Autonomous Networking

Gaia Maselli
Dept. of Computer Science

*Some slides in this course are readapted from lecture slides from **Prof. Tommaso Melodia** (Northeastern University, Boston)*

Today's plan

- Routing protocols
  - proactive
  - Reactive
  - Geographic

Routing

- Routing technique is needed for sending the data between the sensor nodes and the base stations, so as to establish **multi-hop** communication
- The applied routing strategy should ensure the **minimum of the energy consumption** and hence **maximization of the lifetime of the network**

Ad Hoc Routing Protocols – Classification

- Network Topology
  - Flat, Hierarchical

- Which data is used to identify nodes?
  - An arbitrary identifier?
  - The position of a node?
    - Can be used to assist in geographical routing protocols because choice of next hop neighbor can be computed based on destination address
    - Scalable and suitable for sensor networks

Flat routing protocols

- Flat Networks Routing Protocols for WSNs can be classified, according to the routing strategy, into three main different categories depending on:
- ***When*** does the routing protocol operate?
- ***Proactive*** Protocols
  - Routing protocol ***always*** tries to keep its routing data up-to-date
  - Active before tables are actually needed
  - Also known as ***table-driven***
- ***Reactive*** Protocols
  - Route is only determined when actually needed
  - Protocol operates ***on demand***
- ***Hybrid*** protocols
  - Combine these behaviors

Flat routing: proactive

- **Pro-active or Table-Driven Routing Protocols** work in a way similar to wired networks.
- Each node maintains routes to all reachable destinations at all times, whether or not there is currently any need to deliver packets to those destinations
- They also respond to any changes in network topology by sending updates throughout the wireless network and thus maintaining a consistent network view.

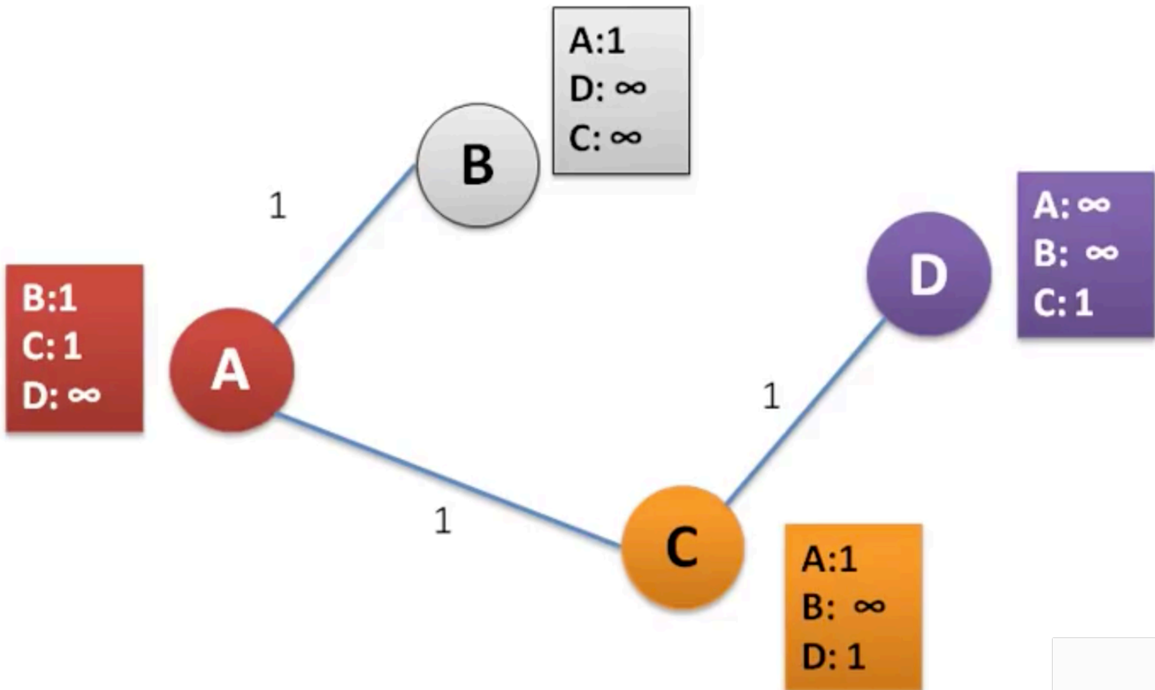
↑ When a path to some destination is needed the route is already known and there is no extra delay due to route discovery.

↓ Keeping the information up-to-date may require a lot of bandwidth and extra battery power. The information may still be out-of-date.

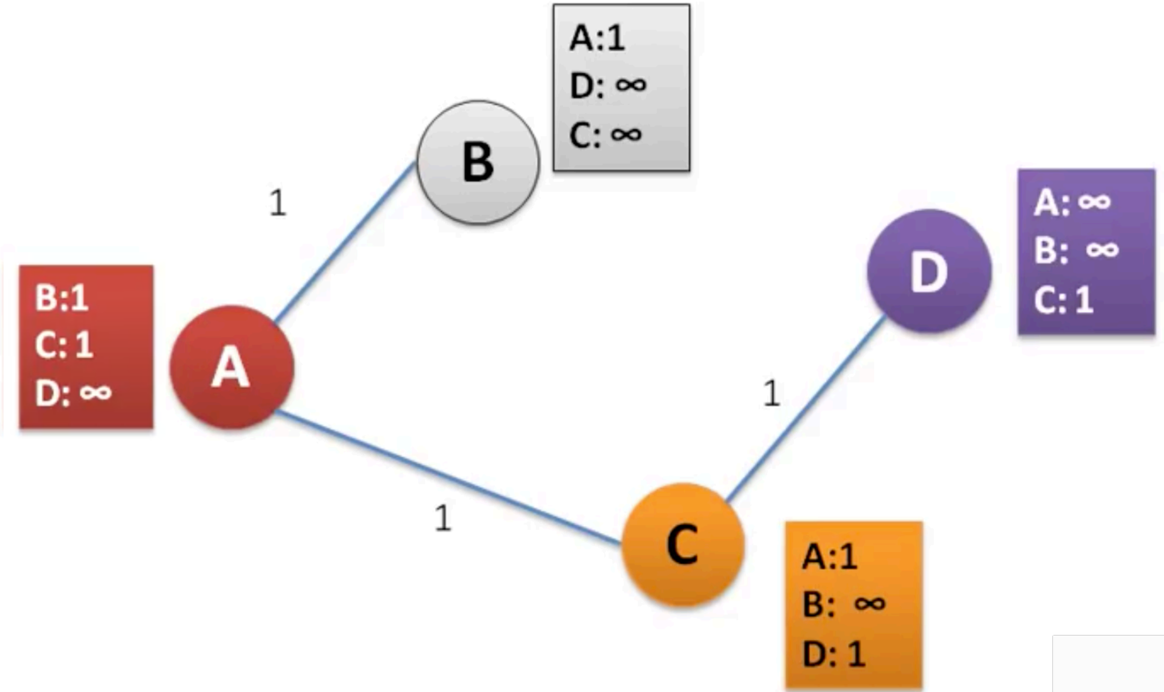
Destination Sequence Distance Vector (DSDV)

- **Adapted DISTANCE VECTOR:**
  Destination Sequence Distance Vector (DSDV)
  - Based on distributed Bellman Ford procedure
  - **Add aging information** to route information propagated by distance vector exchanges; helps to avoid routing loops
  - Periodically send full route updates
  - On topology change, send incremental route updates
  - Unstable route updates are delayed

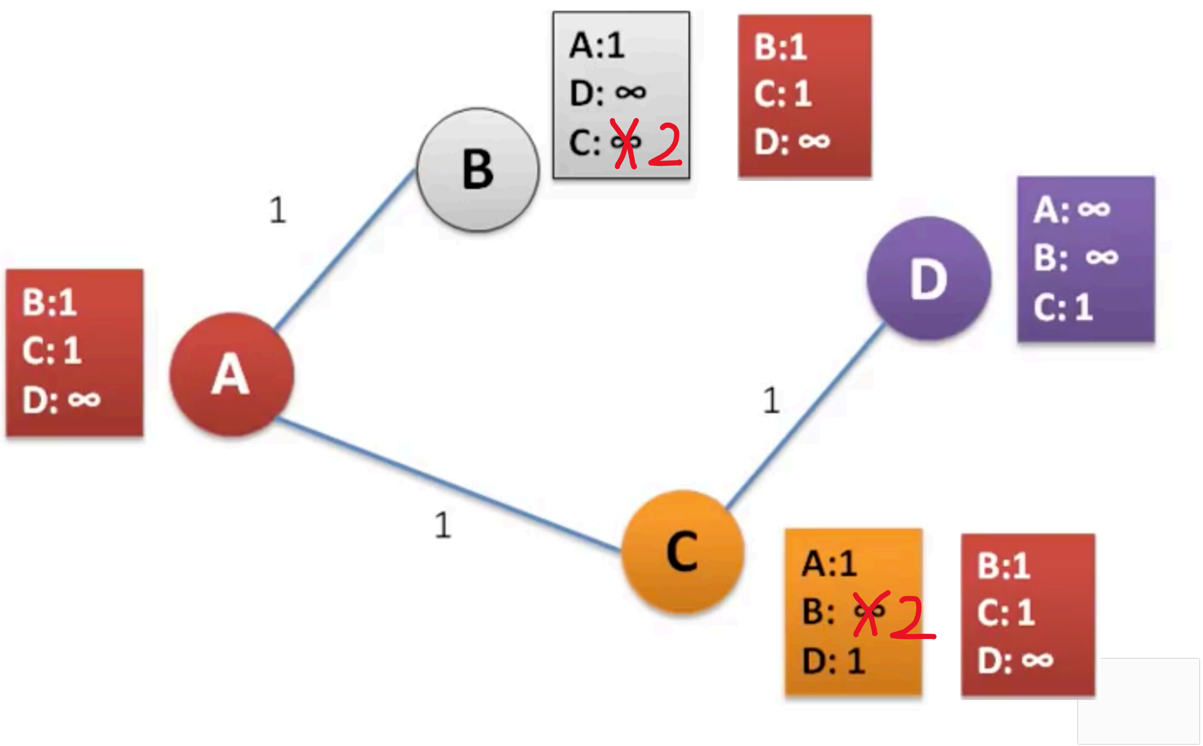
DSDV



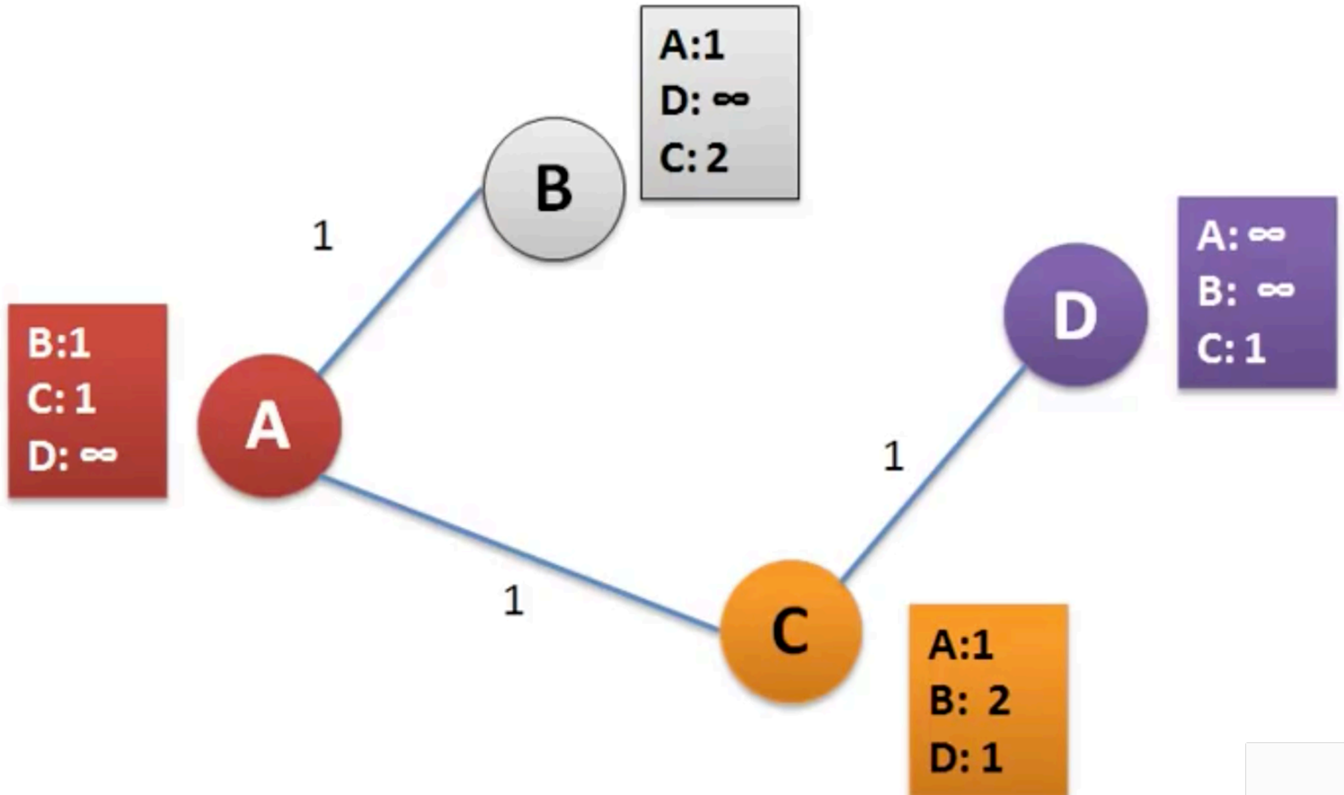
DSDV: distance vector



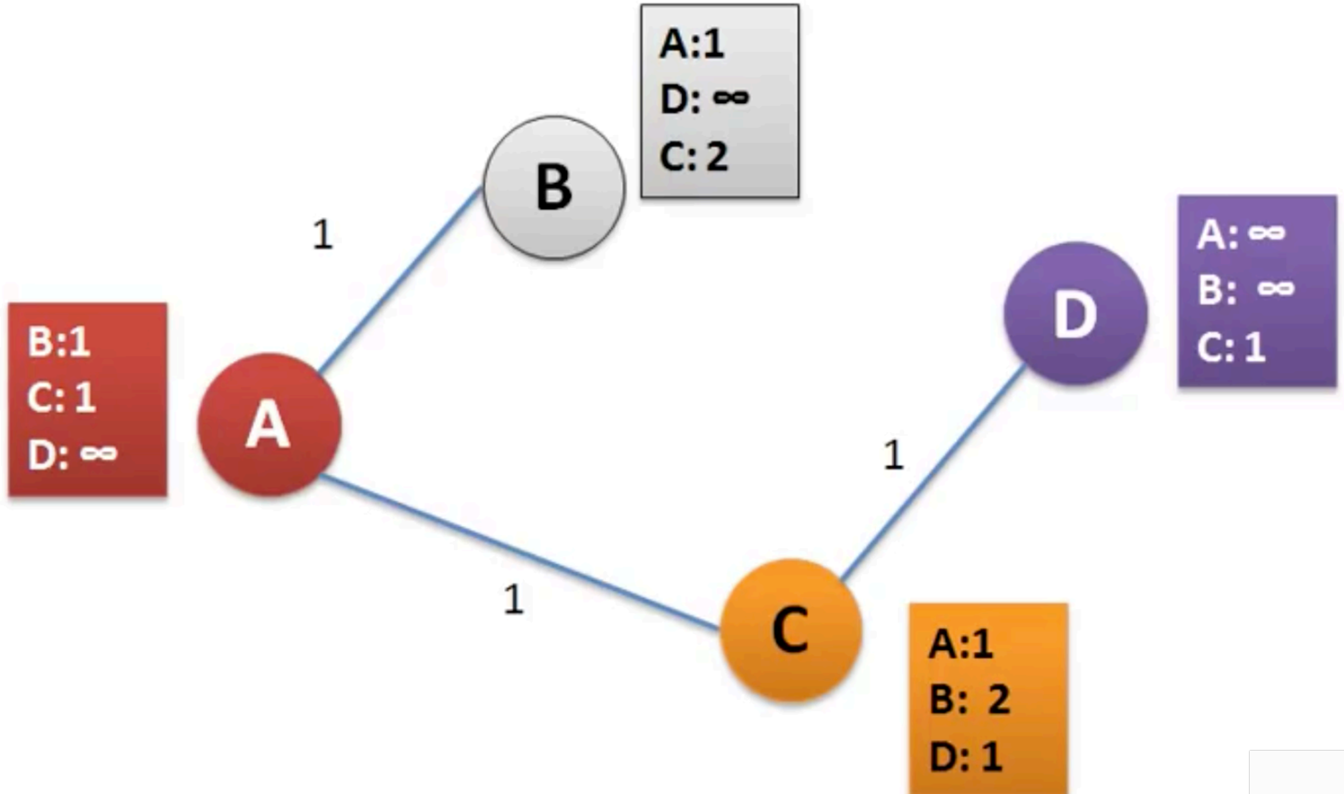
DSDV: distance vector update



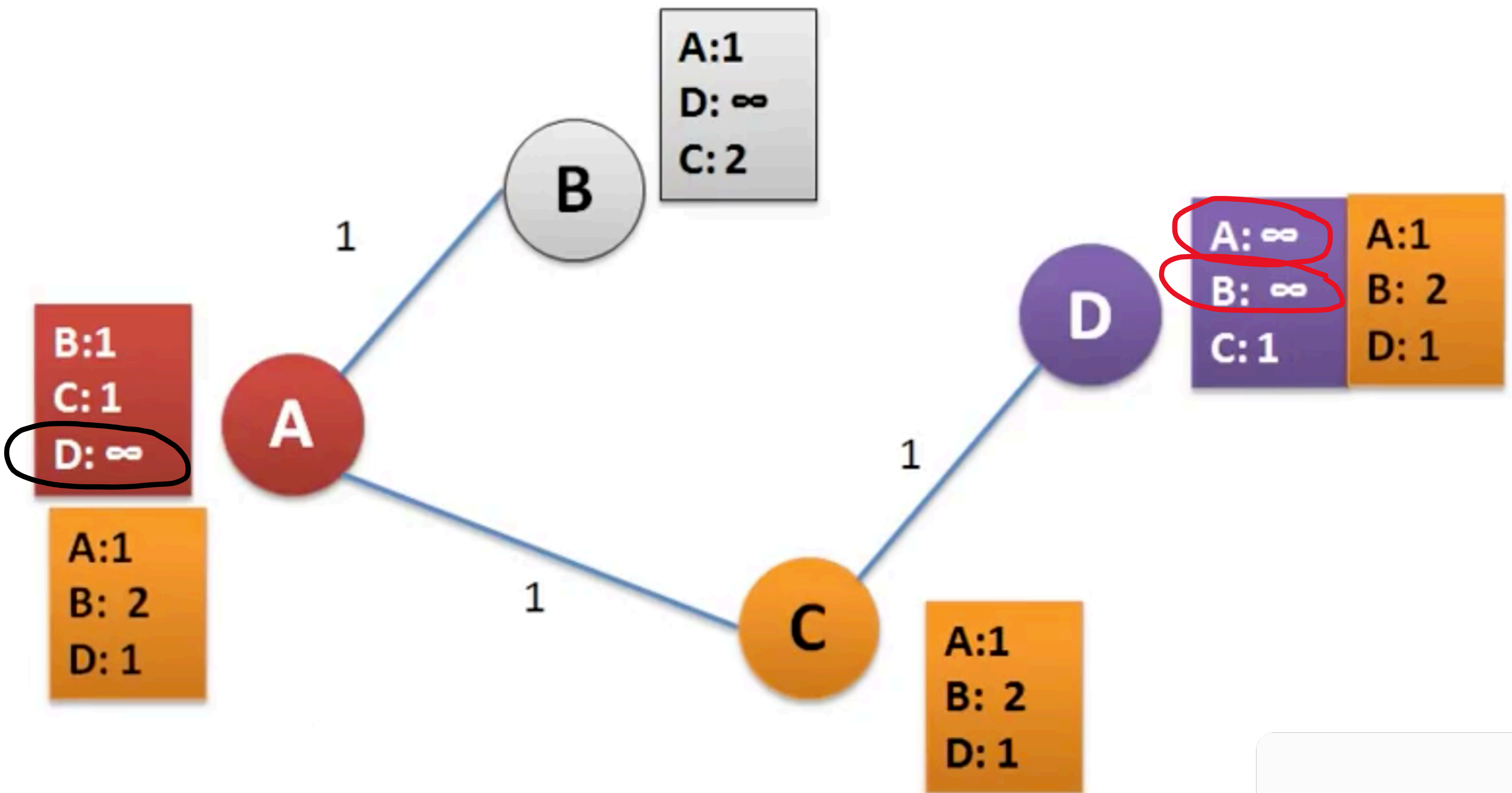
DSDV: distance vector update



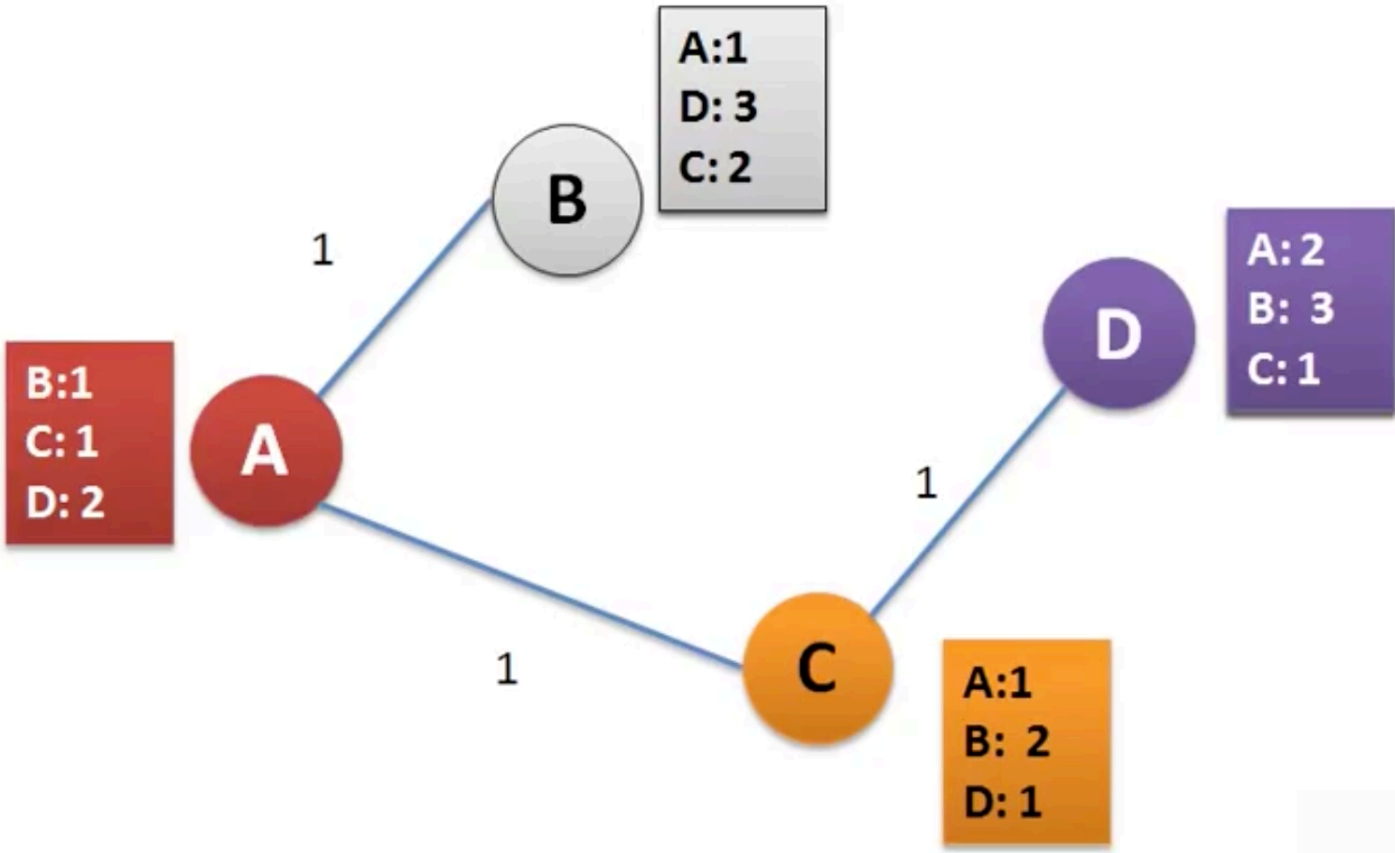
DSDV: distance vector update



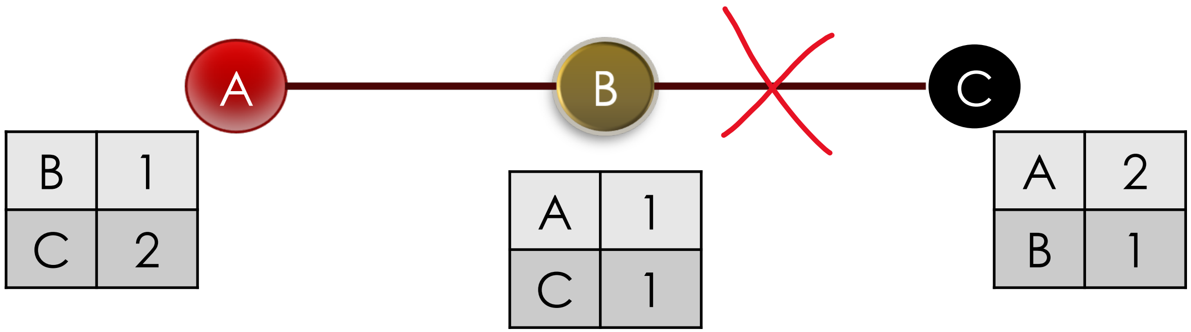
DSDV: distance vector update



DSDV: distance vector update



DSDV: Rooting loop

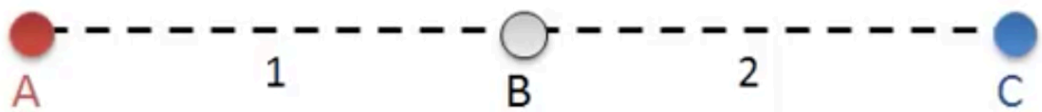


- Rooting loops
  - (B,C) breaks
  - A advertises before B that it can reach C with cost 2
  - A understands that it can reach C with cost 3
- Any data packet form A to C would loop between A and B

DSDV: destination sequence approach

- **Tag each** route table **entry with a sequence number** so that nodes can quickly distinguish stale routes from the new ones and thus avoid formation of routing loops.
- Routing table lists all available destinations and the number of hop to each
- Each route table entry is tagged by a sequence number which is originated by the destination station (the transmitter)
- Routes with more recent frequent numbers are always preferred
- When a link to a next hop has broken, any route through that next hop is immediately assigned an ∞ metric and assigned an updated sequence number

DSDV: distance sequence



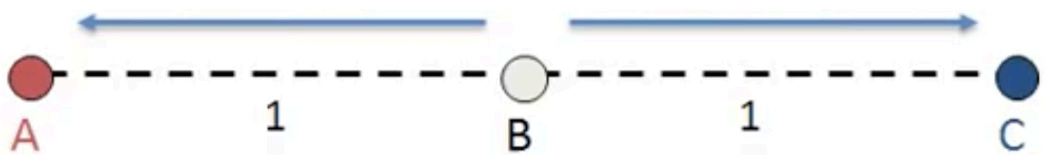
Dest.	Next	Metric	Seq
A	A	0	A-550
B	B	1	B-100
C	B	3	C-588

Dest.	Next	Metric	Seq
A	A	1	A-550
B	B	0	B-100
C	C	2	C-588

Dest.	Next	Metric	Seq.
A	B	1	A-550
B	B	2	B-100
C	C	0	C-588

DSDV: distance sequence

- B increases sequence number (100 → 102) before broadcasting its distance vector
- A and C understand that B's information is updated

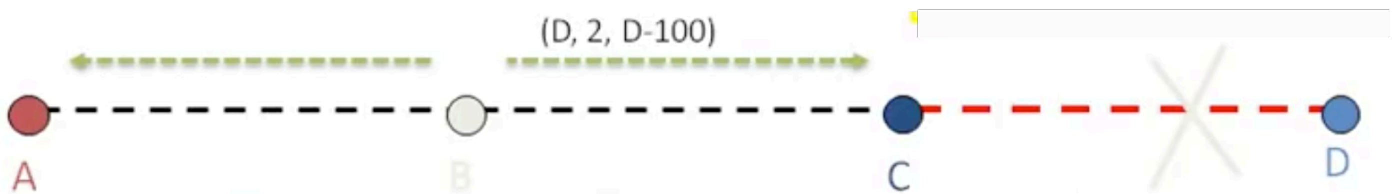


Dest.	Next	Metric	Seq
A	A	0	A-550
B	B	1	B-102
C	B	2	C-588

Dest.	Next	Metric	Seq
A	A	1	A-550
B	B	0	B-102
C	C	1	C-588

Dest.	Next	Metric	Seq.
A	B	2	A-550
B	B	1	B-102
C	C	0	C-588

DSDV: distance sequence



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

- When node C detects broken link (C,D)
  - Node C update its metric to ∞ and sequence number (to a odd number). Note this is the only case a node different from the destination can update it seq.
  - C sends its distance vector
  - If node B sends its distance vector, C does not update the entry of D because Seq nr is less than the Seq nr it stores (for D)

Flat routing: reactive

- **Re-active or Source-Initiated On-Demand Routing Protocols** only start a route discovery procedure when needed
- When a route from a source to a destination is needed, a kind of global search procedure is started. This task does not request the constant updates to be sent through the network, as in pro-active protocols, but this process does cause delays, since the requested routes are not available and have to be found.
- In some cases, the desired routes are still in the route cache maintained by the sensor nodes. When this is the case, there is no additional delay since routes do not have to be discovered.
- The whole process is completed as soon as a route is found or all possible route combinations have been examined.

Reactive protocols

- Flooding
- Gossiping
- Dynamic Source Routing (DSR)
- Ad hoc On Demand Distance Vector (AODV)

Flooding

- Flooding is an old and very simple technique which can be also used for routing in WSNs
- Copies of incoming packets are sent by every link except the one by which the packets arrived.
- This procedure generates an enormous amount of superfluous traffic.
- Flooding is a reactive technique, and does not require costly topology maintenance and complex route discovery algorithms.

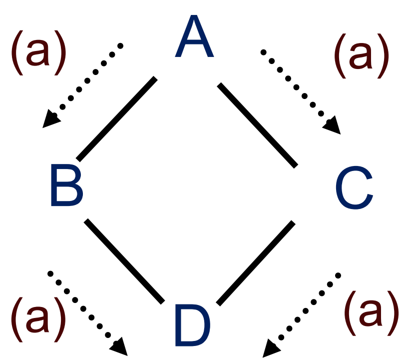
Flooding: characteristics

- Flooding has two interesting characteristics which arise from the fact that all possible routes are tried

1. As long as there is a route from source to destination, the delivery of the packet is guaranteed.
2. One copy of the packet will arrive by the quickest possible route

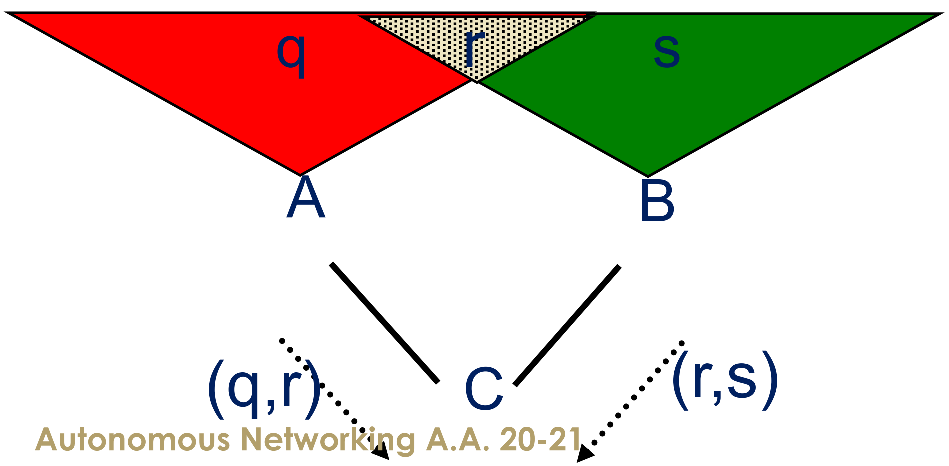
Flooding: drawbacks

Implosion



- Implosion: a situation where duplicated messages are broadcasted to the same node

Data Overlap



- Overlap: If two nodes share the same under observation region, both of them may sense the same stimuli at the same time. As a result, neighbor nodes receive duplicated messages

Flooding: drawbacks

- **Resource blindness** (No knowledge about the available power of resources)
- The flooding protocol does not take into consideration all the available energy resources. An energy resource-aware protocol must take into account the amount of energy which is available to them all the time.
- Flooding consumes much energy, as for each data packet, all the nodes that are in the broadcast domain will receive packets that they will forward it to their neighbors. Thus, they require a large amount of power that causes a prohibitively short network lifetime.

Gossiping

- In gossiping, every node in the network knows a unique item of information and needs to communicate it to everyone else
- Nodes send the incoming packets to a randomly selected neighbor.
- Although this approach avoids the implosion problem by just having one copy of a message at any node, it takes long to propagate the message to all sensor nodes in the network.

Dynamic Source Routing (DSR) – RFC 4728

- **Source routing**: Each data packet sent carries in its header the complete, ordered list of nodes through which the packet will pass
- The sender can select and control the routes used for its own packets and supports the use of multiple routes to any destination
- Including source route in the header of each data packet, other nodes forwarding or overhearing any of these packets can also easily cache this routing information for future use

Dynamic Source Routing (DSR)

- The DSR protocol is composed of two main mechanisms that work **(on demand)** together to allow the discovery and maintenance of source routes
- **Route Discovery** is the mechanism by which a node S wishing to send a packet to a destination node D obtains a source route to D
- Route Discovery is used only when S attempts to send a packet to D and does not already know a route to D
- **Route Maintenance** is the mechanism by which node S is able to detect, while using a source route to D, if the network topology has changed such that it can no longer use its route to D because a link along the route no longer works.
- When Route Maintenance indicates a source route is broken, S can attempt to use any other route it happens to know to D, or it can invoke Route Discovery again to find a new route for subsequent packets to D. Route Maintenance for this route is used only when S is actually sending packets to D.

DSR: route request

- Each Route Request contains
  - a unique request identification
  - a record listing the address of each intermediate node through which this particular copy of the Route Request has been forwarded.
- This route record is initialized to an empty list by the initiator of the Route Discovery
- When another node receives this Route Request, it appends its own address to the route record in the Route Request and propagates it by transmitting it as a local broadcast packet (with the same request identification)

```
   ^    "A"     ^    "A,B"    ^   "A,B,C"   ^  "A,B,C,D"
   |   id=2     |    id=2     |    id=2     |    id=2
+-----+      +-----+      +-----+      +-----+      +-----+
|  A  |----->|  B  |----->|  C  |----->|  D  |----->|  E  |
+-----+      +-----+      +-----+      +-----+      +-----+
   |            |            |            |
   v            v            v            v
```

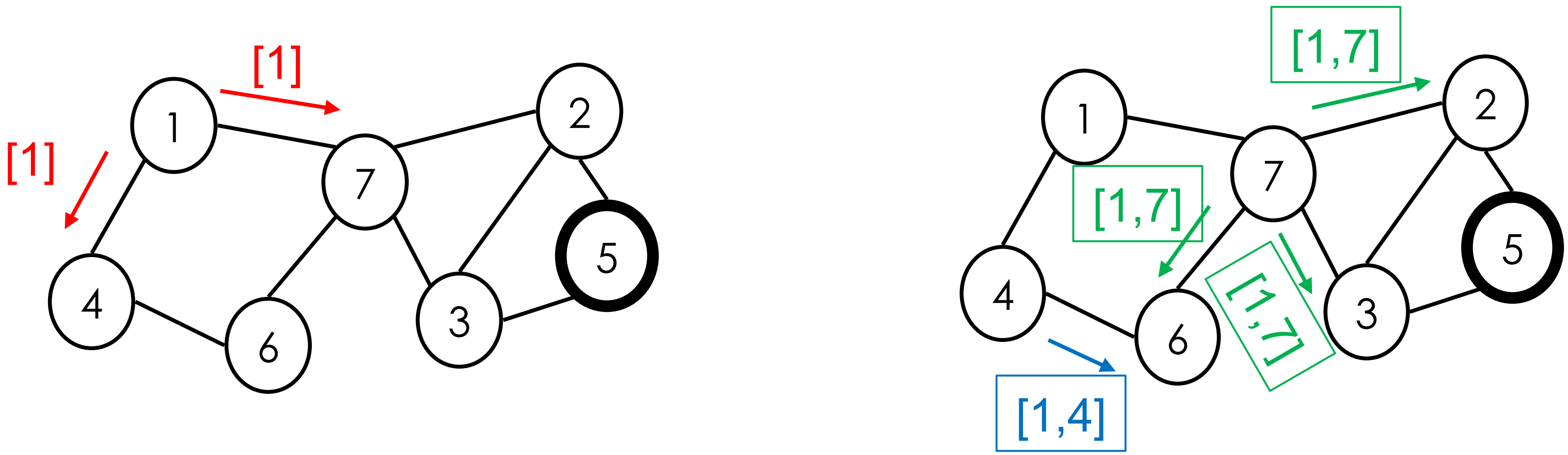
DSR: route reply

```
      ^    "A"     ^    "A,B"    ^   "A,B,C"  ^  "A,B,C,D"
      |   id=2     |    id=2     |    id=2    |    id=2
   +-----+      +-----+      +-----+      +-----+      +-----+
   |  A  |----->|  B  |----->|  C  |----->|  D  |----->|  E  |
   +-----+      +-----+      +-----+      +-----+      +-----+
      |            |            |            |
      v            v            v            v
```

- When a node receives a Route Request (such as node E in this example), if it is the target of the Route Discovery, it returns a "Route Reply" to the initiator of the Route Discovery, giving a copy of the accumulated route record from the Route Request
- When the initiator receives this Route Reply, it caches this route in its Route Cache for use in sending subsequent packets to this destination.

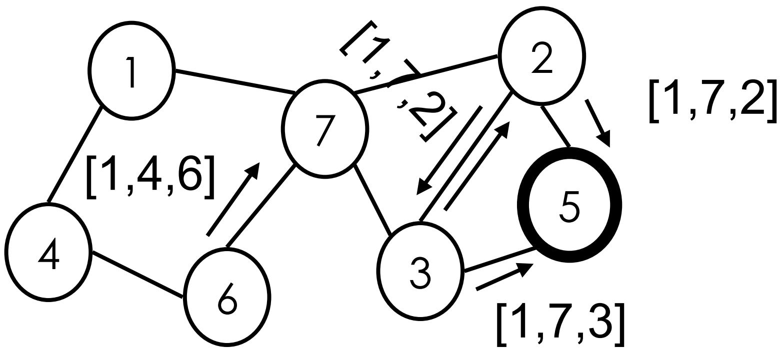
DSR Route Discovery Procedure

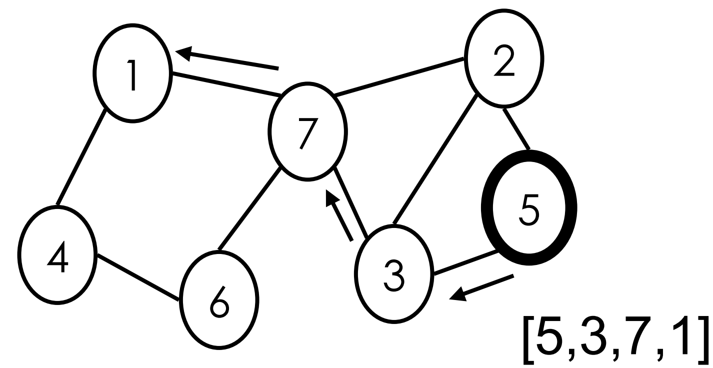
- Search for route from 1 to 5



- Each Route Request contains
  - a unique request identification
  - a record listing the address of each intermediate node through which this particular copy of the Route Request has been forwarded.

DSR Route Discovery Procedure





- Node 5 uses route information recorded in RREQ to send back, via ***source routing***, a route reply [5,3,7,1]

DSR: Multiple routes

- In response to a single Route Discovery a node may learn and cache multiple routes to any destination.
- This support for multiple routes allows the **reaction to routing changes to be much more rapid**, since a node with multiple routes to a destination can try another cached route if the one it has been using should fail.

Reactive Protocols – AODV

- **Ad hoc On Demand Distance Vector routing (AODV)**
  - Very popular routing protocol
  - Essentially same basic idea as DSR for discovery procedure
  - Nodes maintain routing tables instead of source routing
  - Sequence numbers added to handle stale caches
  - Nodes remember from where a packet came and populate routing tables with that information

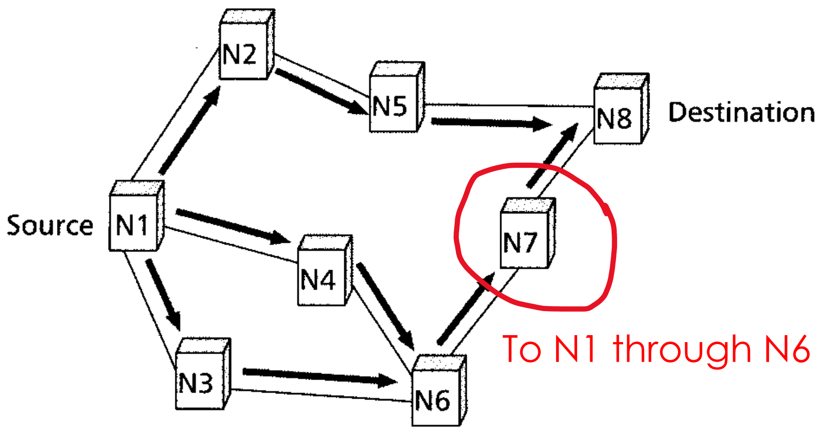
AODV

- AODV is an improvement on DSDV because it typically minimizes the number of required broadcasts by creating routes **on** a **demand** basis, as opposed to maintaining a complete list of routes as in the DSDV algorithm
- When a source node desires to send a message to some destination node and does not already have a valid route to that destination, it initiates a ***path discovery*** process to locate the other node.
- Source node broadcasts a route request (RREQ) packet to its neighbors, which then forward the request to their neighbors, and so on, until either the destination or an intermediate node with a “fresh enough” route to the destination is located.

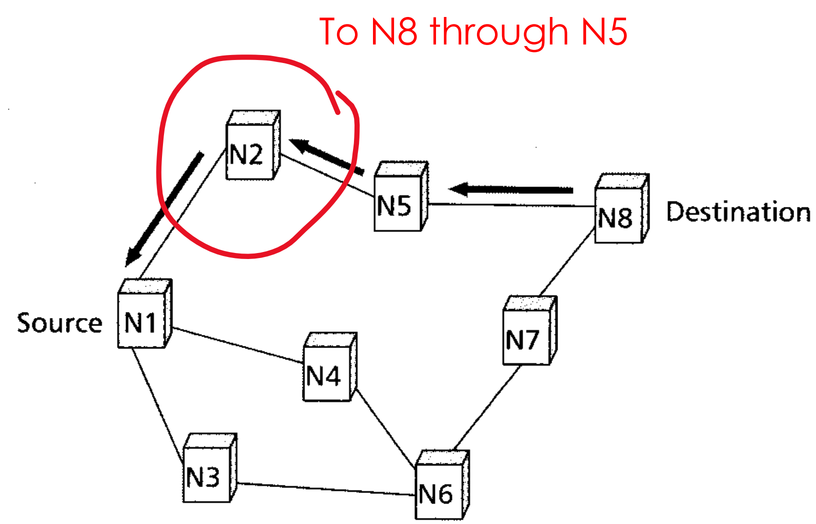
AODV

- During the process of forwarding the RREQ, intermediate nodes record in their route tables the address of the neighbor from which the first copy of the broadcast packet is received, thereby establishing a reverse path.
- Once the RREQ reaches the destination or an intermediate node with a fresh enough route, the destinationiintermediate node responds by unicasting a route reply (RREP) packet back to the neighbor from which it first received the RREQ
- As the RREP is routed back along the reverse path, nodes along this path set up forward route entries in their route tables which point to the node from which the RREP came.

AODV



(a) Propagation of the RREQ



(b) Path of the RREP to the source

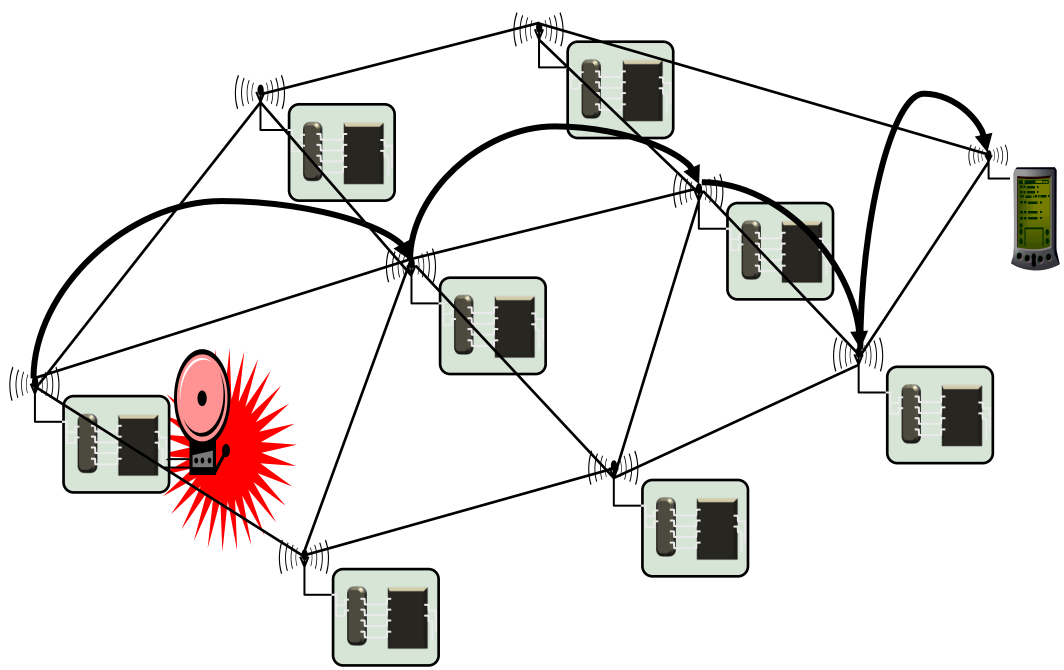
- Intermediate nodes learn reverse paths
- Intermediate nodes learn forward paths

Geographic Routing

- Routing tables contain information to which next hop a packet should be forwarded
  - Explicitly constructed
- Alternative: **Implicitly infer this information from physical placement of nodes**
  - Position of current node, current neighbors, destination known – send to a neighbor in the right direction as next hop
  - **Geographic routing** (also **position-based routing**)
- Options
  - Send to any node in a given area – geocasting
  - Use position information to aid in routing – position-based routing
    - Might need a location service to map node ID to node position

Basics of Position-based Routing

- "**Most forward within range r**" strategy
  - Send to that neighbor that realizes the most forward progress towards destination
  - NOT: farthest away from sender!

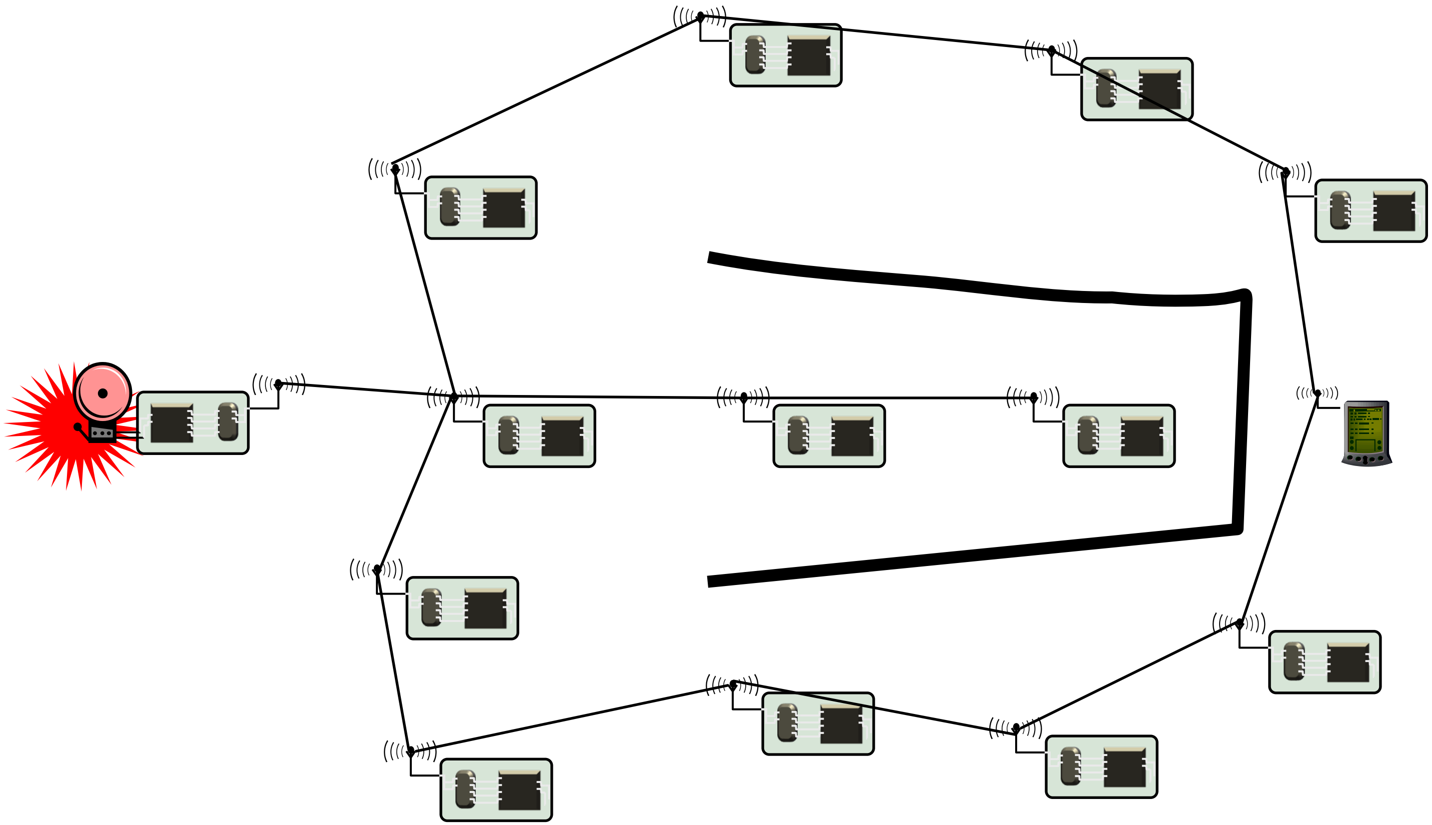


Alternative Strategies

- **Nearest node with (any) forward progress**
  - Idea: Minimize transmission power
- **Directional routing**
  - Choose next hop that is angularly closest to destination
  - Choose next hop that is closest to the connecting line to destination
  - Problem: Might result in loops!

Problem: Dead ends

- Simple strategies might send a packet into a dead end or void



Routing in WSN

- There are many other protocols
- What is the best solution?
- Depends on the network characteristics
  - Mobility
  - Node capabilites
- Geographic approach allows to save more energy
- Proactive approach is fast but involves overhead
- Reactive approach generate much less overhead but it is slower

What if

- The network topology is continuoulsy changing..

- Consider a network of drones flying over an area
- Which approach would be best?