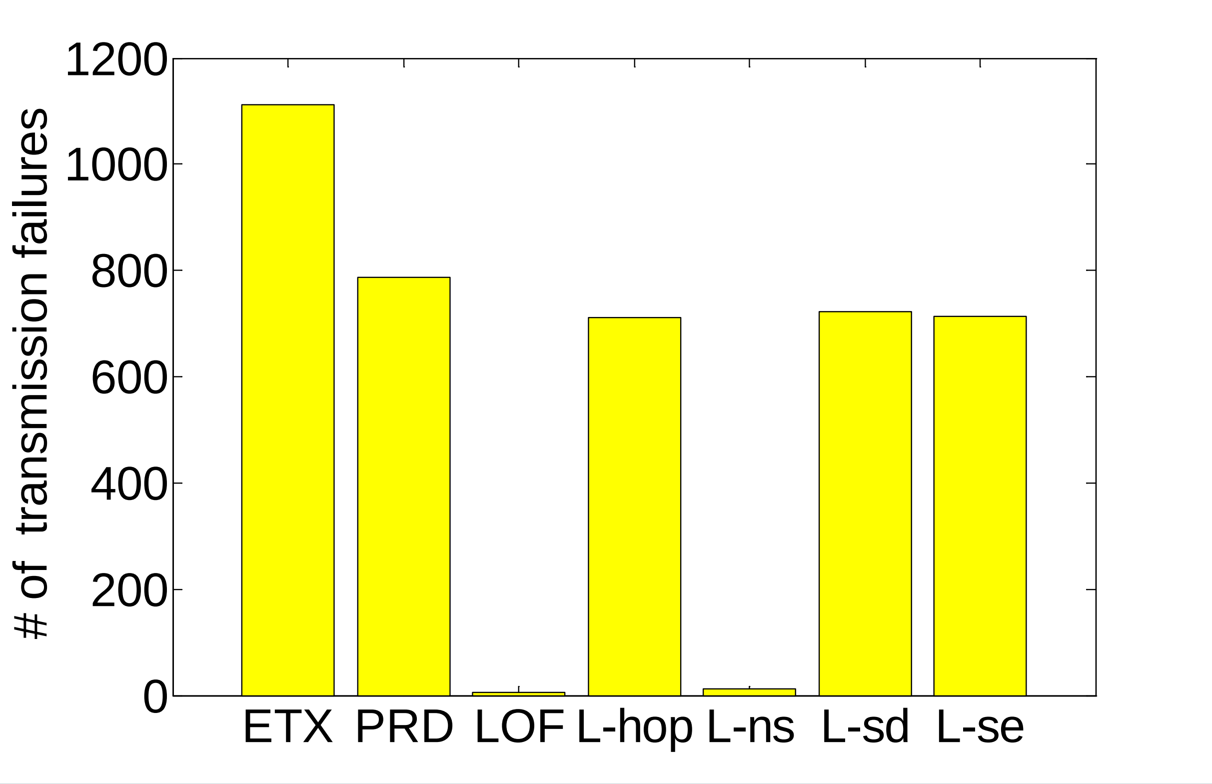
LOF End-to-end MAC latency

LOF reduces MAC latency by a factor of 3

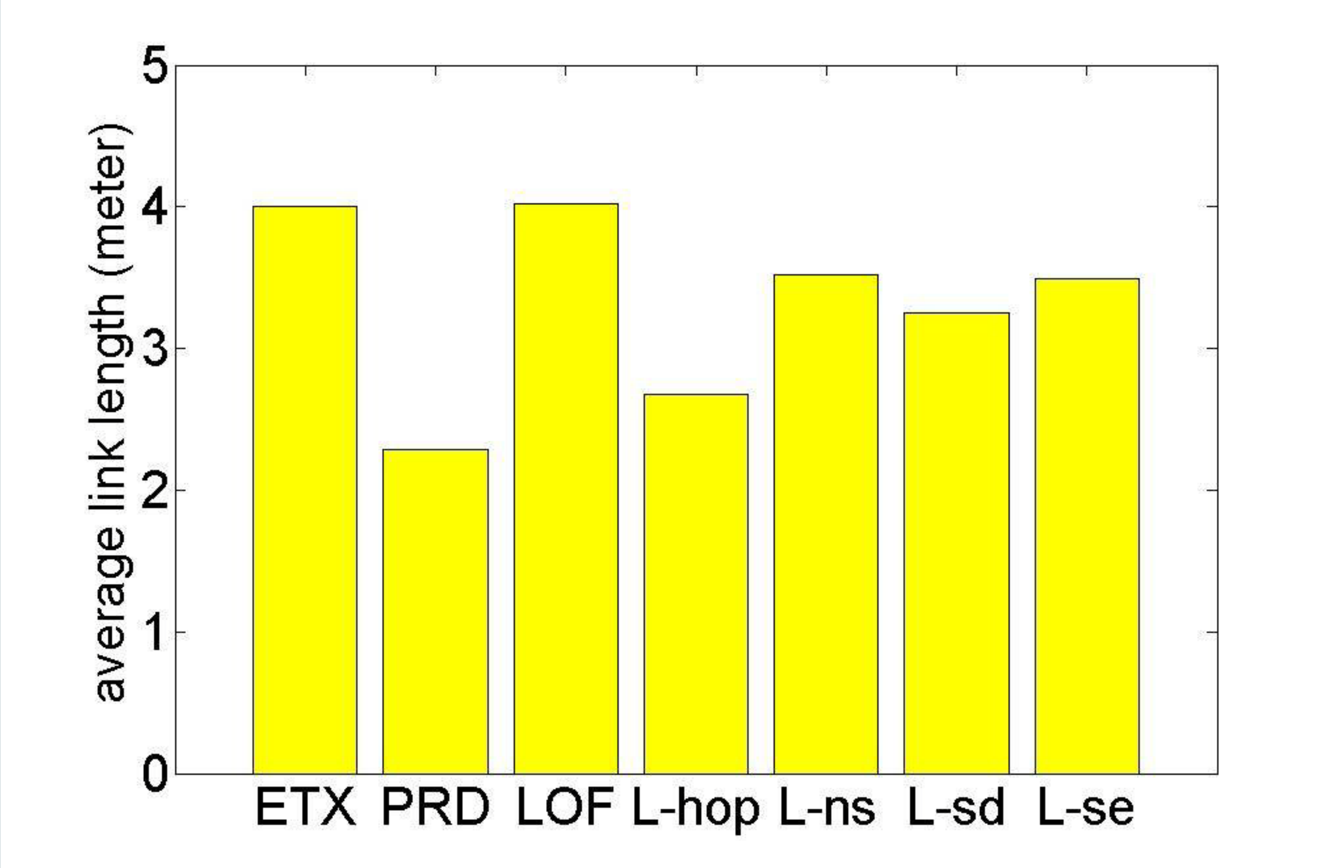


LOF transmission reliability

LOF uses reliable links



LOF path length



Summary

- Demonstrates that beacon based link estimation approach is inefficient
- Proposes to perform link estimation at the data-plane
- Proposes ELD: a new performance metric for routing in sensor networks
- Design of a routing protocol that uses data-plane link estimation and ELD