Internet Of Things 2017-2018

UAVNET - Unmanned Aerial Vehicles Network

Domenicomichele Silvestri – Andrea Coletta

DIPARTIMENTO
DI INFORMATICA

SAPIENZA
UNIVERSITÀ DI ROMA

What we will see today?

- What is a UAVNET
- Common issues and challenges
- Routing protocols
- Energy Constraints

What is a UAVNET

- **Unnamed Aerial Vehicles Network: «***Ad-hoc or infrastructure based network of airborne nodes***»**

UAV, commonly known as a **Drone**, is an aircraft without a human pilot aboard.
May operate with various degrees of autonomy: either under **remote control** by a human operator or **autonomously** by on board computers.

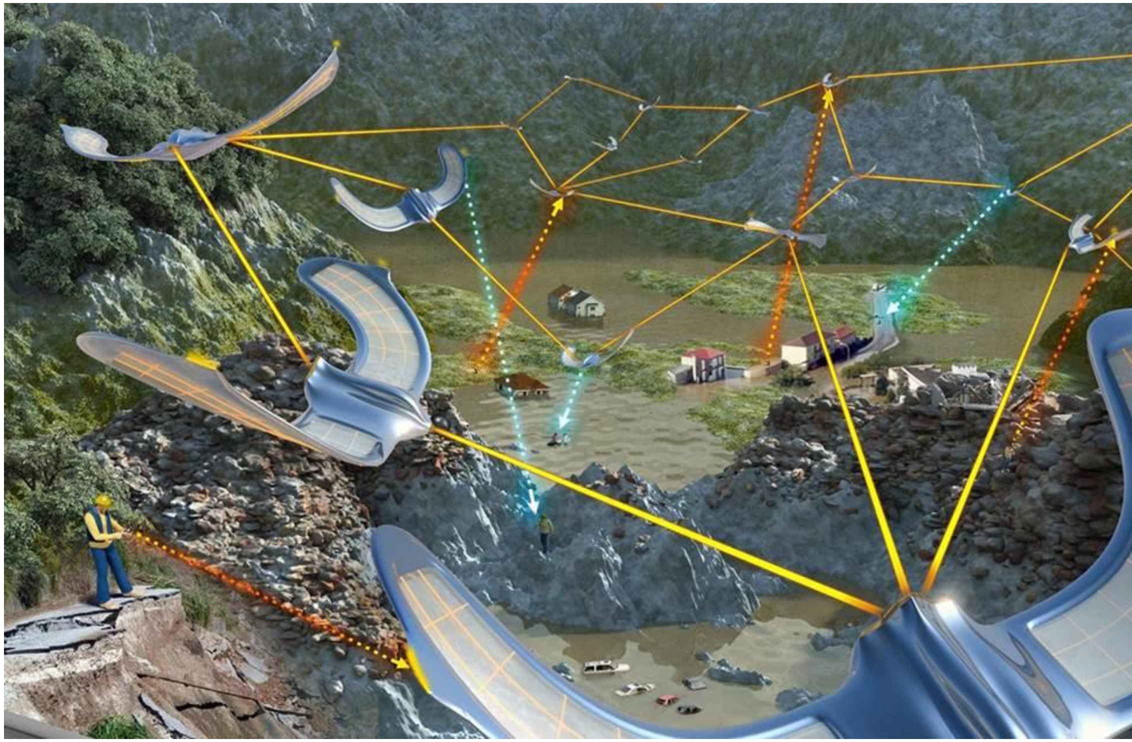
Why UAVs?

- Can provide timely **disaster warnings and assist** in speeding up rescue and recovery operations.
- Can **carry medical supplies** to areas rendered inaccessible.
- Can be used in **dangerous situations.**
- Can be used in common applications as **traffic monitoring**, **wind estimation** and **remote sensing**.
- And so on…



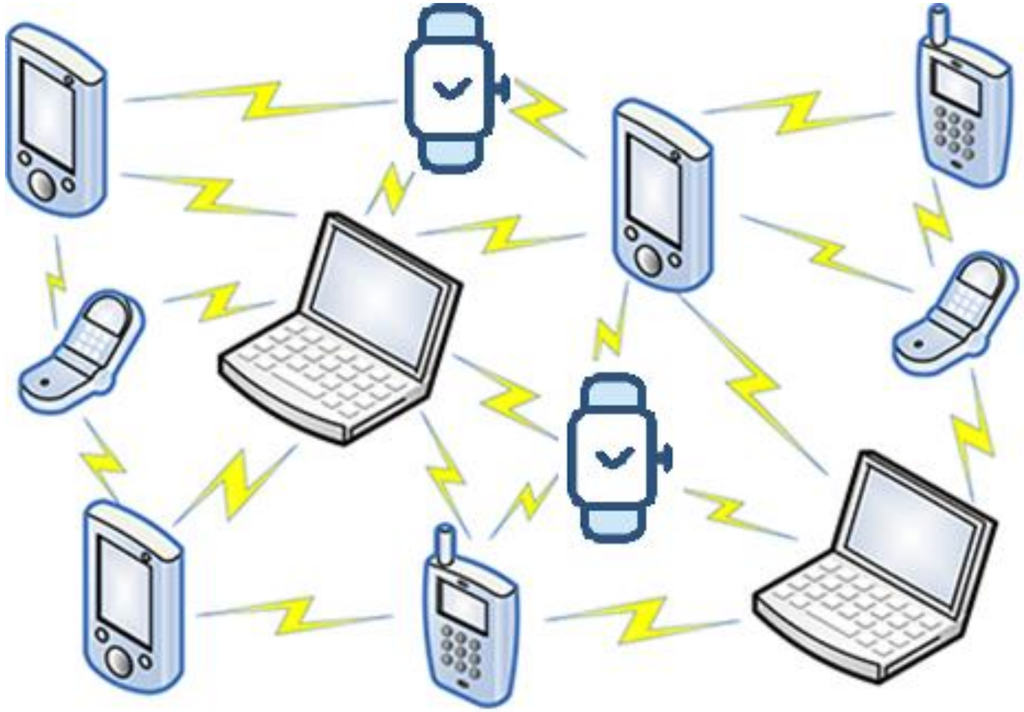

UAVNET Issues and Challenges

- **Fluid Topology**
- **Node Failure**
- **High mobility**
- **Routing**
- **Energy constraints**



UAVNET, MANET and VANET (1)

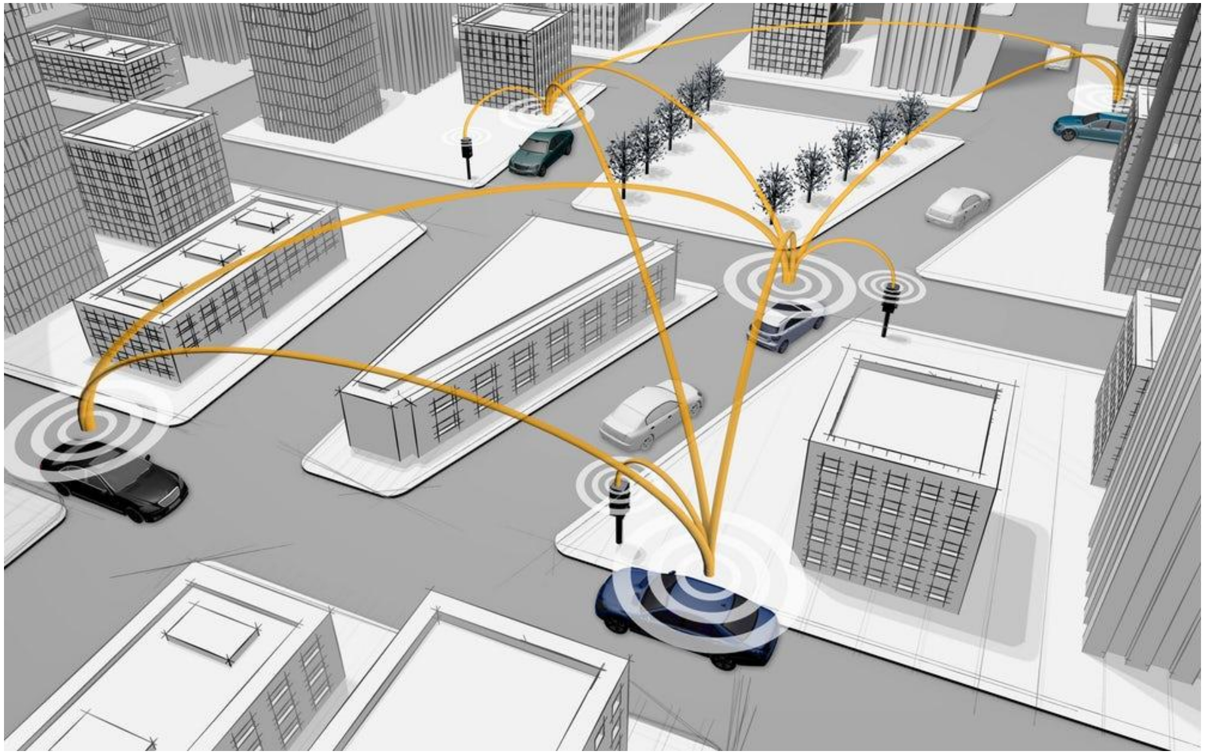
MANET– Mobile Ad hoc Network.



- Full decentralised network
- Nodes and devices cans be mobile
- No fixed infrastructure

UAVNET, MANET and VANET (2)

VANET– Vehicular Ad hoc networks.

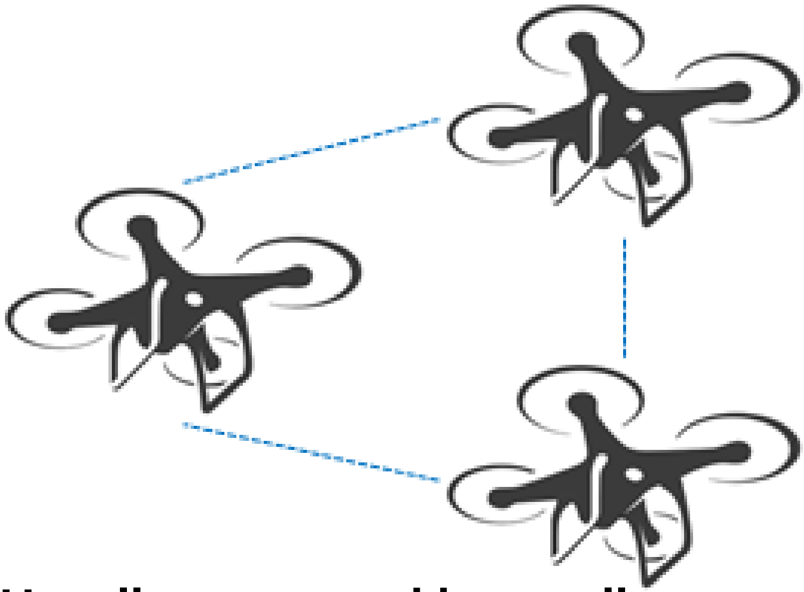


- Decentralised network or star network
- Nodes are mobile
- Can be present fixed infrastructures

	MANET	VANET	UAVNET
Mobility	Slow. Random movement.	Medium speed. Follow road.	Low to high speed 2 or 3 dimensions.
Topology	Random, ad-hoc	Star or ad-hoc	Star or ad-hoc
Topology change	Dynamic, nodes join/fail, Possible partitioning	Dynamic. Movement linear. Possible partitioning	Stationary or dynamic Possible partitioning
Energy constraints	Battery powered.	Powered by car or battery.	Battery powered in small UAVs.
Typical use	Information distribution, internet delivery, …	Traffic info, emergency, infotainment, …	Rescue, sensing, Surveillance, …

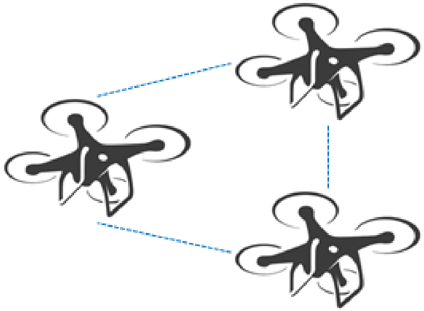
Single-UAV vs Multi-UAVs

Usually used in military operations.
First usage in Vietnam War.



Usually composed by small civilian UAVs, their use is increasing.

Single-UAV vs Multi-UAVs

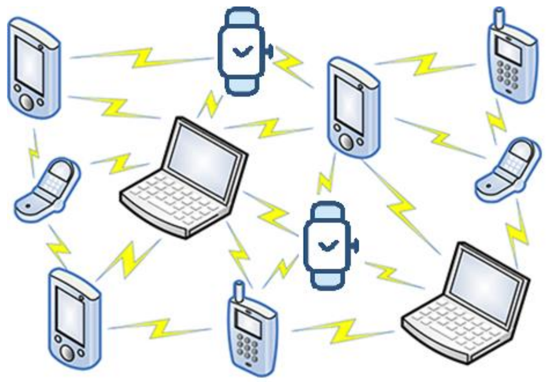


	Feature	
High	**Impact of failure**	Low
Medium	**Cost**	Low
Limited	**Scalability**	High
Poor	**Survivability**	High
Slow	**Speed of mission**	Fast
High	**Bandwidth required**	Medium
Omni-directional	**Antenna**	Directional
Low	**Complexity of Control**	High
Low	**Failure to Coordinate**	Present

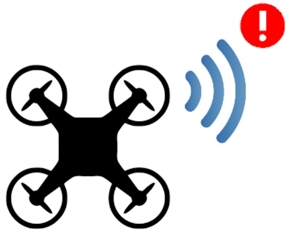
Features of the UAV Networks (1)

- ***Infrastructure-Based or Ad Hoc?***

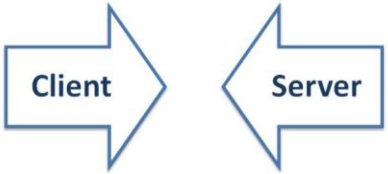
Or



- ***Delay and Disruptions Prone Networks***

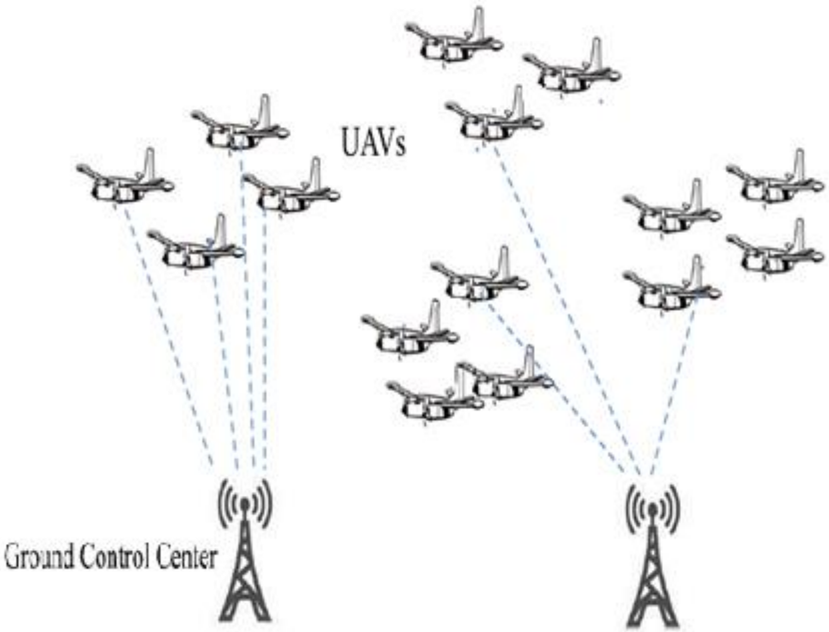


- ***Server or Client***

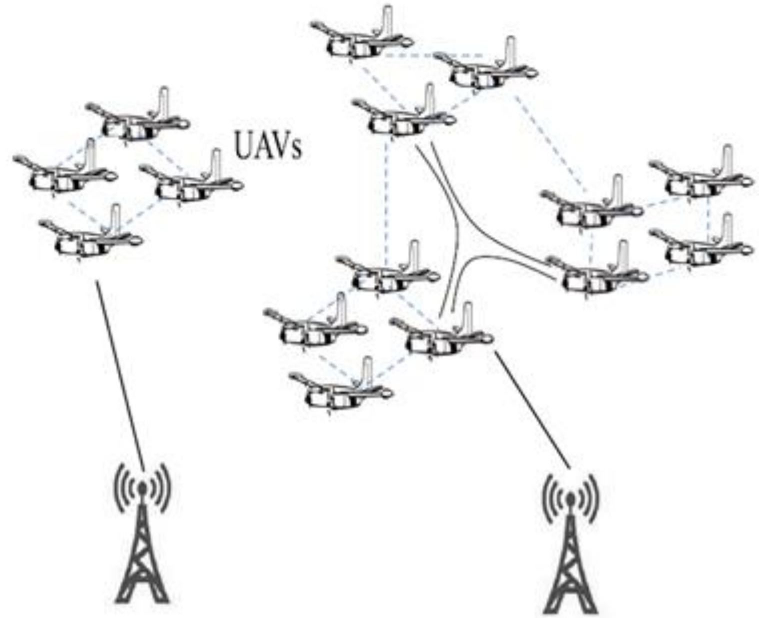


Features of the UAV Networks (2)

- ***Star or Mesh?***



Start Configuration Multi-start Configuration



Flat Mesh Network Hierarchical Mesh Network.

Categorization of UAV Networks (1)

- **Internet Delivery,** disaster communication, oil exploration, remote health, …

- **Sensing,** reconnaissance, search, detecting fires, tracking, …

- **Attack**, war and active defence.

Categorization of UAV Networks (2)

- **Internet Delivery**

  Position: Fixed Position

  Mobility speed: ~0 miles/hour

  Network: Infrastructure based (base can be in sky)

- **Sensing**

  Position: Slow change, coordinated movement

  Mobility speed: <10 miles/hour

  Network: Infrastructure based

- **Attack**

  Position: Frequent change

  Mobility speed: >10 miles/hour

  Network: Infrastructure based, Ad hoc

Categorization of UAV Networks (3)

- **Internet Delivery**

  Control (communication): Centralized (position based)

  Server/Client: Server

  Routing: Through server

- **Sensing**

  Control (communication): Centralized (task based)

  Server/Client: both

  Routing: Central or mesh

- **Attack**

  Control (communication): Distributed, individuals

  Server/Client: either

  Routing: Mesh

Categorization of UAV Networks (4)

- **Internet Delivery**

  Delay: Low probability ($p < 0.1$)

  Control (path, position): Remote – position control

- **Sensing**

  Delay: Medium probability ($0{,}1 \leq p < 0.5$)

  Control (path, position): remote – position/path control

- **Attack**

  Delay: High probability ($p \geq 0.5$)

  Control (path, position): remote – auto – path control

Routing in UAVNET (1)

Routing Issues and Challenges:

of **common wireless network**:

- finding the most efficient route
- allowing the network to scale
- controlling latency
- ensuring reliability
- taking care of mobility and ensuring required QoS

AND

of **airborne networks**:

- requires location-awareness
- energy awareness
- robustness to intermittent links
- robustness to changing topology

Routing in UAVNET (2)

Common approaches in Literature:

- Use of existing routing protocols.
- Optimize existing routing protocols.
- Build ad hoc routing protocols.

BUT:

- there still exists a need for a routing protocol tailored to the particular needs of airborne networks.

Routing in UAVNET (3)

Goals of Routing protocol:

Routing Protocols

1. **Static protocols**: *have static routing tables computed at the task starts and cannot be updated.*

2. **Proactive protocols:** *use tables in their nodes to store all the routing information, they are updated when topology changes.*

3. **Reactive protocols:** a route is stored when there is need of communication between nodes (on-demand).

4. **Hybrid protocols:** try to reduce overhead of protocol mixing proactive and reactive approaches.

5. **Geographic Protocols:** *a routing scheme based on the geographical position of the nodes.*

Static Protocols

Characteristics:

- fixed routing tables computed when task starts
- routing tables not updated
- not fault tolerant
- not suitable for dynamic topology
- not scalable

Applicability to UAV networks is limited.

Protocols:

- **LCAD** - *Load Carry and Deliver Routing*
- **MLHR** - *Multi Level Hierarchical Routing*
- **DCR** - *Data Centric Routing*

Static Protocols: LCAD - *Load Carry and Deliver Routing*

Ground node passes the data to a UAV, which carries it to the destination.

GOAL: maximize the throughput while increasing security.

Single UAV:

- long data delivery delays
- higher throughput

LCAD can scale its throughput by using multiple UAVs to relay the information to multiple destinations.

Useful only for delay tolerant and latency insensitive data transfer.

Not very useful for UAVNET.

Static Protocols: MLHR - Multi Level Hierarchical Routing

Vehicular networks are organized as flat structures because of which performance degrades when the size increases.

GOAL: solve scalability problems in large-scale Vehicular networks.

Organizing the network as hierarchical structure increases size and operation area.

UAV networks grouped into clusters where only the cluster head has connections outside cluster.
The cluster head disseminates data by broadcasting to other nodes in the cluster.

In UAV networks frequent change of cluster head would impose large overhead on the network.

Static Protocols: DCR - Data Centric Routing

Routing based on the content of data.

GOAL: use for one-to-many transmission in UAV networks when the data is requested by a number of nodes.

Works well with cluster topologies where cluster head is responsible for disseminating information to other nodes in the cluster.

In UAV networks frequent change of cluster head would impose large overhead on the network.

Proactive Protocols

Characteristics:

- routing tables saved on nodes to store all the routing information of other nodes
- tables need to be updated when topology changes.

Advantage: proactive routing contains the latest information of the routes.

Disadvantage: to keep the tables up-to-date needed a number of messages exchanged between nodes.

Unsuitable for UAV network because:

- bandwidth constraints
- slow reaction to topology changes causing delays

Protocols:

- **OLSR** - Optimized Link State Routing
- **DSDV** - Destination-Sequenced Distance Vector
- **B.A.T.M.A.N.** – Better Approach to Mobile Ad Hoc Network

Proactive Protocols: OLSR - Optimized Link State Routing (1)

Main characteristics:

- link state protocol for MANETs
- suited for large and dense networks
- routes determined at startup and then maintained by an update process
- fully distributed: routes always available when needed
- routing decision: best path used for comparison

Naïve approach:

- nodes exchange topology information with other nodes by **flooding strategy**
- then nodes update their routes by **choosing the next hop by a shortest path algorithm to all destinations.**

Increased overhead of control messages when topology change quickly

Proactive Protocols: OLSR - Optimized Link State Routing (2)

How it works:

Decrease the overhead of flooding by means of selecting some nodes as **multipoint relays** (**MPR**), which alone forward the control traffic reducing the transmission required

MPRs:

- declare link state information to all the nodes that selected them as an MPR
- periodically announce to the network that it has reachability to all the nodes that selected it as an MPR.

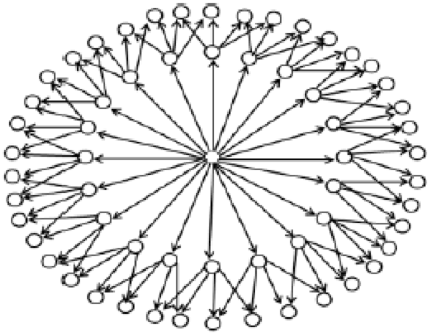


Fig. 1 Normal Flooding

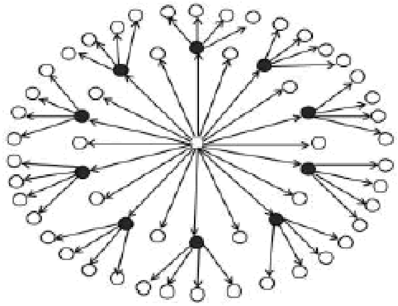


Fig. 2 MPR Flooding

Proactive Protocols: OLSR - Optimized Link State Routing (3)

OLSR in UAVNET:

Pros:

- flooding of control signals avoided with MPRs
- network information always available
- updates time tuneable to increase reactivity to topological changes

Cons:

- routes recalculation issue because limited computing power of UAV nodes

Proactive Protocols: DSDV - Destination Sequenced Distance Vector (1)

Main characteristics:

- table-driven protocol
- distance vector approach (use Bellman-Ford algorithm with small adjustment)
- fresh routes better than stale routes
- updates on route periodically or when significant new information available

Type of update packets:

- **full-dump:** send all the route information
- **incremental-only changes:** when topology of the network changes

Identify fresh routes:

By means of sequence numbers identifying the freshness of the communicated information.
When changes occur, the sequence number increase.

Proactive Protocols: DSDV - Destination Sequenced Distance Vector (2)

How it works:

- every entry in the routing table has a sequence number with updates having increasing sequence numbers
- periodically destination nodes transmit updates with a new sequence number
- updates periodically sent by nodes contain information on the costs to achieve the different destinations and the freshness of the route
- when two routes to a destination received from two different neighbours:
  - Choose the one with greatest destination sequence number
  - If equal choose the smaller metric (hop count)

This protocol needs large network bandwidth for update procedures and this puts DSDV at a disadvantage for UAV network.

Proactive Protocols: B.A.T.M.A.N. - Better Approach to Ad Hoc Mobile Network (1)

Main characteristics:

- new protocol for wireless ad hoc mesh networks
- decentralization of the knowledge about the best route through the network: no single node has the route to all the destination
- routing decision: next hop
- data accessible via single-hop or multi-hop communication links
- link quality detection
- loop-free

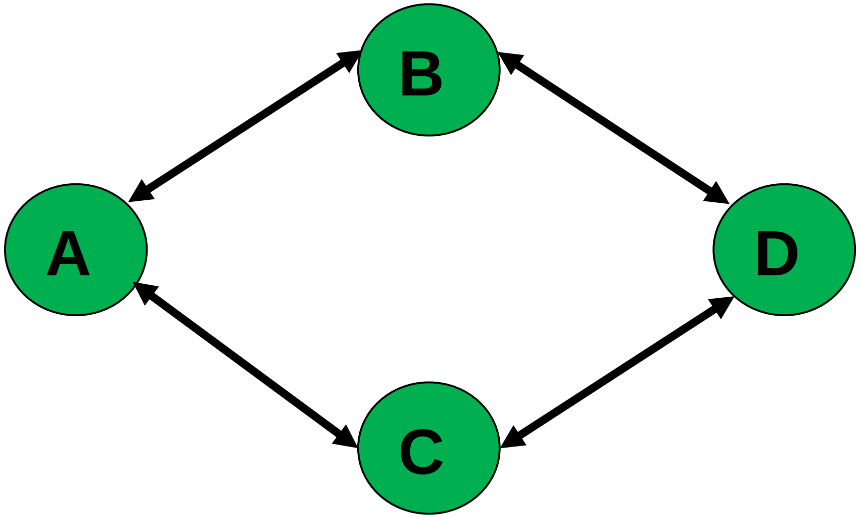
Basic Idea:

each node only maintains the general direction toward the destination and relays the data to the best next-hop neighbor

Proactive Protocols: B.A.T.M.A.N. - Better Approach to Ad Hoc Mobile Network (2)

How it works:

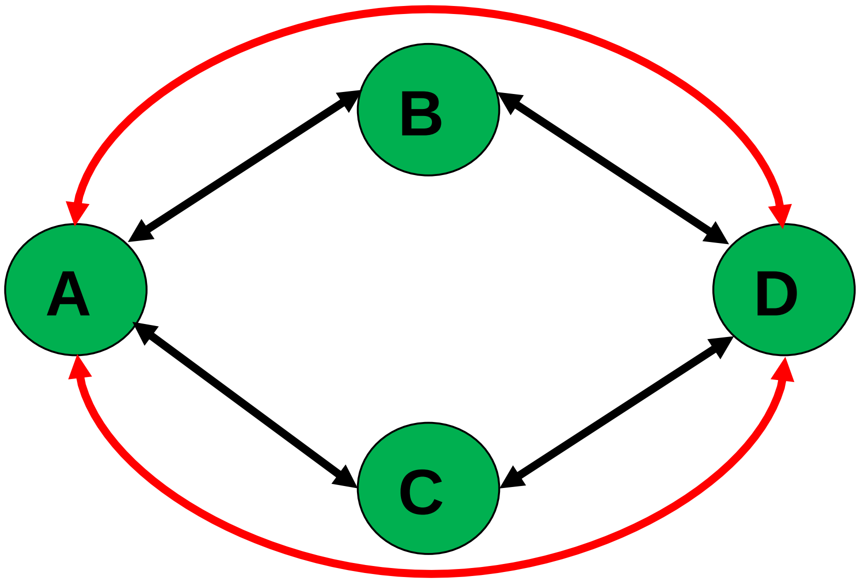
- Each node has a set of direct-link neighbors.
- A has neighbors B and C. These are the nodes through which A sends and receives all its packets.
- Each node in the network sends an **Originator Message** (OGM) periodically, in order to inform all other nodes of its presence.



Proactive Protocols: B.A.T.M.A.N. - Better Approach to Ad Hoc Mobile Network (3)

How it works:

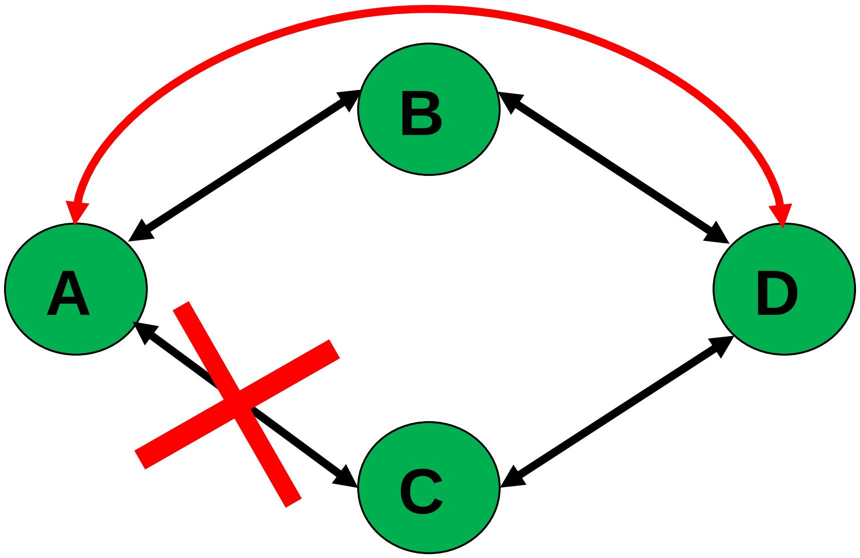
- If all links are perfect, Node A will receive node D's OGM through both of its neighbors B and C.
- If all of D's OGMs arrive through both B and C, then when A needs to send something to D, it can use either B or C as the next hop towards the destination node D.



Proactive Protocols: B.A.T.M.A.N. - Better Approach to Ad Hoc Mobile Network (4)

What happens if the link between nodes A and C goes down?:

- Node D's OGM will only arrive to A through node B.
- Node A therefore considers node B as the best next hop neighbor for all packets destined for node D
- Node C's OGMs will also only reach node A through node B. Node B is the best next hop for data destined for Node C



Proactive Protocols: B.A.T.M.A.N. - Better Approach to Ad Hoc Mobile Network (5)

B.A.T.M.A.N. in UAVNET:

Pros:

- **Decentralized**
- **Loop-free**
- **Protocol packet very small**

Cons:

- **Not perform well if network is reliable**
- **Performance depend strictly on packet loss**

Reactive Protocols

Characteristics:

- On demand
- No periodic messages
- **Source** or **hop by hop** routing
- Route acquisition latency

Protocols:

- **DSR** – Dynamic Source Routing
- **AODV** – *Ad hoc On Demand Distance Vector*

Reactive Protocols: DSR – Dynamic Source Routing (1)

Main characteristics:

- On Demand
- Source Routing
- Route discovery
- Load balancing
- Unsuitable for large network
- Loop free

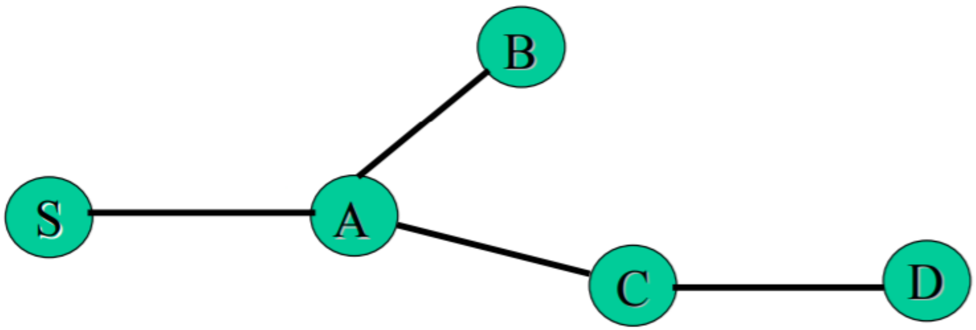
Types of messages:

- **RREQ - ** Route Request
- **RREP - ** Route Response
- **RERR** - Route Error

Reactive Protocols: DSR – Dynamic Source Routing (2)

How it works – A brief Recap:

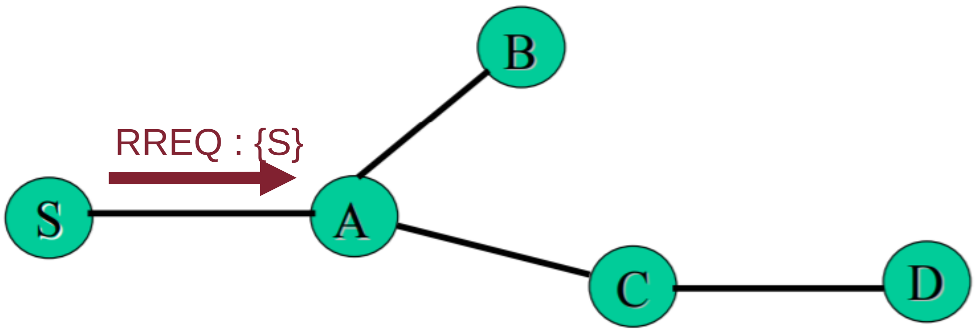
- **Node S wants to speak with D**



Reactive Protocols: DSR – Dynamic Source Routing (2)

How it works – A brief Recap:

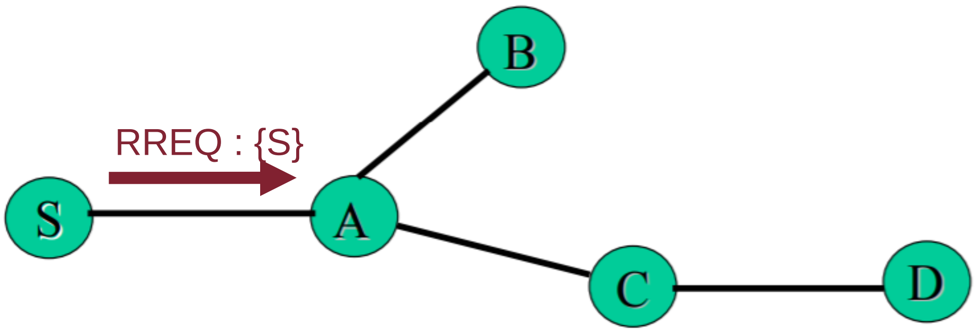
- Node S wants to speak with D
- **S broadcasts RREQ packets**



Reactive Protocols: DSR – Dynamic Source Routing (2)

How it works – A brief Recap:

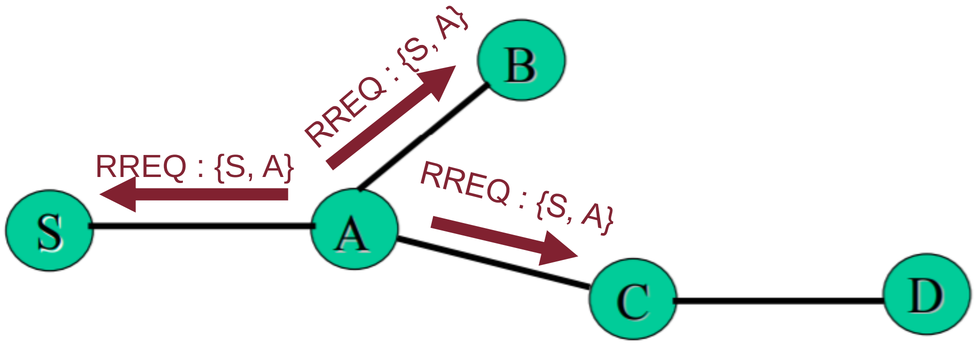
- Node S wants to speak with D
- S broadcasts RREQ packets
- **A has not route to D**



Reactive Protocols: DSR – Dynamic Source Routing (2)

How it works – A brief Recap:

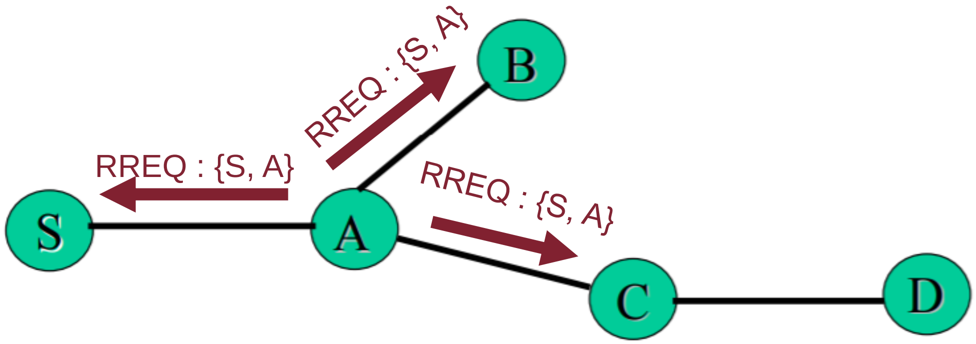
- Node S wants to speak with D
- S broadcasts RREQ packets
- A has not route to D
- **A broadcasts RREQ packets**



Reactive Protocols: DSR – Dynamic Source Routing (2)

How it works – A brief Recap:

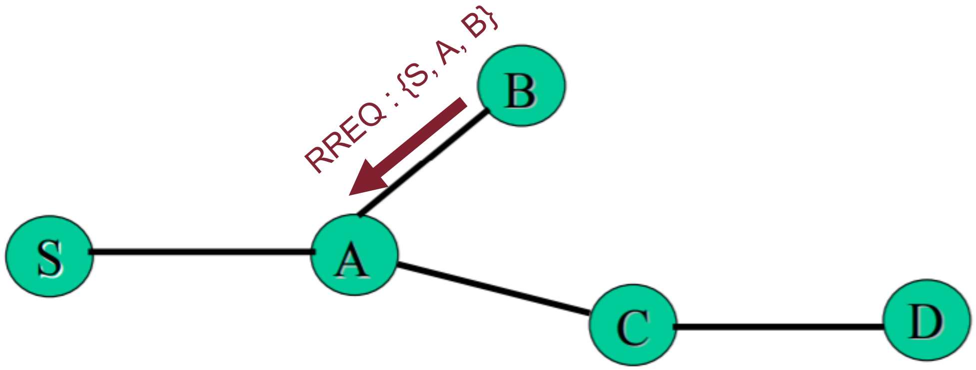
- Node S wants to speak with D
- S broadcasts RREQ packets
- A has not route to D
- A broadcasts RREQ packets
- **B has not route to D**



Reactive Protocols: DSR – Dynamic Source Routing (2)

How it works – A brief Recap:

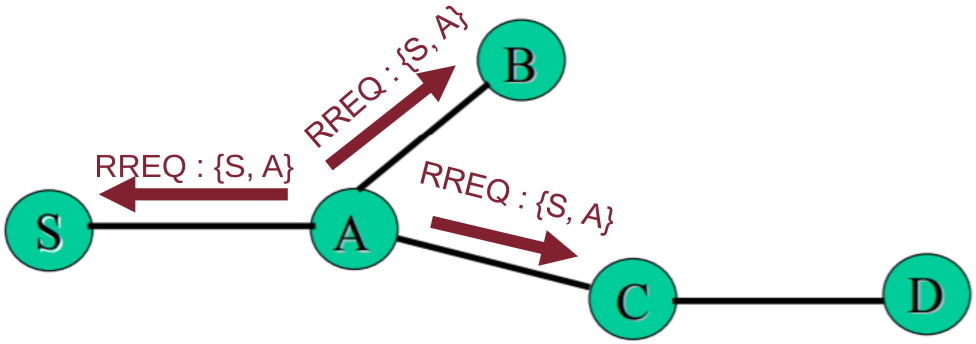
- Node S wants to speak with D
- S broadcasts RREQ packets
- A has not route to D
- A broadcasts RREQ packets
- B has not route to D
- **B broadcasts RREQ packets**



Reactive Protocols: DSR – Dynamic Source Routing (2)

How it works – A brief Recap:

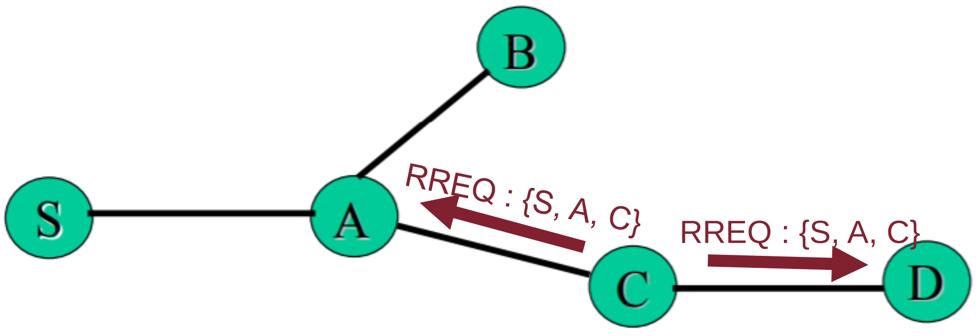
- Node S wants to speak with D
- S broadcasts RREQ packets
- A has not route to D
- A broadcasts RREQ packets
- B has not route to D
- B broadcasts RREQ packets
- **C has not route to D**



Reactive Protocols: DSR – Dynamic Source Routing (2)

How it works – A brief Recap:

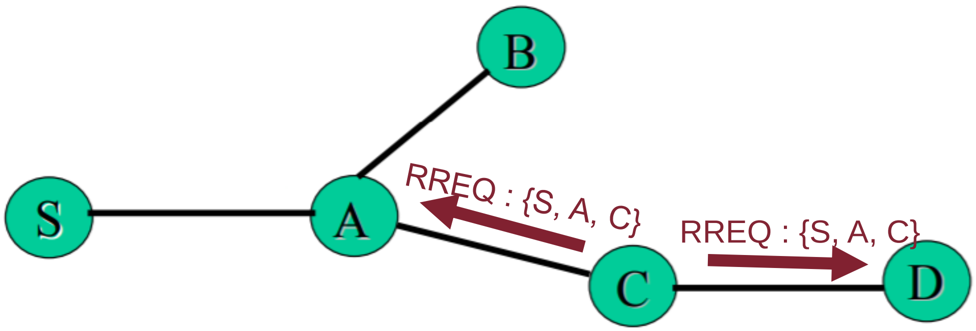
- Node S wants to speak with D
- S broadcasts RREQ packets
- A has not route to D
- A broadcasts RREQ packets
- B has not route to D
- B broadcasts RREQ packets
- C has not route to D
- **C broadcasts RREQ packets**



Reactive Protocols: DSR – Dynamic Source Routing (2)

How it works – A brief Recap:

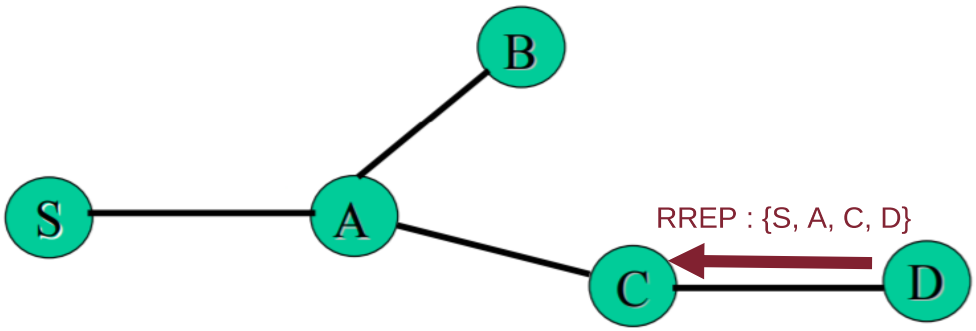
- Node S wants to speak with D
- S broadcasts RREQ packets
- A has not route to D
- A broadcasts RREQ packets
- B has not route to D
- B broadcasts RREQ packets
- C has not route to D
- C broadcasts RREQ packets
- **D is the Target Node**



Reactive Protocols: DSR – Dynamic Source Routing (2)

How it works – A brief Recap:

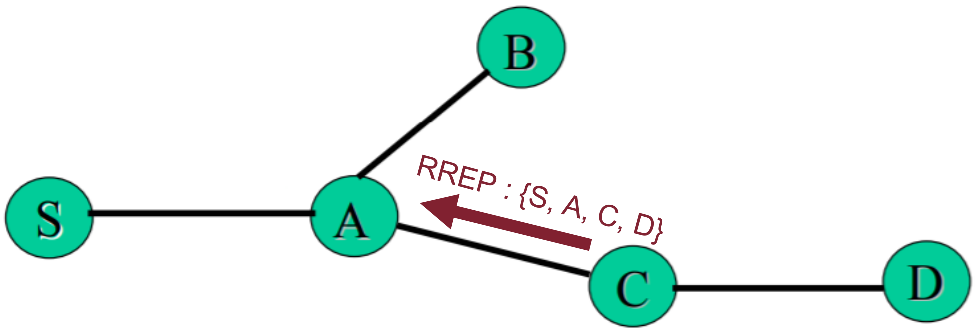
- Node S wants to speak with D
- S broadcasts RREQ packets
- A has not route to D
- A broadcasts RREQ packets
- B has not route to D
- B broadcasts RREQ packets
- C has not route to D
- C broadcasts RREQ packets
- D is the Target Node
- **D response with a RREP packet to C**



Reactive Protocols: DSR – Dynamic Source Routing (2)

How it works – A brief Recap:

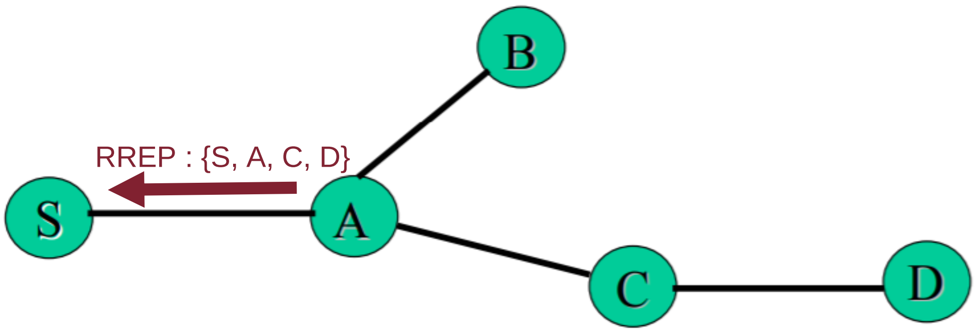
- Node S wants to speak with D
- S broadcasts RREQ packets
- A has not route to D
- A broadcasts RREQ packets
- B has not route to D
- B broadcasts RREQ packets
- C has not route to D
- C broadcasts RREQ packets
- D is the target node
- D response with a RREP packet to C
- **C forward RREP to A**



Reactive Protocols: DSR – Dynamic Source Routing (2)

How it works – A brief Recap:

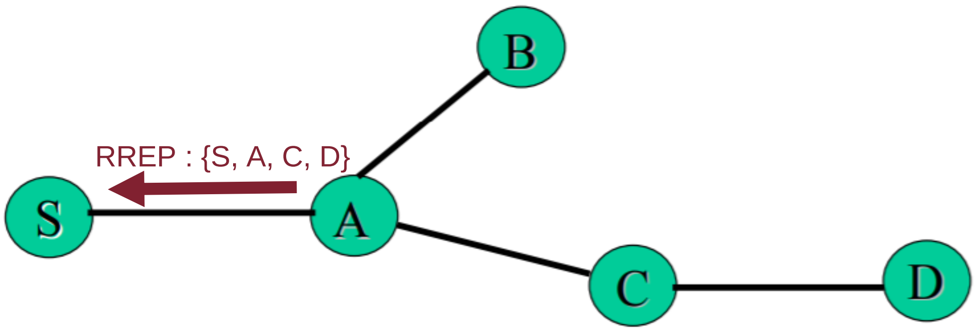
- Node S wants to speak with D
- S broadcasts RREQ packets
- A has not route to D
- A broadcasts RREQ packets
- B has not route to D
- B broadcasts RREQ packets
- C has not route to D
- C broadcasts RREQ packets
- D is the target node
- D response with a RREP packet to C
- C forward RREP to A
- **A forward RREP to S**



Reactive Protocols: DSR – Dynamic Source Routing (2)

How it works – A brief Recap:

- Node S wants to speak with D
- S broadcasts RREQ packets
- A has not route to D
- A broadcasts RREQ packets
- B has not route to D
- B broadcasts RREQ packets
- C has not route to D
- C broadcasts RREQ packets
- D is the target node
- D response with a RREP packet to C
- C forward RREP to A
- A forward RREP to S
- **S cache the new route**



Reactive Protocols: DSR – Dynamic Source Routing (3)

DSR in UAVNET:

Pros:

- **On demand**
- **Free loop**
- **Load balancing**
- **Less control overhead**

Cons:

- **Not scale well on large network**
- **High latency in route finding (flooding)**
- **Dynamic network can be a problem**

Reactive Protocols: AODV – Ad Hoc On demand Distance Vector (1)

Main characteristics:

- On Demand
- Hop by hop Routing
- Routing table on each node
- TTL of each route
- Based on distance vector
- Loop free
- ‘Hello’ messages

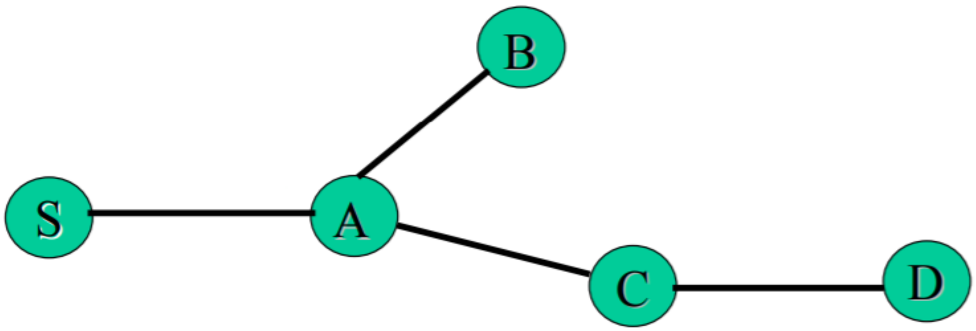
Types of messages:

- **RREQ** **-** Route Request
- **RREP** **-** Route Reply
- **RERR** - Route Error

Reactive Protocols: AODV – Ad Hoc On demand Distance Vector (2)

How it works – A brief Recap:

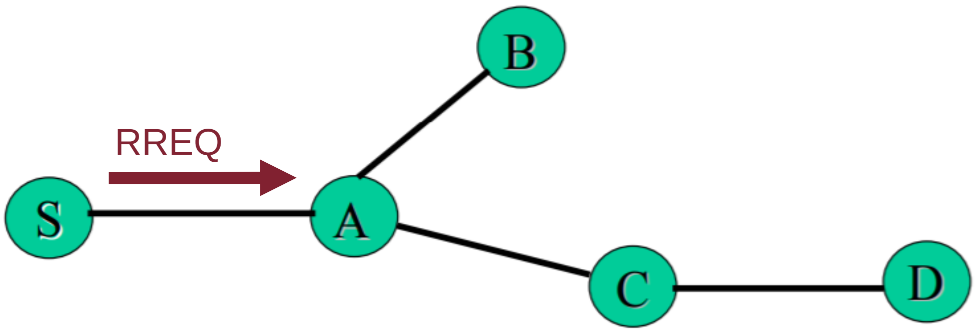
- **Node S wants to speak with D**



Reactive Protocols: AODV – Ad Hoc On demand Distance Vector (2)

How it works – A brief Recap:

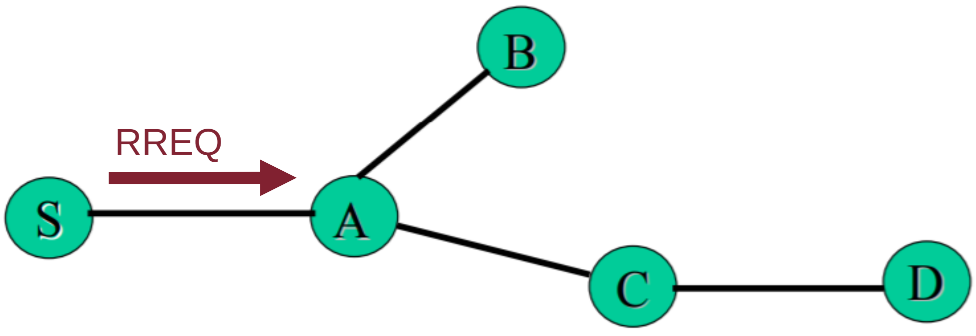
- Node S wants to speak with D
- **S broadcasts RREQ packets with Destination “D” and Source “S”**



Reactive Protocols: AODV – Ad Hoc On demand Distance Vector (2)

How it works – A brief Recap:

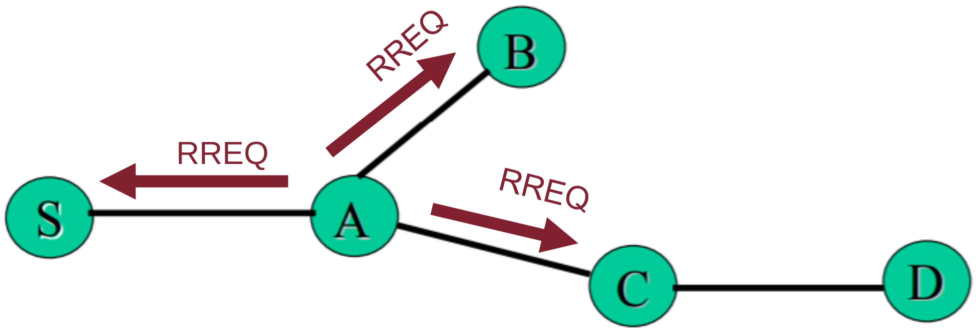
- Node S wants to speak with D
- S broadcasts RREQ packets with Destination “D” and Source “S”
- **A store route reversal: Dest “S” – Next hop “S”**



Reactive Protocols: AODV – Ad Hoc On demand Distance Vector (2)

How it works – A brief Recap:

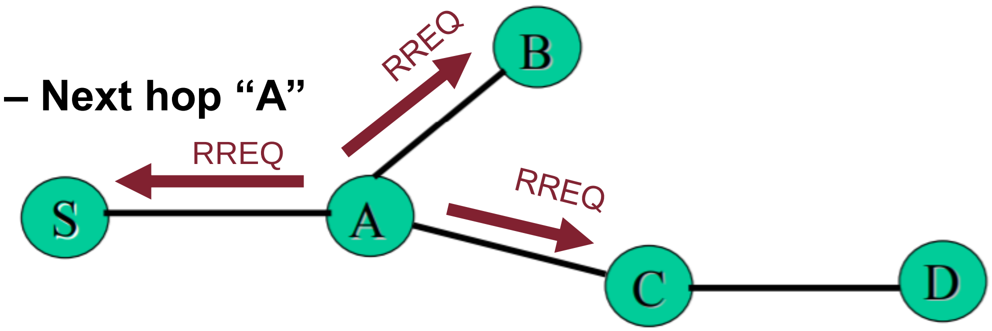
- Node S wants to speak with D
- S broadcasts RREQ packets with Destination “D” and Source “S”
- A store route reversal: Dest “S” – Next hop “S”
- **A rebroadcasts RREQ**



Reactive Protocols: AODV – Ad Hoc On demand Distance Vector (2)

How it works – A brief Recap:

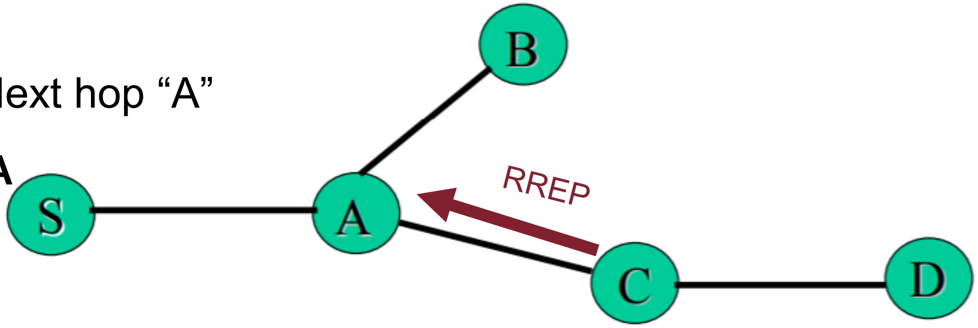
- Node S wants to speak with D
- S broadcasts RREQ packets with Destination “D” and Source “S”
- A store route reversal: Dest “S” – Next hop “S”
- A rebroadcasts RREQ
- **C store reversal route: Dest “S” – Next hop “A”**



Reactive Protocols: AODV – Ad Hoc On demand Distance Vector (2)

How it works – A brief Recap:

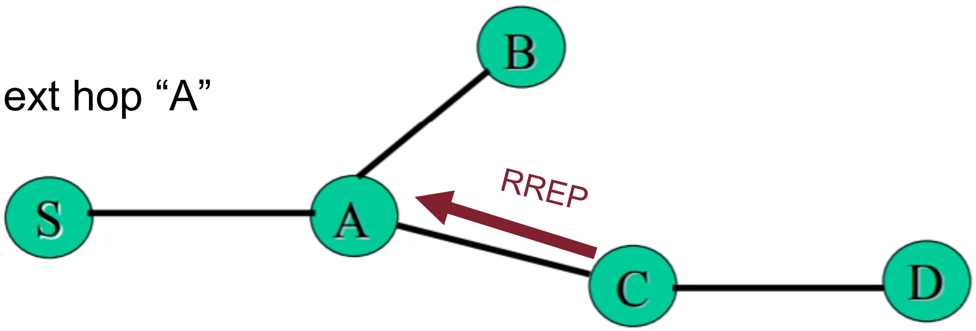
- Node S wants to speak with D
- S broadcasts RREQ packets with Destination “D” and Source “S”
- A store route reversal: Dest “S” – Next hop “S”
- A rebroadcasts RREQ
- C store reversal route: Dest “S” – Next hop “A”
- **C create RREP and unicast it to A**



Reactive Protocols: AODV – Ad Hoc On demand Distance Vector (2)

How it works – A brief Recap:

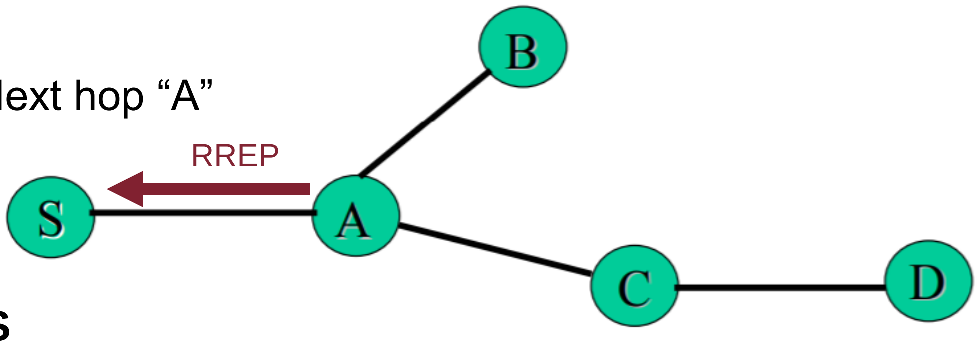
- Node S wants to speak with D
- S broadcasts RREQ packets with Destination “D” and Source “S”
- A store route reversal: Dest “S” – Next hop “S”
- A rebroadcasts RREQ
- C store reversal route: Dest “S” – Next hop “A”
- C create RREP and unicast it to A
- **A store: Dest “D” – Next hop “C”**



Reactive Protocols: AODV – Ad Hoc On demand Distance Vector (2)

How it works – A brief Recap:

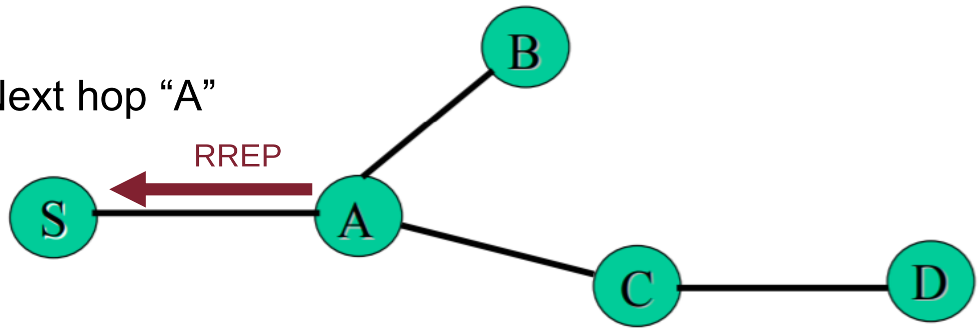
- Node S wants to speak with D
- S broadcasts RREQ packets with Destination “D” and Source “S”
- A store route reversal: Dest “S” – Next hop “S”
- A rebroadcasts RREQ
- C store reversal route: Dest “S” – Next hop “A”
- C create RREP and unicast it to A
- A store: Dest “D” – Next hop “C”
- **A create RREP and unicast it to S**



Reactive Protocols: AODV – Ad Hoc On demand Distance Vector (2)

How it works – A brief Recap:

- Node S wants to speak with D
- S broadcasts RREQ packets with Destination “D” and Source “S”
- A store route reversal: Dest “S” – Next hop “S”
- A rebroadcasts RREQ
- C store reversal route: Dest “S” – Next hop “A”
- C create RREP and unicast it to A
- A store: Dest “D” – Next hop “C”
- A create RREP and unicast it to S
- **S store the route: Dest “D” – Next hop “A”**



Reactive Protocols: AODV – Ad Hoc On demand Distance Vector (3)

AODV in UAVNET:

Pros:

- On demand
- Hop by hop routing
- No loop
- Minimal routing traffic

Cons:

- “Hello” messages timing and overhead
- High latency in route finding (flooding)
- Intermittent links affect the throughput
- No load balancing

Reactive Protocols: AODV – Ad Hoc On demand Distance Vector (4)

Performance evaluation of AODV in UAVNET:

Simulator: **NS-2**

Mac Protocol: **IEEE 802.11**

Transport Layer protocol: **TCP/UDP**

Performance Metrics: **Throughput, #AODV RREQ messages**

Parameters and factors:

- Probability **p** that some node is down. (0.005 ≤ **p** ≥ 0.55)
- Slot time - 3.0 sec
- 36 nodes
- TCP/UDP
- Fixed position of nodes, no mobility

- B. Fu and L. A. DaSilva, “A mesh in the sky: A routing protocol for airborne networks,” in *Proc. IEEE Mil. Commun. Conf. (MILCOM’07)*, Oct. 2007, pp. 1–7.

Reactive Protocols: AODV – Ad Hoc On demand Distance Vector (5)

Performance evaluation of AODV in UAVNET:

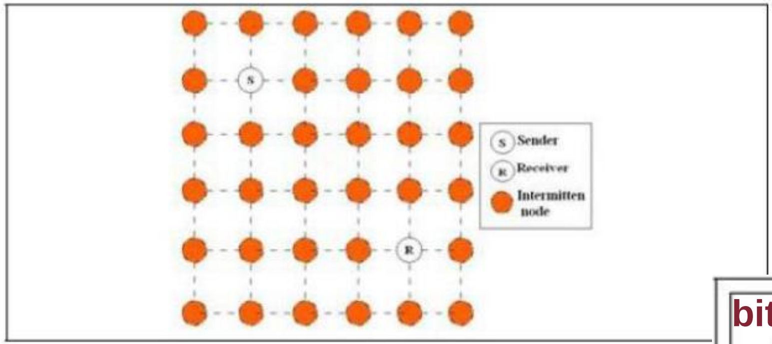


Figure 6. 36-node scenario with IEEE 802.11 MAC

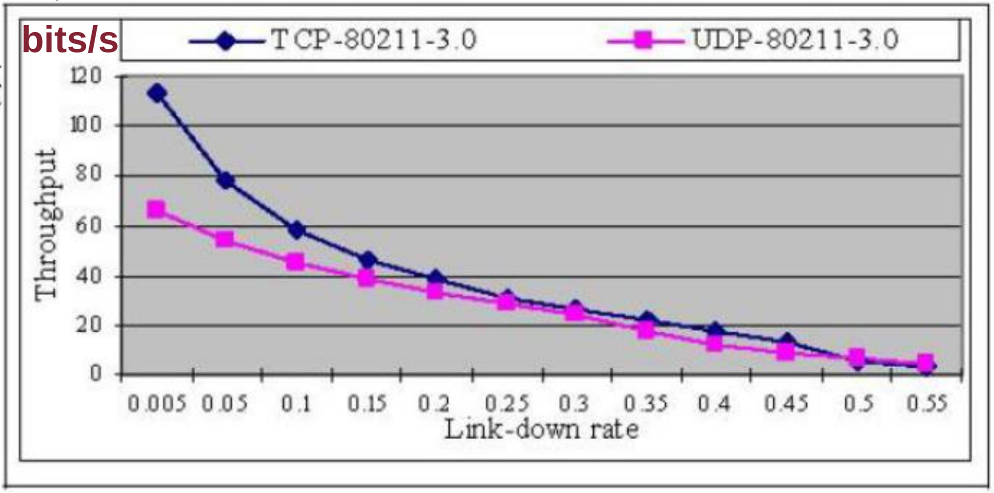


Figure 7. Throughput

- B. Fu and L. A. DaSilva, “A mesh in the sky: A routing protocol for airborne networks,” in *Proc. IEEE Mil. Commun. Conf. (MILCOM’07)*, Oct. 2007, pp. 1–7.

Hybrid Protocols

Characteristics/Goals:

- Mix reactive and proactive approach
- Reduce reactive delay of discovery process
- Reduce proactive overhead of control messages
- Scale well on large network
- Hard to implement

Protocols:

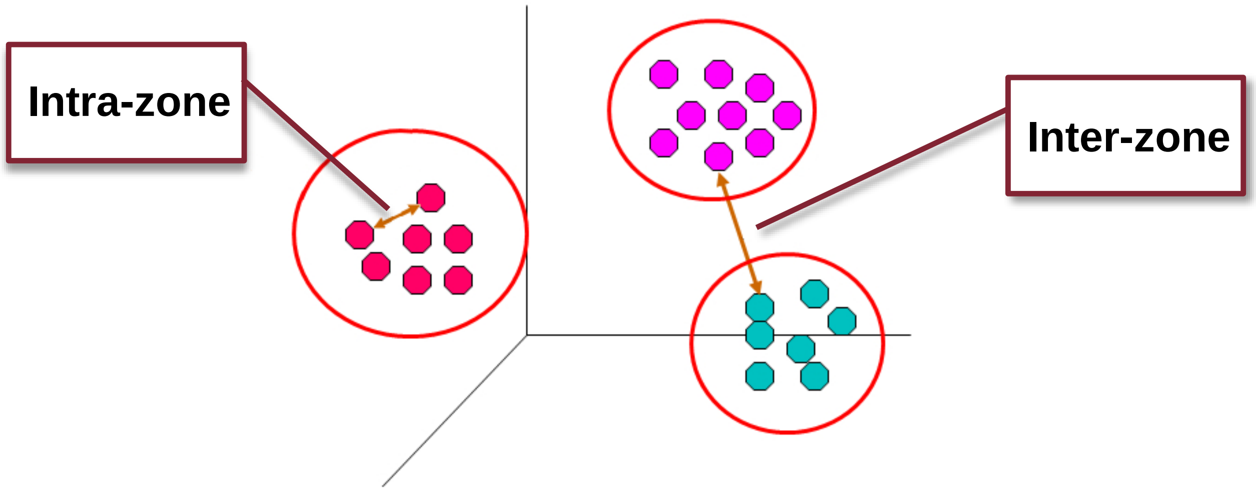
- **ZRP** – Zone Routing Protocol
- **TORA** – Temporarily Ordered Routing Algorithm

Hybrid Protocols: ZRP – Zone Routing Protocol (1)

Characteristics/Goals:

- ZRP **combines** both routing approaches and reduce their disadvantages:
  - **reduce** the amount of update traffic.
  - **reduce** the delay of route discovery.

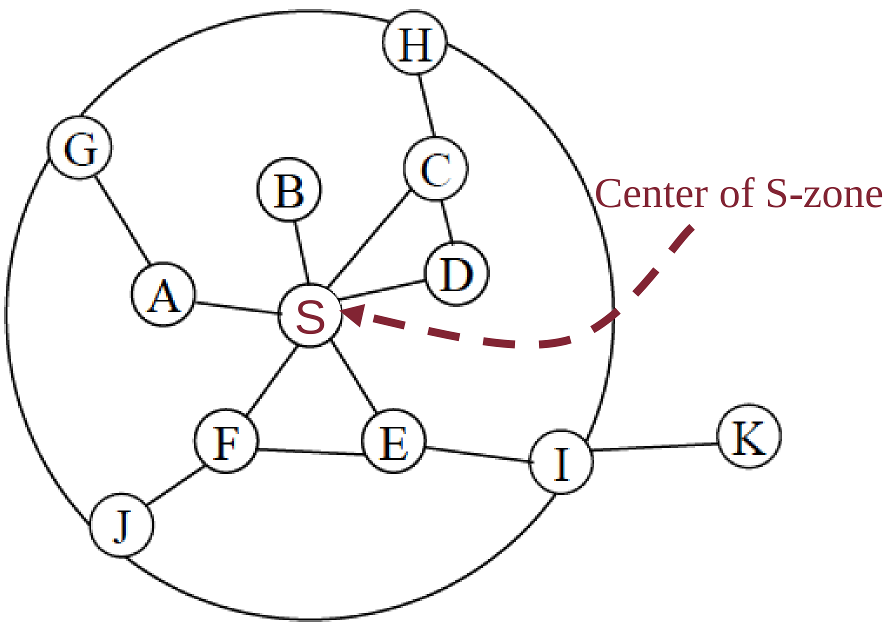
- The network is divided into a number of zones where:
  - **intra-zone** routing is performed with the **proactive** approach.
  - **inter-zone** routing is done using the **reactive** approach.



Hybrid Protocols: ZRP – Zone Routing Protocol (2)

Characteristics:

- A zone is defined for each node
- Zone radius ρ = *#hops*
- ρ affect the performance
- The zones overlap
- Regular "Hello" messages among neighbors.



Notations:

S is **central node.**

{G,H,J,I} are **peripheral nodes.**

{E,D,B,..} are **internal nodes.**

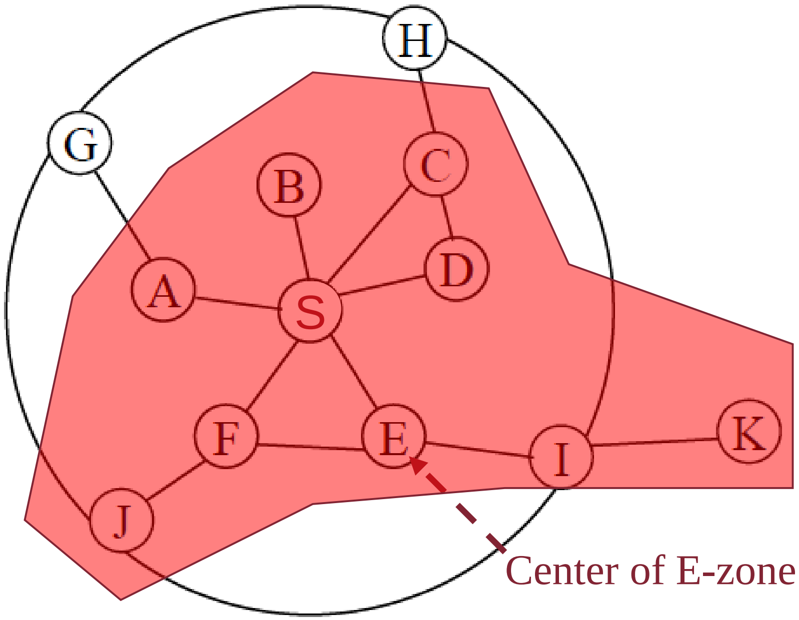
{K} is an **external nodes.**

Radius ρ = 2 hops

Hybrid Protocols: ZRP – Zone Routing Protocol (2)

Characteristics:

- A zone is defined for each node
- Zone radius ρ = *#hops*
- ρ affect the performance
- The zones overlap
- Regular "Hello" messages among neighbors.



Notations:

S is **central node.**

{G,H,J,I} are **peripheral nodes.**

{E,D,B,..} are **internal nodes.**

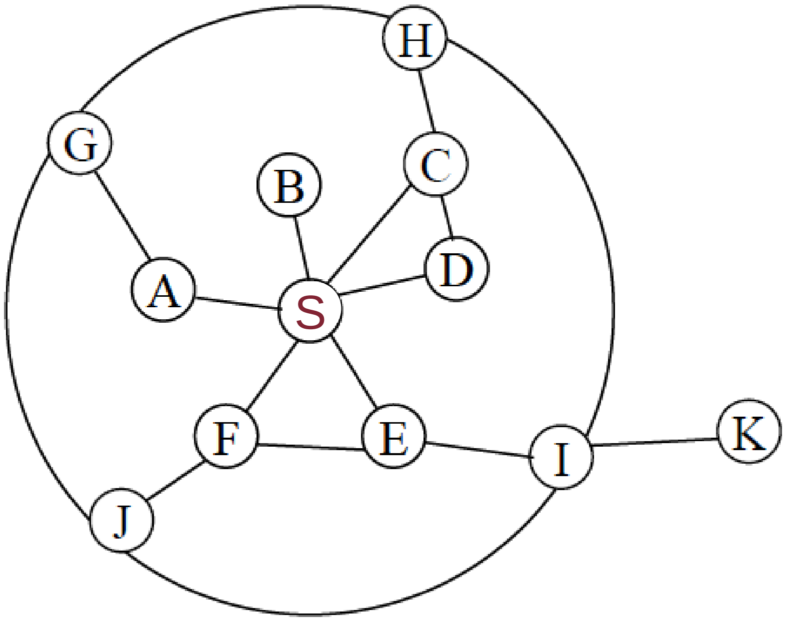
{K} is an **external nodes.**

Radius ρ = 2 hops

Hybrid Protocols: ZRP – Zone Routing Protocol (3)

Proactive Intra-zone Routing:

- Not a specific routing protocol
- **IARP** routing protocols family
- Each zone use the protocol to update routing info
- Node S store routing info about its local zone

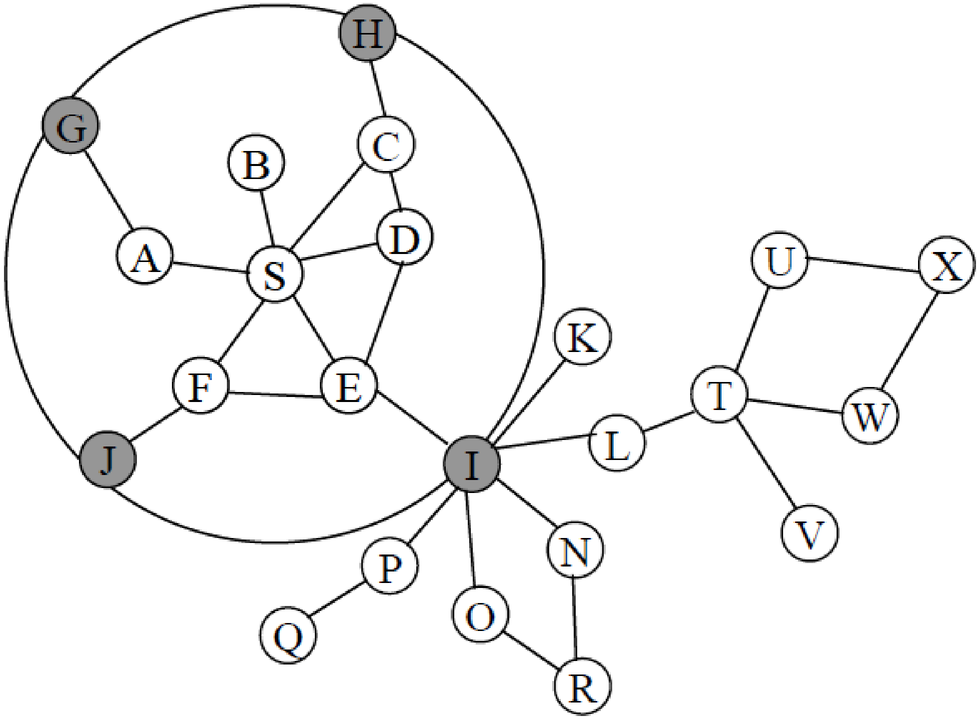


IARP: is a family of limited-depth, proactive link-state routing protocols

Hybrid Protocols: ZRP – Zone Routing Protocol (4)

Reactive Inter-zone Routing:

- Not a specific routing protocol
- **IERP** routing protocols family
- Try to discover routes to destination beyond a zone
- Don’t use flooding
- Use peripheral nodes.

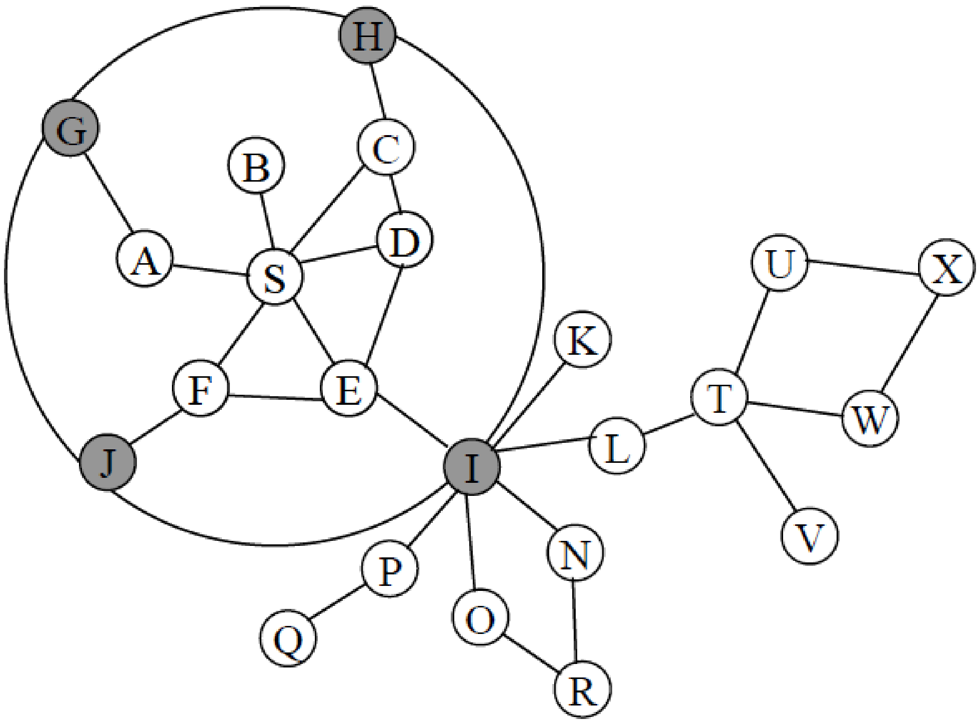


IERP: is a family of reactive routing protocols that offert enhanced route discovery based on IARP info.

Hybrid Protocols: ZRP – Zone Routing Protocol (5)

Reactive Inter-zone Routing:

- **Route request**
- **Route reply**
- Use **border-casts** service (among peripheral nodes)
- Each node in the path adds its ID to the query.
- Destination **reply** by reversing the accumulated route IDs.

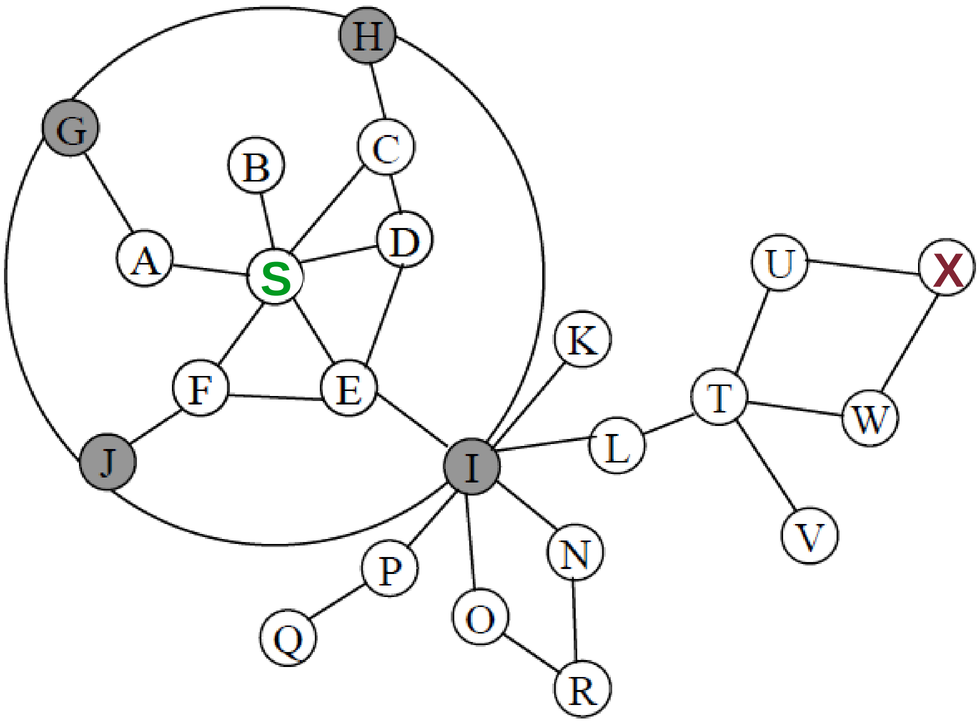


Border-casts service use **Bordercast Resolution Protocol (BRP)**

Hybrid Protocols: ZRP – Zone Routing Protocol (6)

Example of Inter-zone Routing protocol:

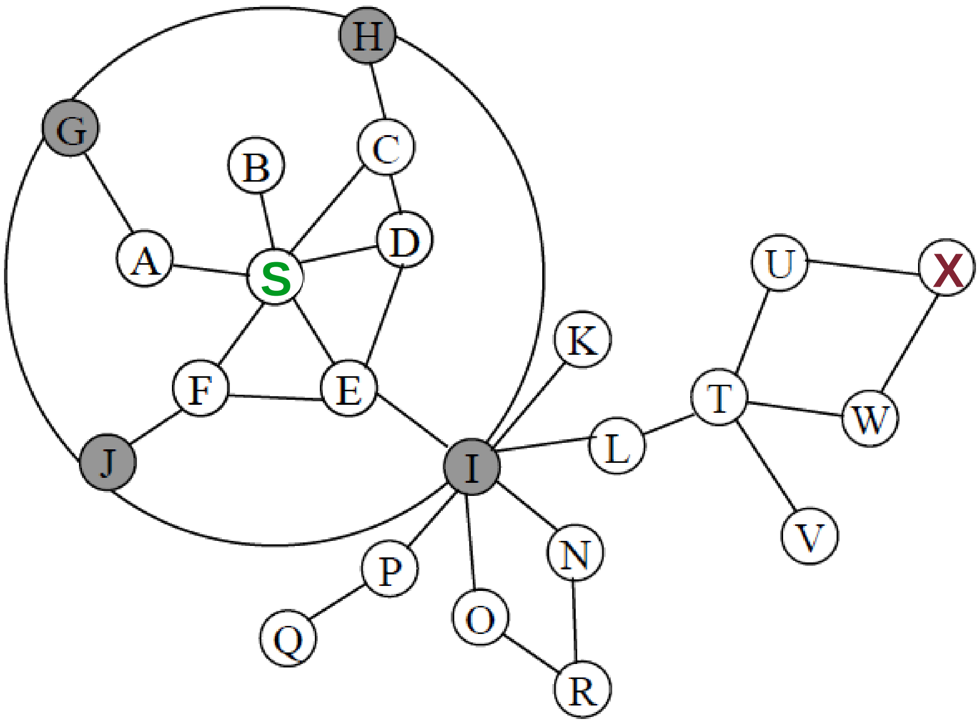
- Node **S** wants to speak with **X**



Hybrid Protocols: ZRP – Zone Routing Protocol (6)

Example of Inter-zone Routing protocol:

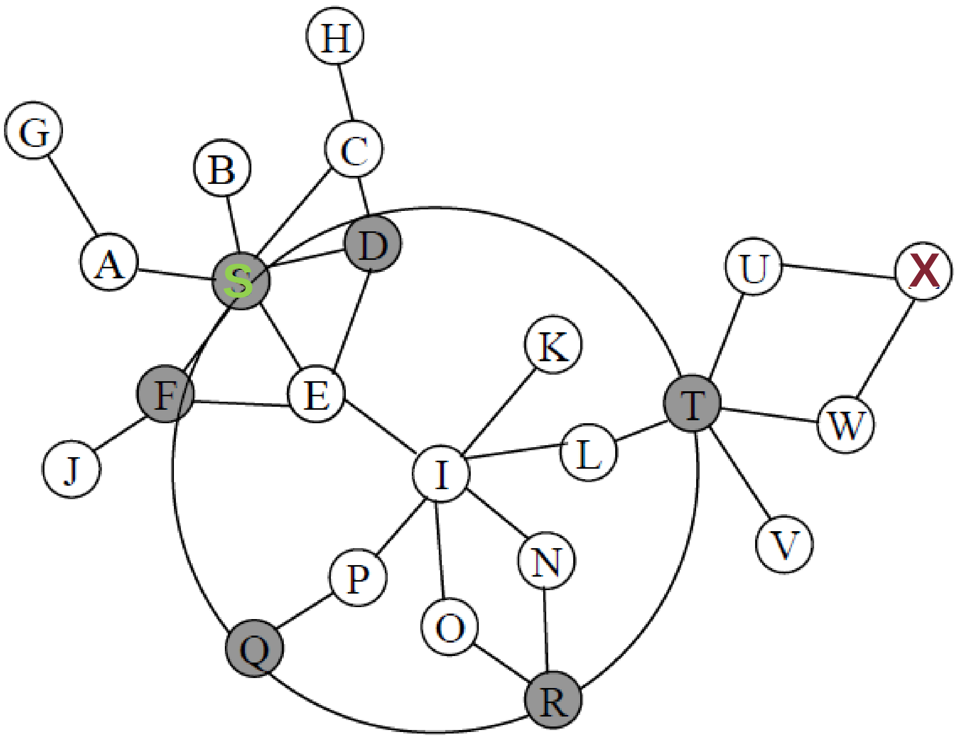
- Node **S** wants to speak with **X**
- The **route request** from S is bordercaste to peripheral nodes of its zone
- S doesn't see the destination in its zone



Hybrid Protocols: ZRP – Zone Routing Protocol (6)

Example of Inter-zone Routing protocol:

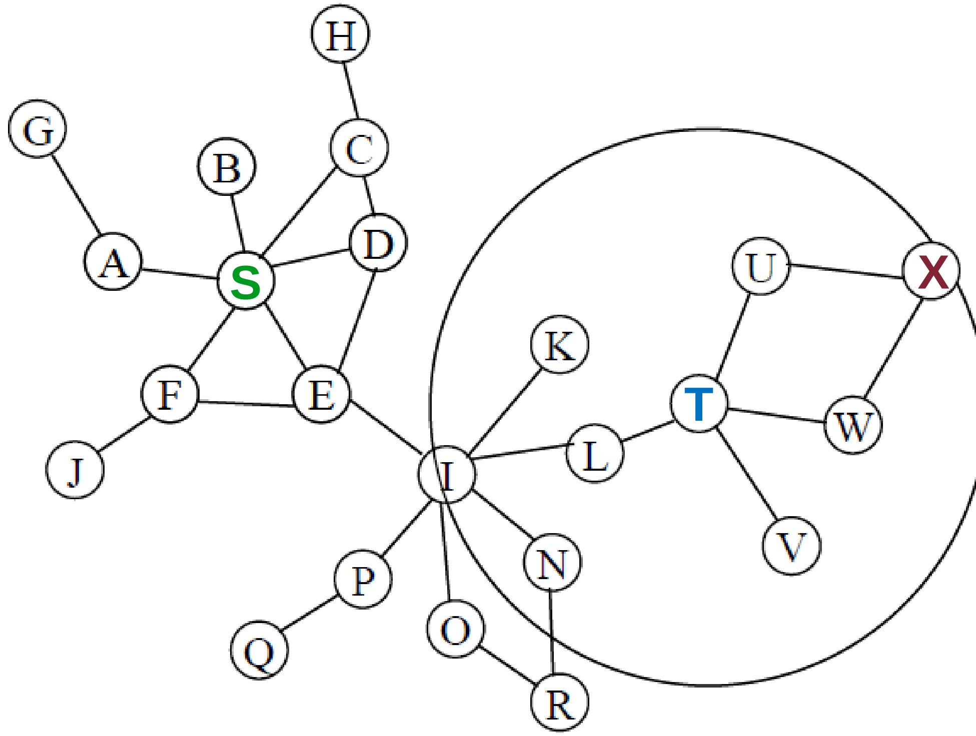
- Node **S** wants to speak with **X**
- The **route request** from S is bordercasted to peripheral nodes of its zone
- S doesn't see the destination in its zone
- Node I bordercasts the request to its peripheral nodes.



Hybrid Protocols: ZRP – Zone Routing Protocol (6)

Example of Inter-zone Routing protocol:

- Node **S** wants to speak with **X**
- The **route request** from S is bordercasted to peripheral nodes of its zone
- S doesn't see the destination in its zone
- Node I bordercasts the request to its peripheral nodes.
- The **route request** is received by Node T
- T finds in its routing zone the node **X** and send back **route reply** to node **S**
- The **route reply** contains the route path.



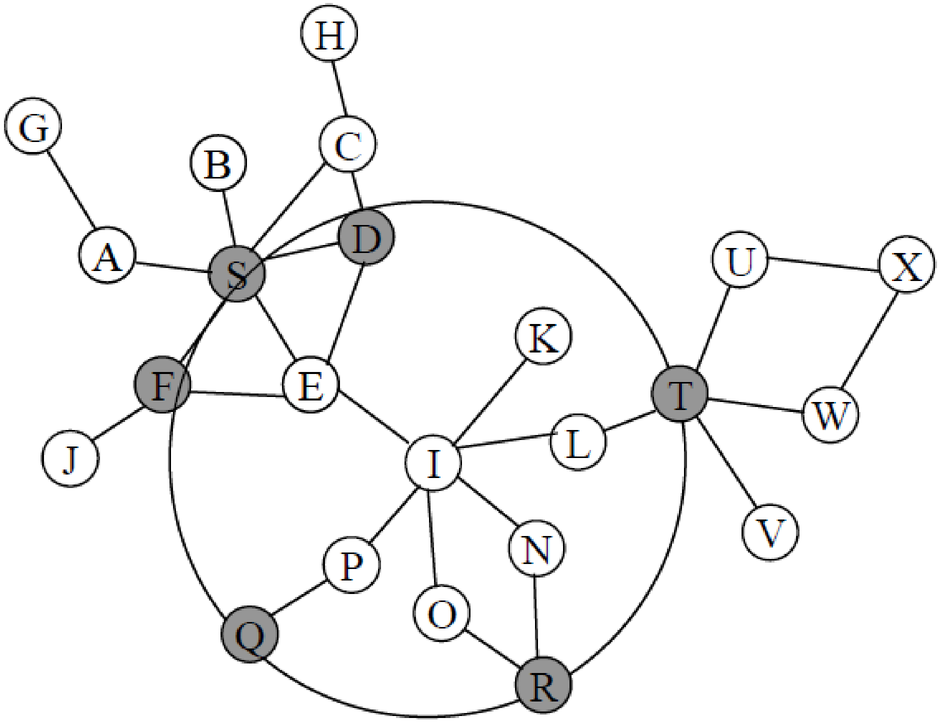
Hybrid Protocols: ZRP – Zone Routing Protocol (7)

Query Control Mechanism:

GOAL: We don't want that peripheral nodes forward the request packet in zone already covered by the routing.

Three query-control mechanism:

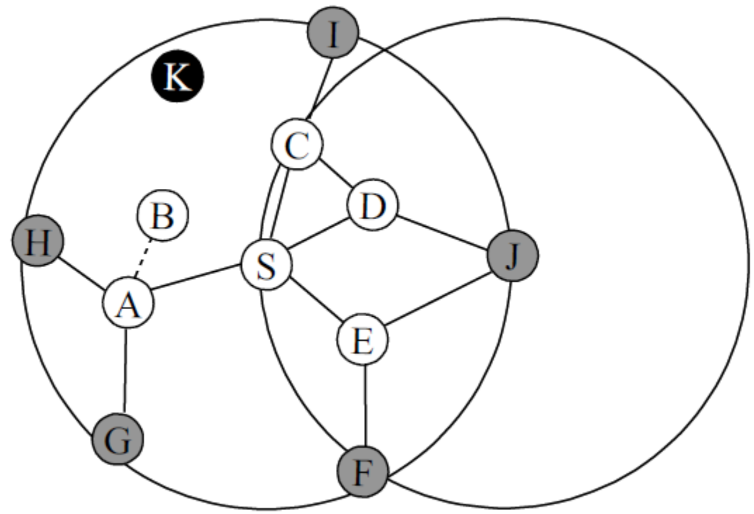
1. Query detection
2. Early Termination
3. Random query-processing delay



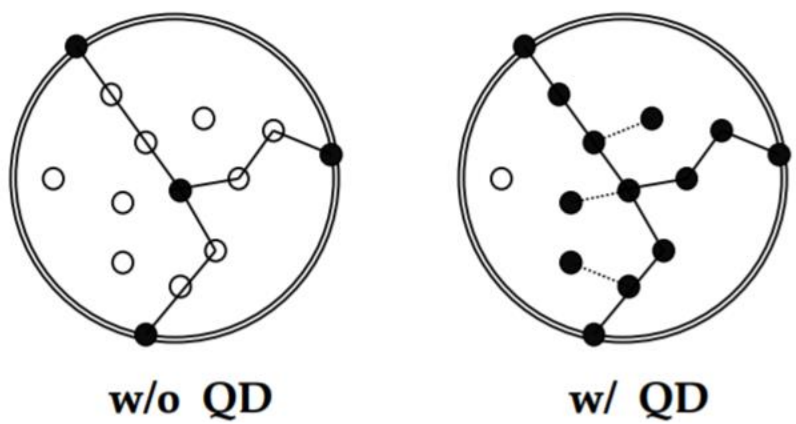
Hybrid Protocols: ZRP – Zone Routing Protocol (8)

Query Detection:

- **QD1 –** the nodes that relay the query are able to detect the same query. (C, A, E…)
- **QD2 –** the nodes that don't relay the query but are within the radio coverage can detect the same query. (B)
- Use a **detection table** to cache the detected queries. (source node – query ID)



The node S bordercasts to F-J.
J continues by bordercasting again to C,S,F through D,E (**redundant**).

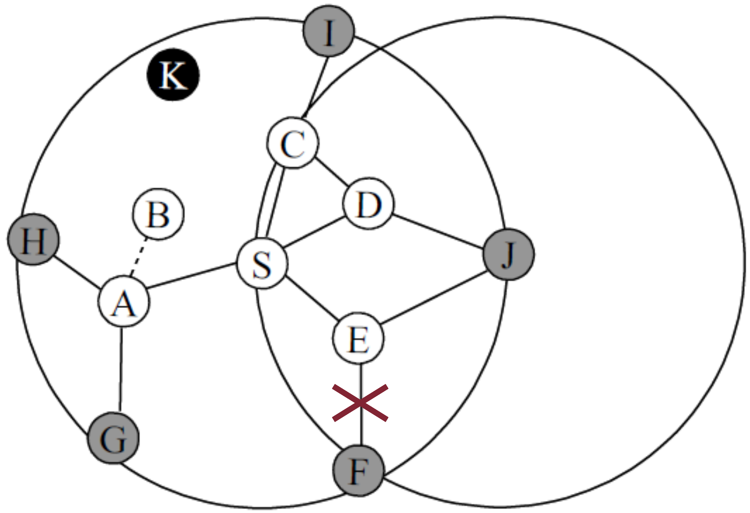


→ without QD only peripheral nodes are able to avoid redundant flooding.

Hybrid Protocols: ZRP – Zone Routing Protocol (9)

Early Termination:

- A node can prevent a route request from entering already covered regions.
- Use information of **query detection** and a knowledge of **local topology.**

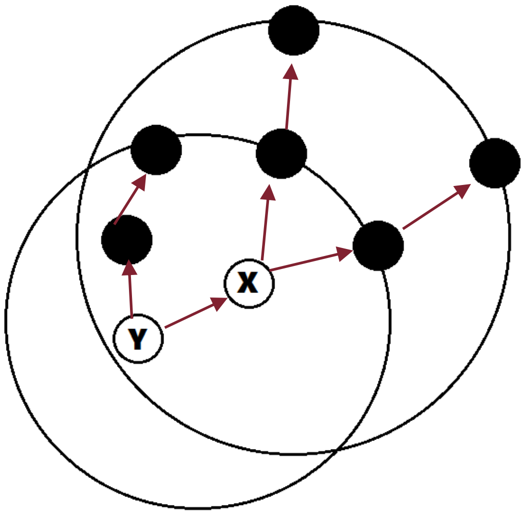


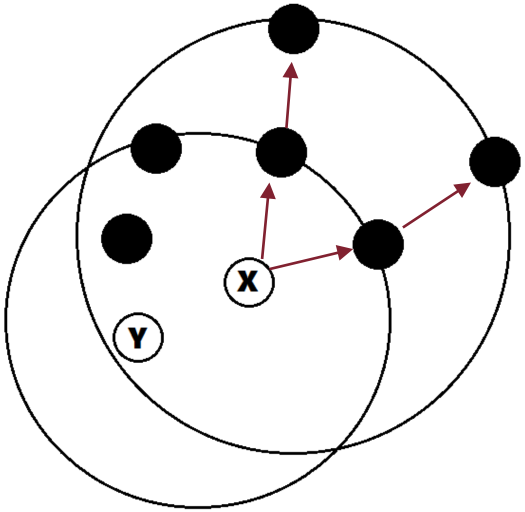
Node E can use the information in its query detection table to prune the query that the node J sends to its peripheral node F.
F was already used as peripheral node by S.

Hybrid Protocols: ZRP – Zone Routing Protocol (10)

Random query-processing delay:

- In some cases is possible that a two nodes receive simultaneously query and bordercast it.
- With RQPD, the nodes backs off a random period of time that allows one of them to detect other as being covered. And avoid redundant query.





Hybrid Protocols: ZRP – Zone Routing Protocol (11)

ZRP in UAVNET:

Pros:

- Take advantages of proactive and reactive
- Reduce update messages and initial delay
- No loop

Cons:

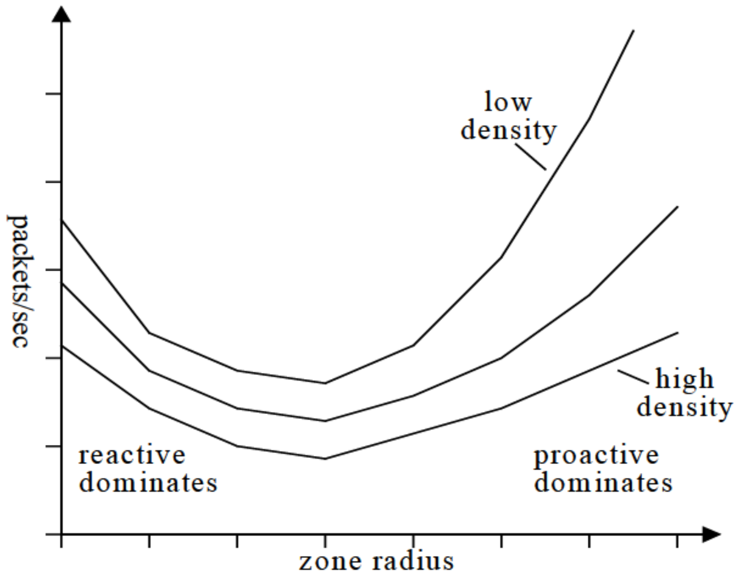
- High complexity
- Difficult to set up Radius ρ in UAV networks
- “Hello” messages timing and overhead

Hybrid Protocols: ZRP – Zone Routing Protocol (12)

ZRP in UAVNET:

- Can speculate whether the cost of added complexity outweighs the performance improvement of pure protocol components.

- The zone radius is a configurable parameter such that IARP traffic increase with the increasing of zone radius, otherwise IERP overhead increase.



Geographic Protocols 3D

Characteristics:

- knowledge of geographic position of the nodes
- no route discovery: sends message to the destination coordinates
- greedy forwarding: each node forwards the message to a node closest to the destination based only on the local information
- recovery mechanism to resolve local minimum (f**ace-routing**)
- 3D routing harder: greedy forwarding tend to encounter more local minima

Protocols:

- **GHG** – Greedy Hull Greedy
- **GRG** – Greedy Random Greedy
- **GDSTR-3D** – Greedy Distributed Spanning Tree Routing
- **MDT** – Multi-hop Delaunay Triangulation

Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (1)

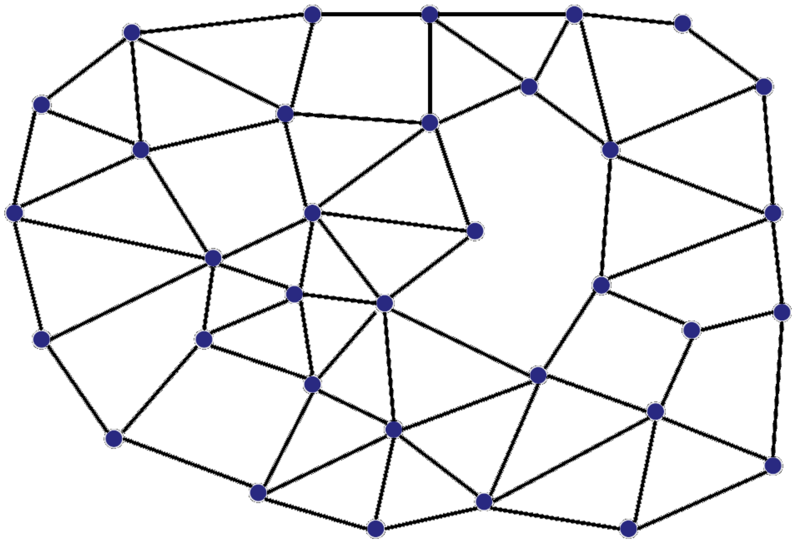
Characteristics:

- nodes have coordinates
- uses 2-hop neighbor information during greedy forwarding to reduce the likelihood of local minima
- if a local minimum is reached, forward packet along a spanning tree of convex hulls
- GDSTR-3D is able to guarantee packet delivery and achieve hop stretch close to 1

Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (2)

How basic GDSTR works:

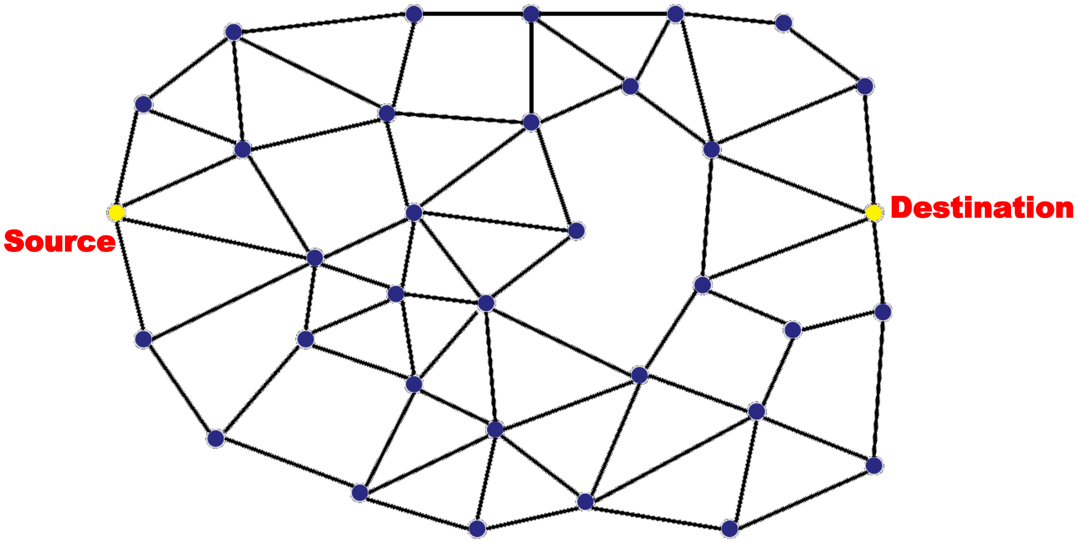
- Nodes have coordinates



Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (2)

How basic GDSTR works:

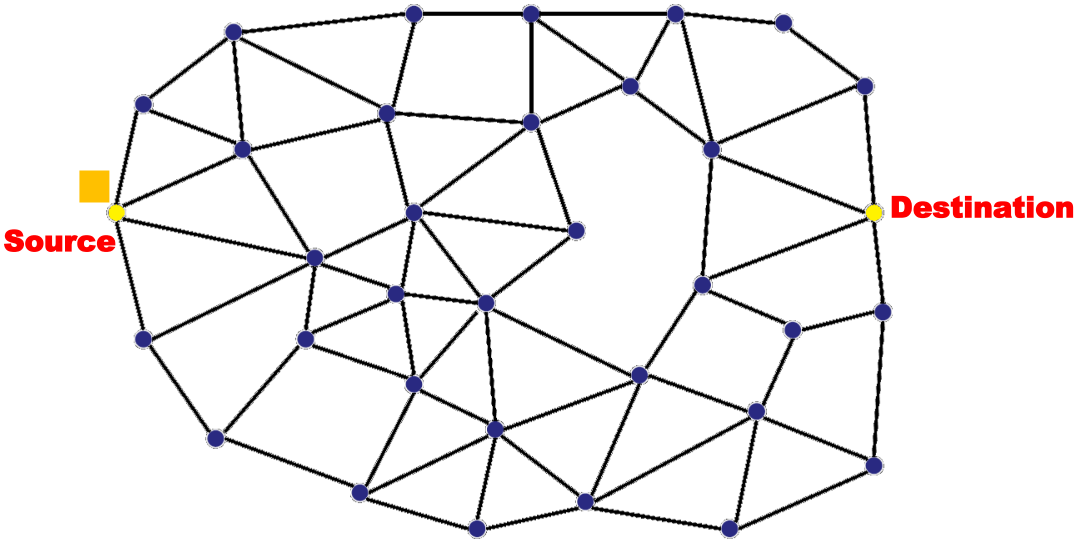
- Nodes have coordinates
- We have a source and a destination



Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (2)

How basic GDSTR works:

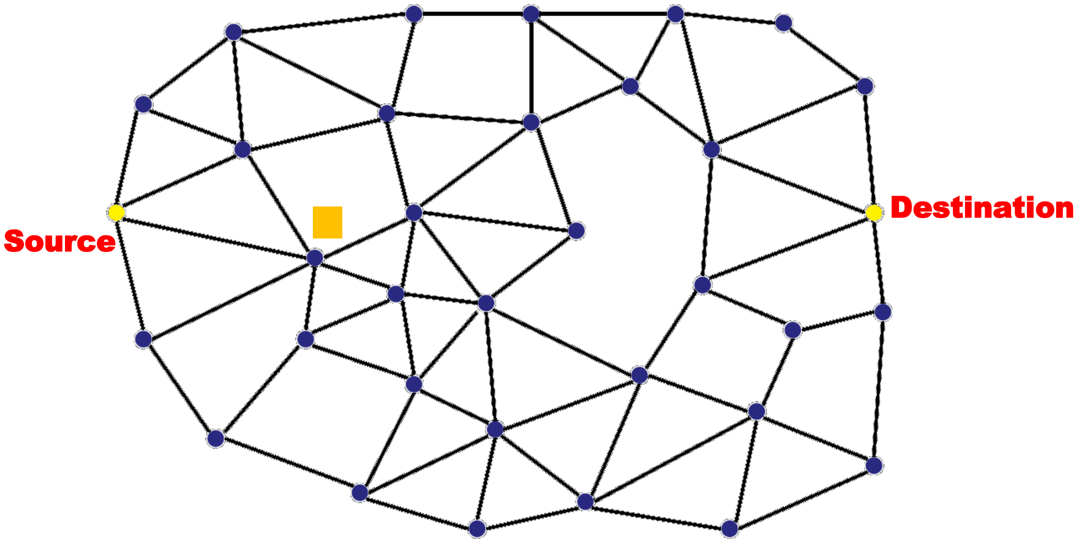
- Nodes have coordinates
- We have a source and a destination
- Packet contains coordinates of destination



Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (2)

How basic GDSTR works:

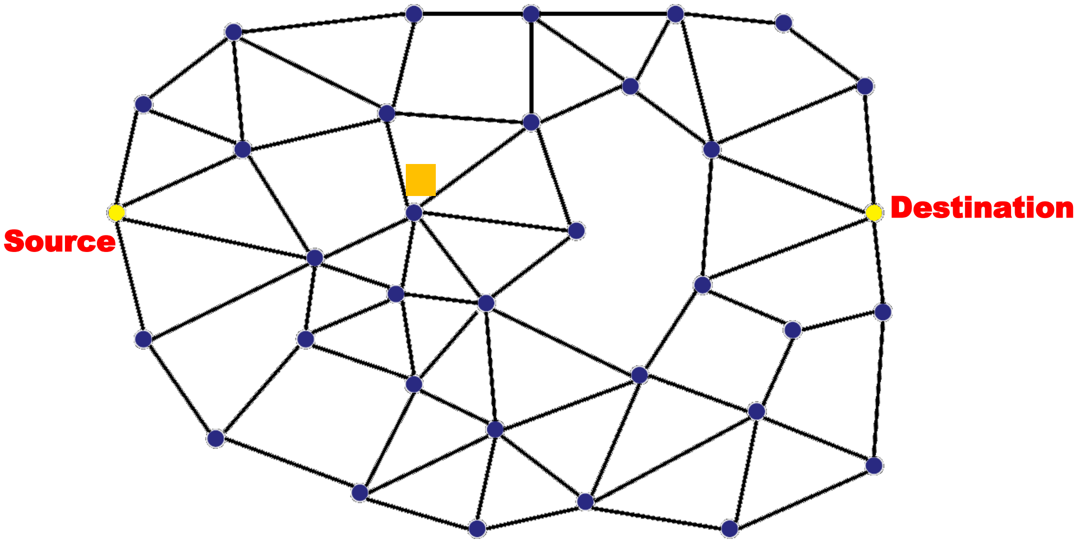
- Nodes have coordinates
- We have a source and a destination
- Packet contains coordinates of destination
- Greedy forwarding (1)



Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (2)

How basic GDSTR works:

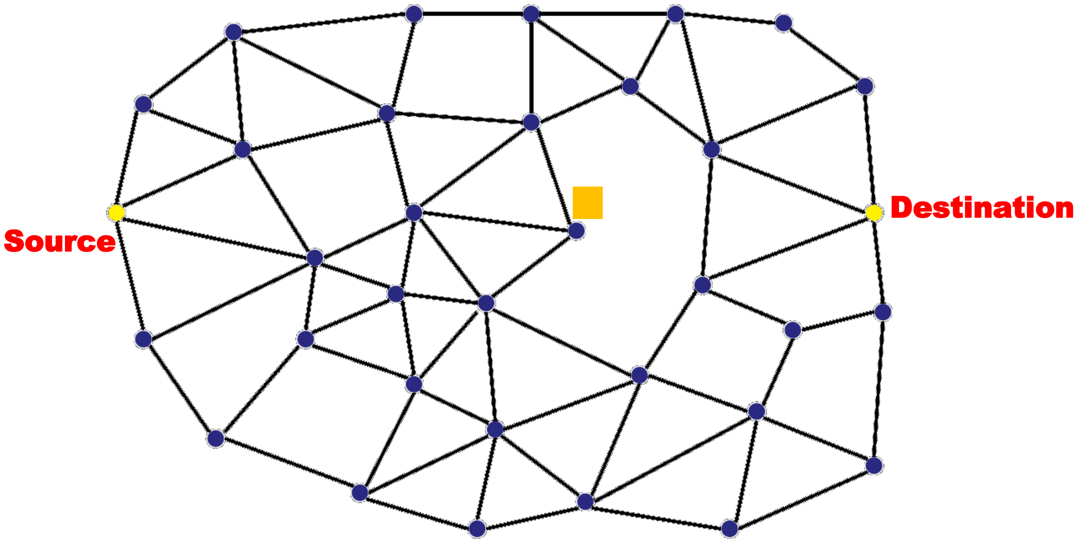
- Nodes have coordinates
- We have a source and a destination
- Packet contains coordinates of destination
- Greedy forwarding (2)



Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (2)

How basic GDSTR works:

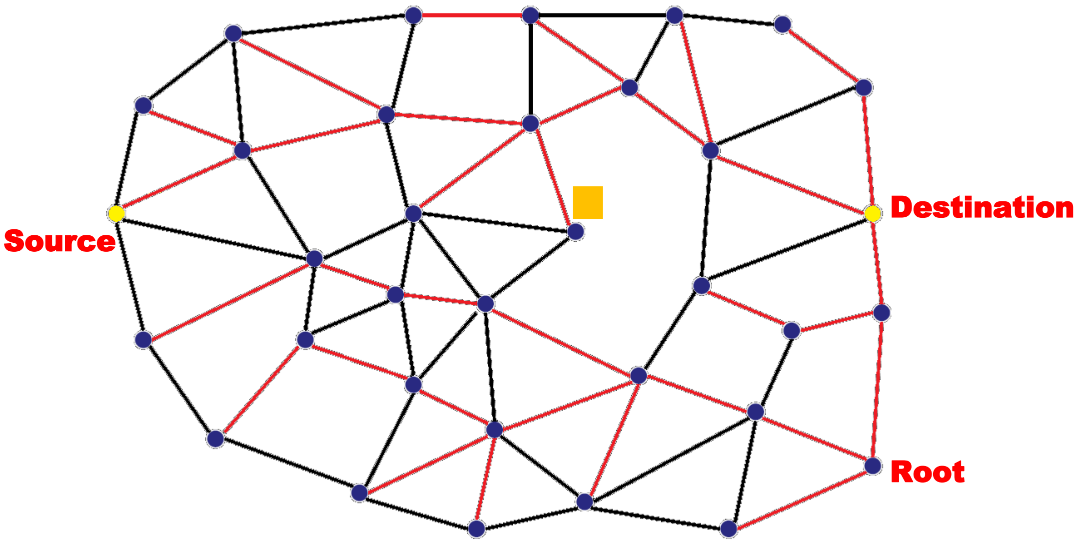
- Nodes have coordinates
- We have a source and a destination
- Packet contains coordinates of destination
- Greedy forwarding (3)
- Local minima 



Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (2)

How basic GDSTR works:

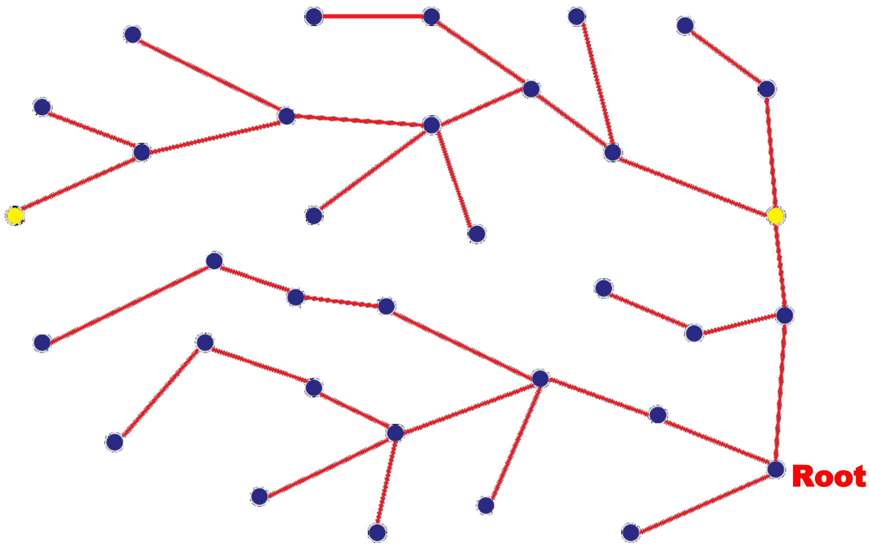
- Nodes have coordinates
- We have a source and a destination
- Packet contains coordinates of destination
- Greedy forwarding
- Local minima 
- Distributed spanning tree



Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (2)

How basic GDSTR works:

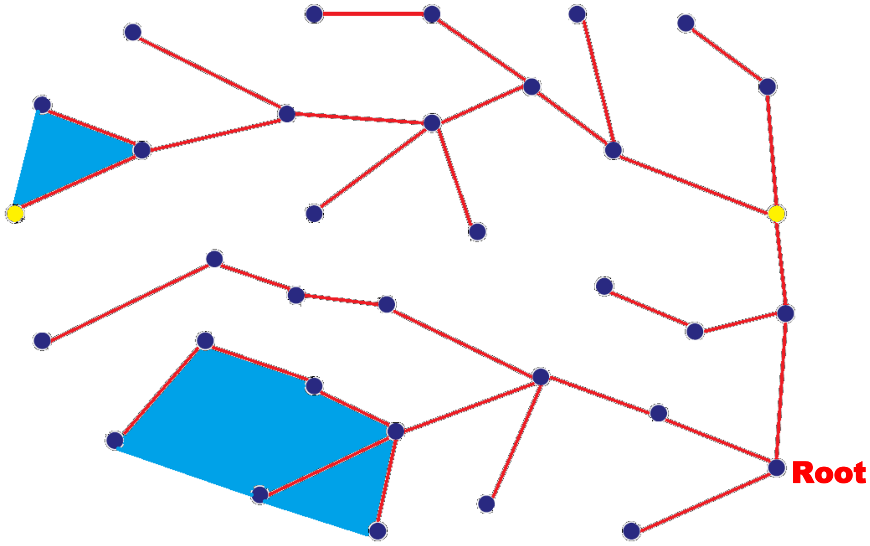
- Nodes have coordinates
- We have a source and a destination
- Packet contains coordinates of destination
- Greedy forwarding
- Local minima 
- Distributed spanning tree
- Aggregate coordinates with convex hulls (1)



Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (2)

How basic GDSTR works:

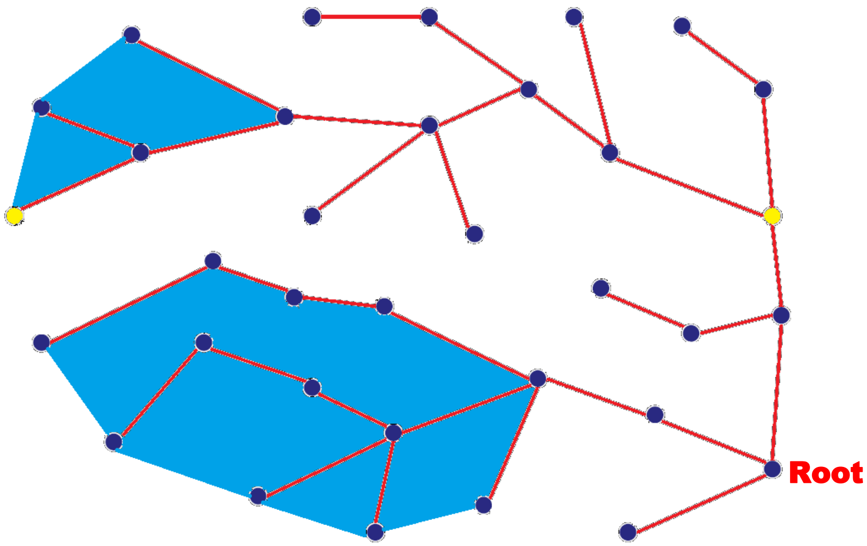
- Nodes have coordinates
- We have a source and a destination
- Packet contains coordinates of destination
- Greedy forwarding
- Local minima 
- Distributed spanning tree
- Aggregate coordinates with convex hulls (2)



Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (2)

How basic GDSTR works:

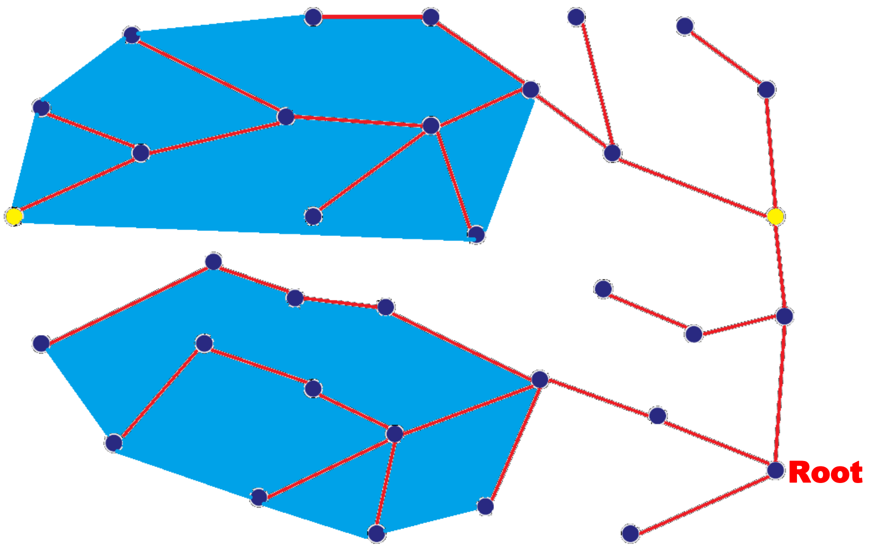
- Nodes have coordinates
- We have a source and a destination
- Packet contains coordinates of destination
- Greedy forwarding
- Local minima 
- Distributed spanning tree
- Aggregate coordinates with convex hulls (3)



Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (2)

How basic GDSTR works:

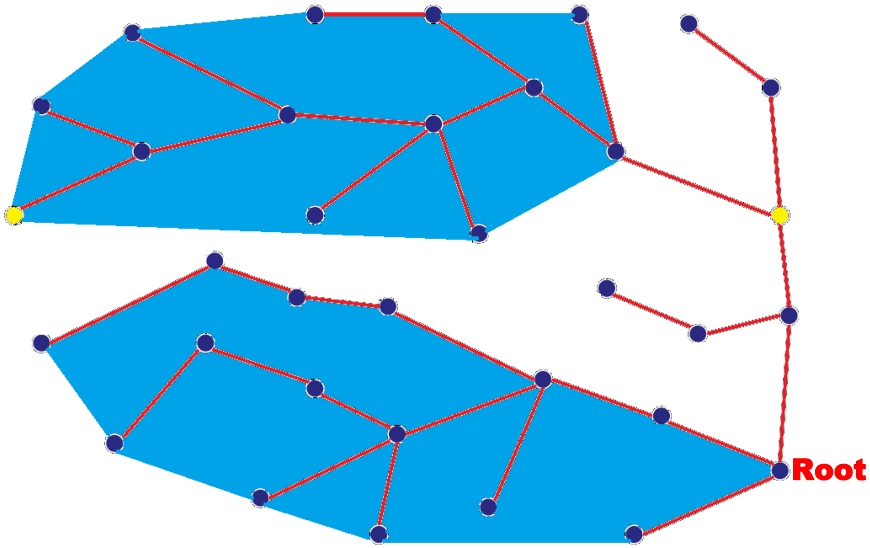
- Nodes have coordinates
- We have a source and a destination
- Packet contains coordinates of destination
- Greedy forwarding
- Local minima 
- Distributed spanning tree
- Aggregate coordinates with convex hulls (4)



Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (2)

How basic GDSTR works:

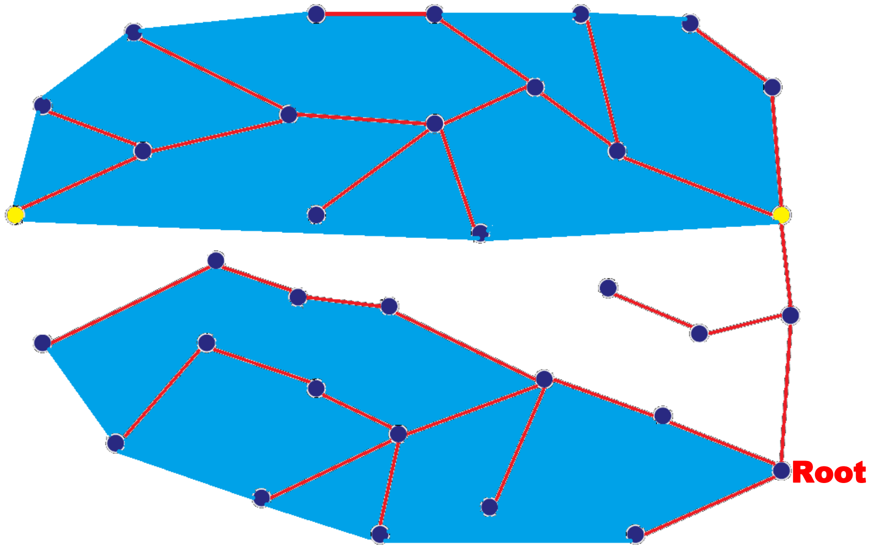
- Nodes have coordinates
- We have a source and a destination
- Packet contains coordinates of destination
- Greedy forwarding
- Local minima 
- Distributed spanning tree
- Aggregate coordinates with convex hulls (5)



Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (2)

How basic GDSTR works:

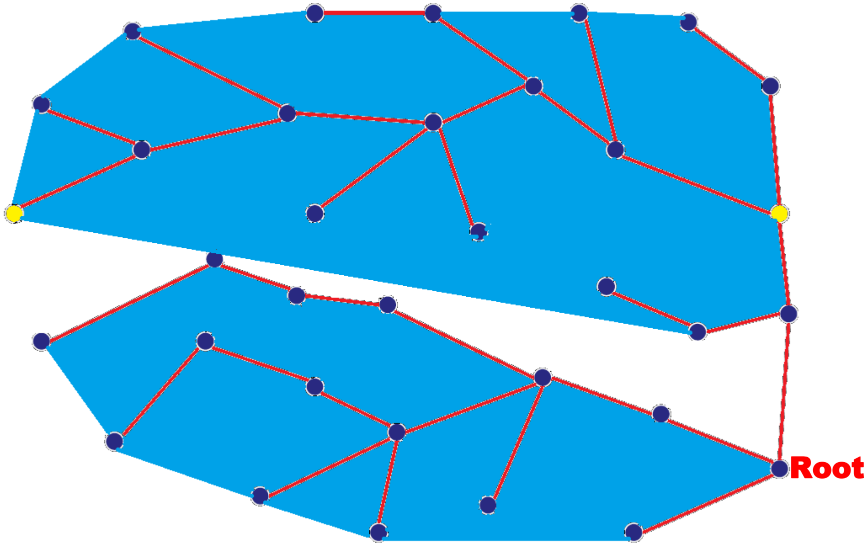
- Nodes have coordinates
- We have a source and a destination
- Packet contains coordinates of destination
- Greedy forwarding
- Local minima 
- Distributed spanning tree
- Aggregate coordinates with convex hulls (6)



Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (2)

How basic GDSTR works:

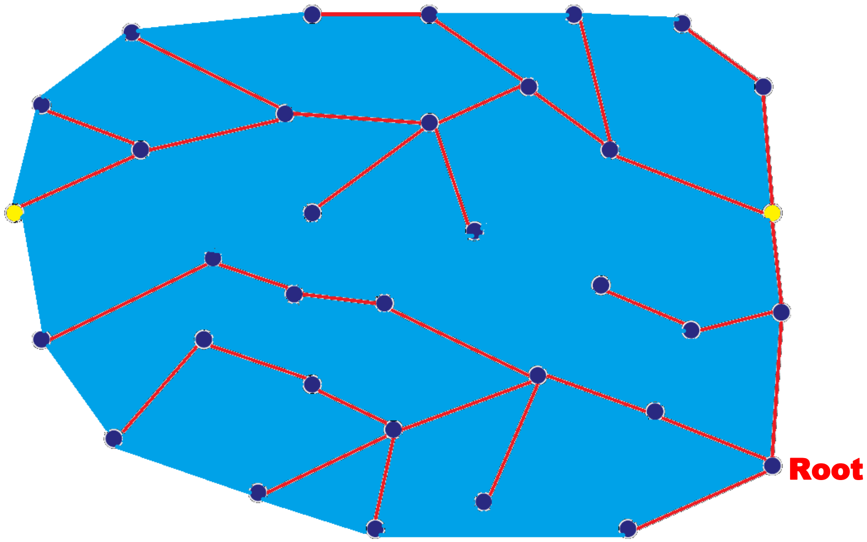
- Nodes have coordinates
- We have a source and a destination
- Packet contains coordinates of destination
- Greedy forwarding
- Local minima 
- Distributed spanning tree
- Aggregate coordinates with convex hulls (7)



Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (2)

How basic GDSTR works:

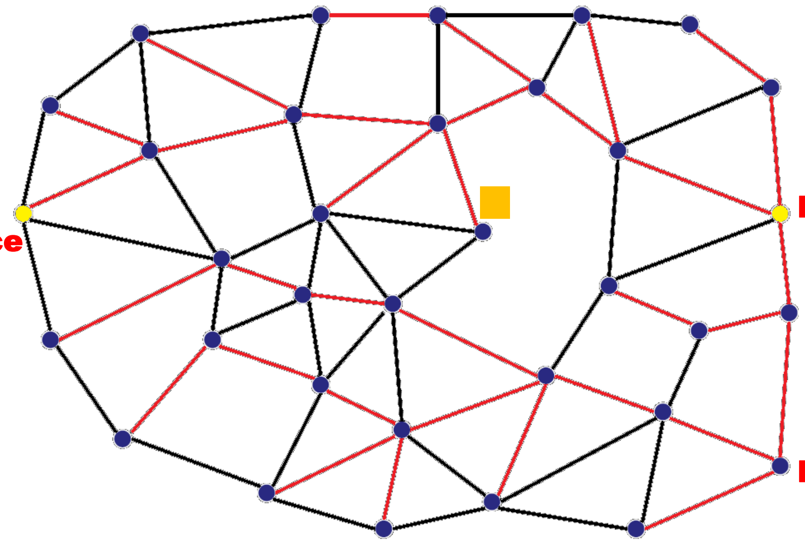
- Nodes have coordinates
- We have a source and a destination
- Packet contains coordinates of destination
- Greedy forwarding
- Local minima 
- Distributed spanning tree
- Aggregate coordinates with convex hulls (8)
- Hull Tree



Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (2)

How basic GDSTR works:

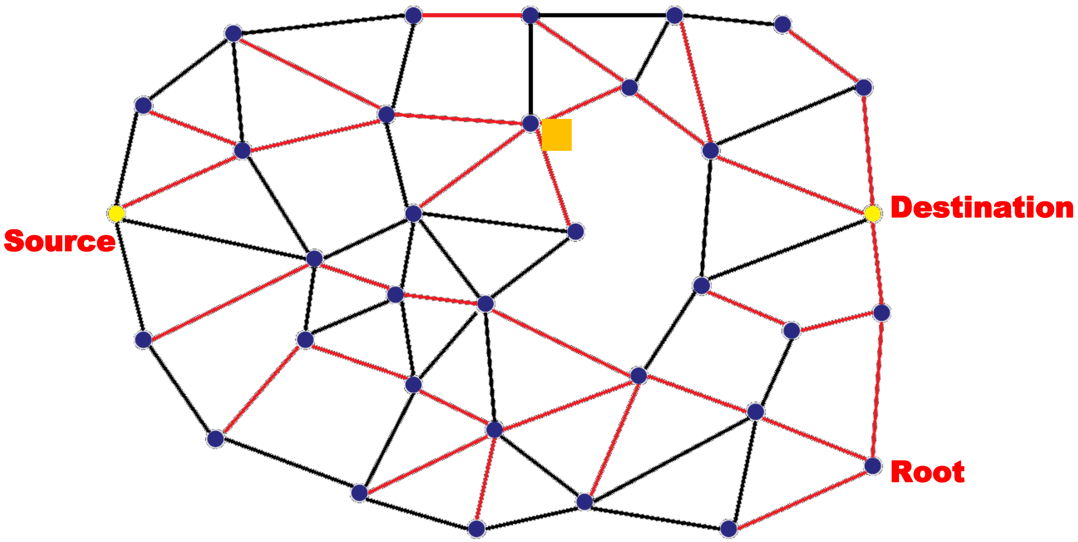
- Nodes have coordinates
- We have a source and a destination
- Packet contains coordinates of destination
- Greedy forwarding
- Local minima 
- Distributed spanning tree
- Aggregate coordinates with convex hulls
- Hull Tree
- Escape from local minima(1)



Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (2)

How basic GDSTR works:

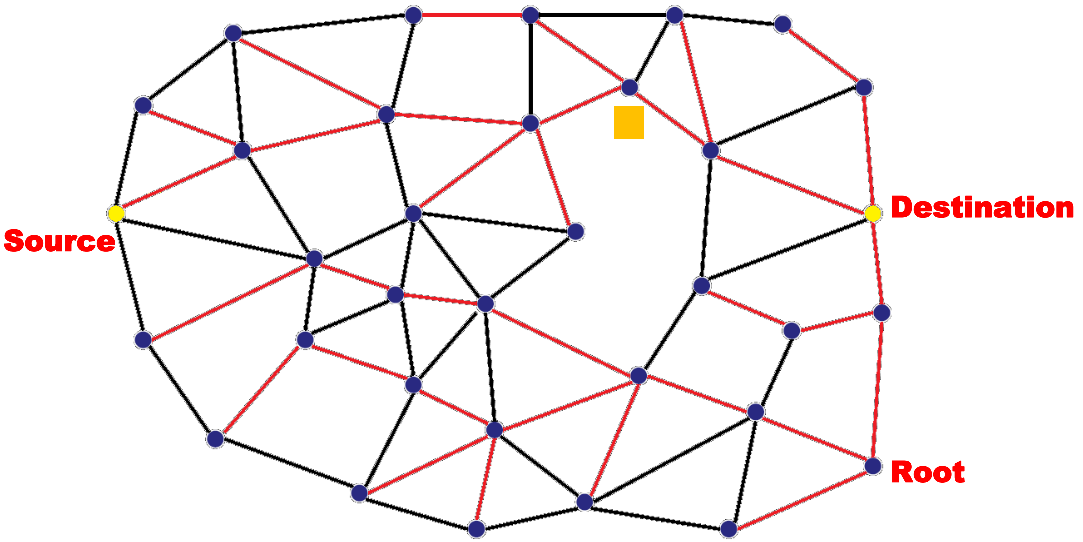
- Nodes have coordinates
- We have a source and a destination
- Packet contains coordinates of destination
- Greedy forwarding
- Local minima 
- Distributed spanning tree
- Aggregate coordinates with convex hulls
- Hull Tree
- Escape from local minima(2)



Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (2)

How basic GDSTR works:

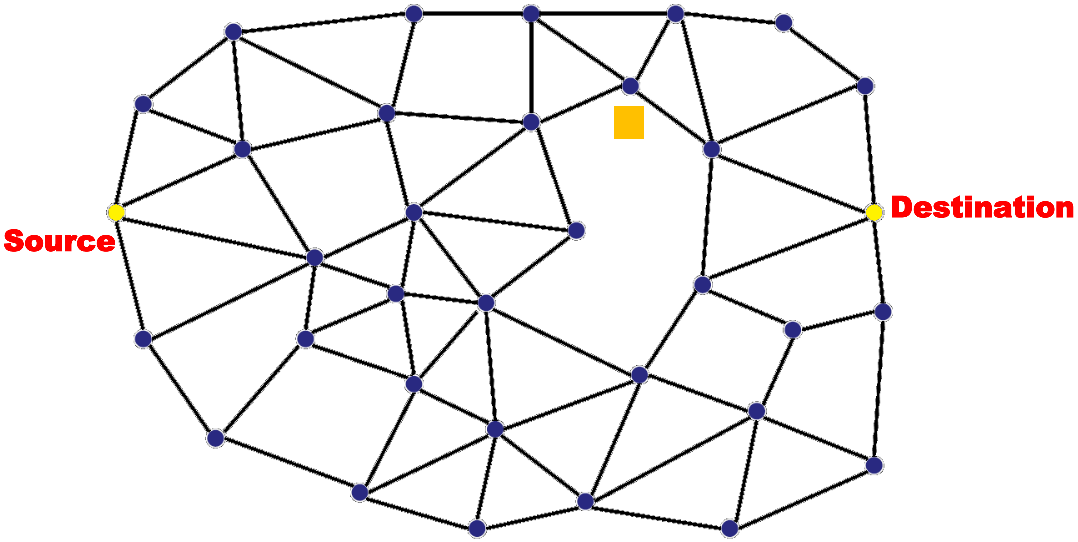
- Nodes have coordinates
- We have a source and a destination
- Packet contains coordinates of destination
- Greedy forwarding
- Local minima 
- Distributed spanning tree
- Aggregate coordinates with convex hulls
- Hull Tree
- Escape from local minima(3)



Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (2)

How basic GDSTR works:

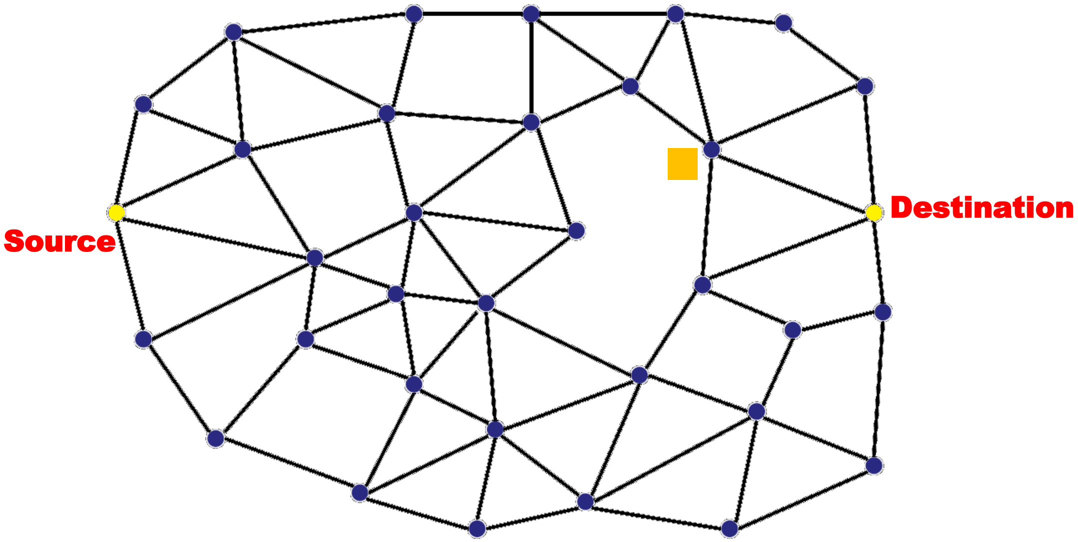
- Back to greedy forwarding (1)



Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (2)

How basic GDSTR works:

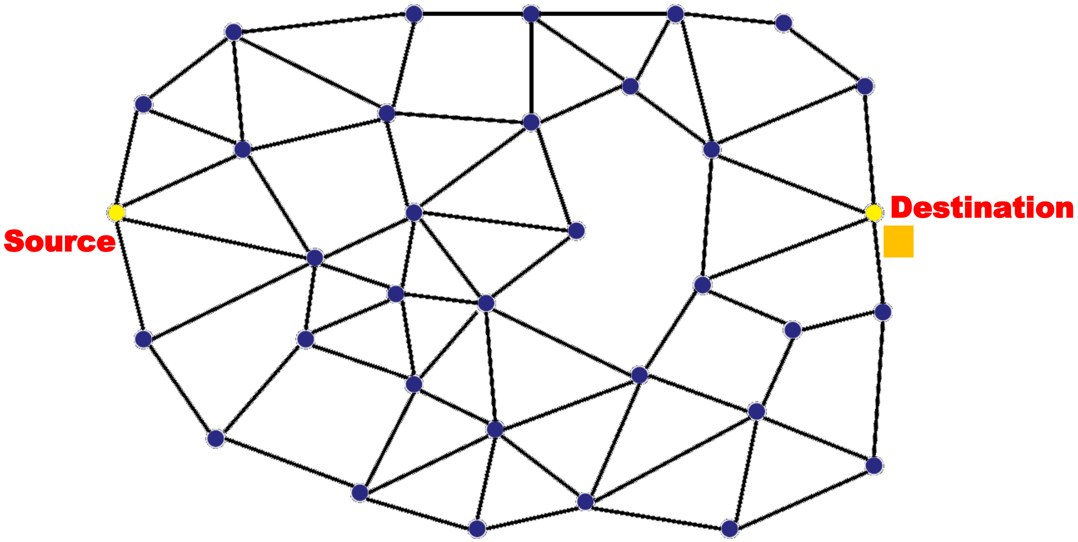
- Back to greedy forwarding (2)



Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (2)

How basic GDSTR works:

- Back to greedy forwarding (3)
- Done!



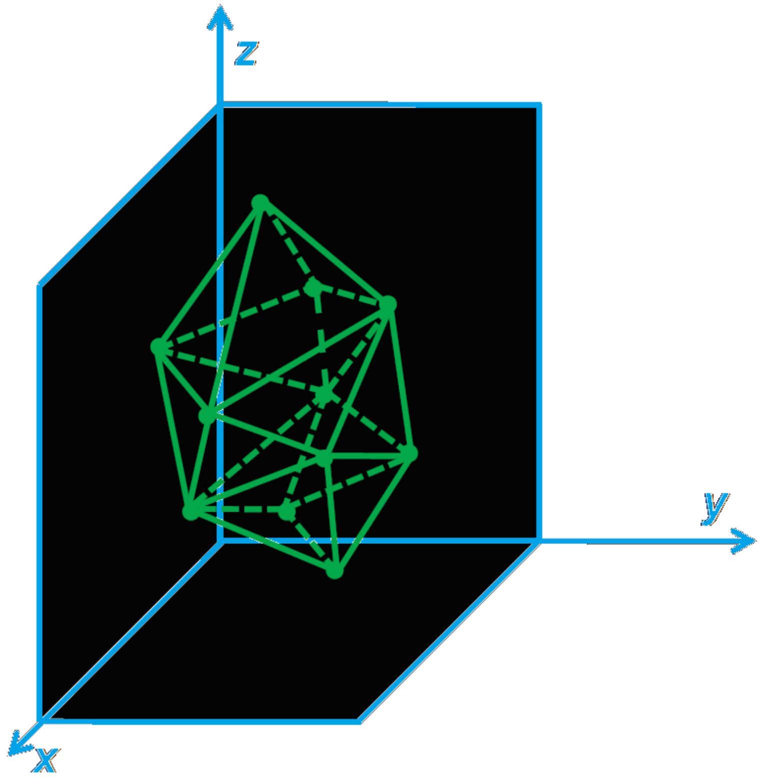
Oh man! Come on tell us about of 3D version. We are getting bored!

Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (3)

How GDSTR-3D works:

- Approximate 3D Convex Hull with 2 x 2D Convex Hull
- Use two-hop greedy forwarding

3D Convex Hull



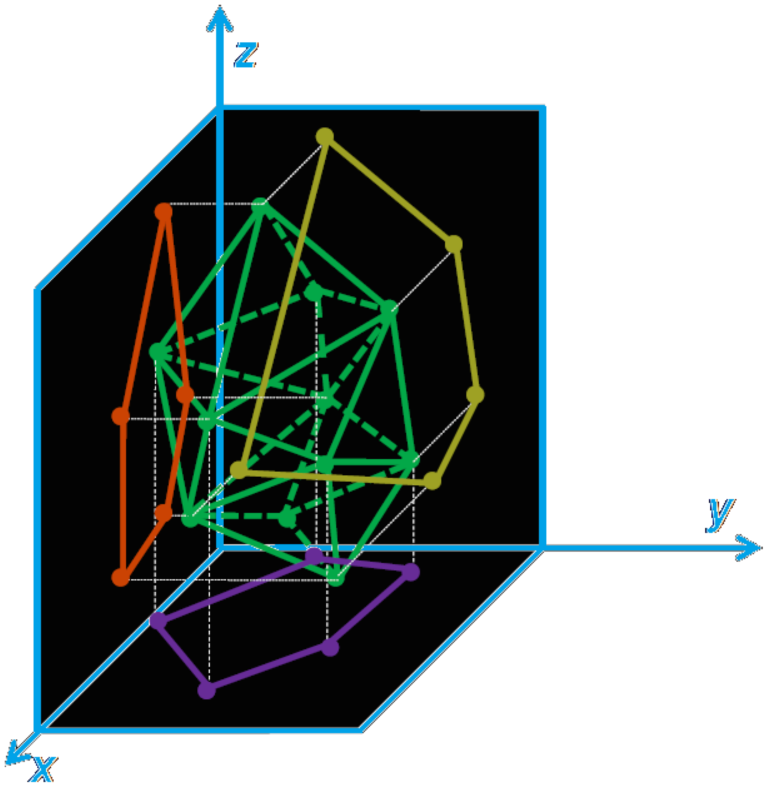
Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (3)

How GDSTR-3D works:

- Approximate 3D Convex Hull with 2 x 2D Convex Hull
- Use two-hop greedy forwarding

3D Convex Hull

- Projection onto orthogonal planes (1)



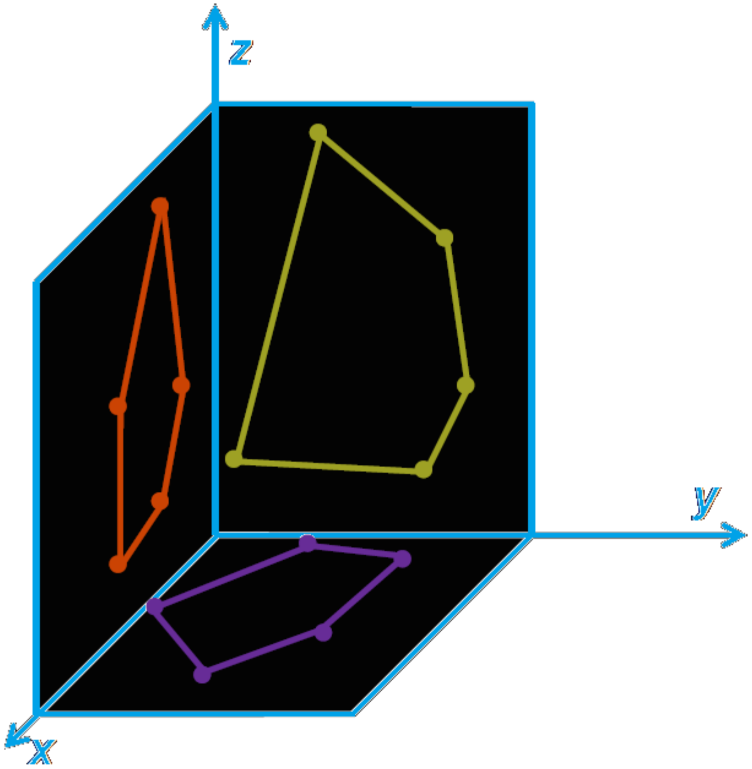
Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (3)

How GDSTR-3D works:

- Approximate 3D Convex Hull with 2 x 2D Convex Hull
- Use two-hop greedy forwarding

3D Convex Hull

- Projection onto orthogonal planes (2)



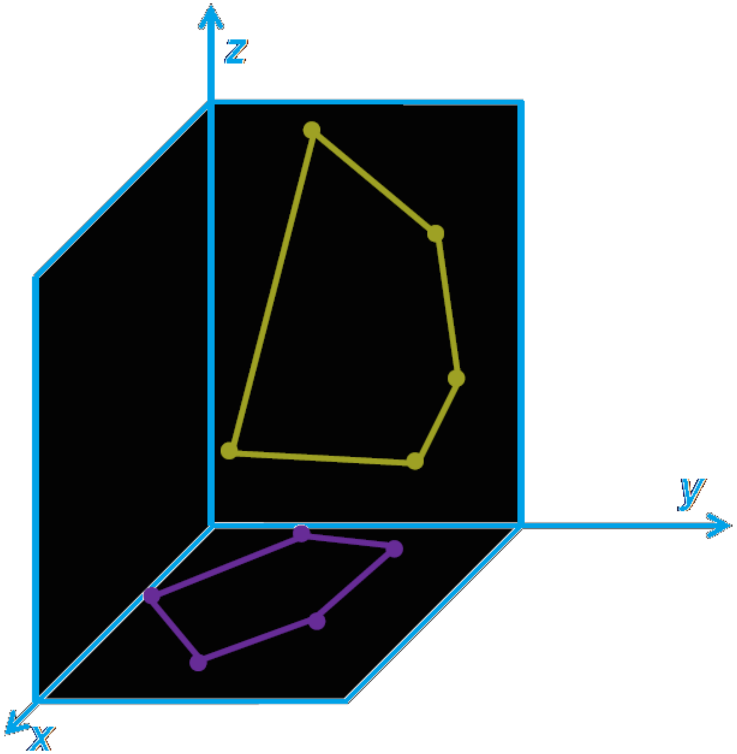
Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (3)

How GDSTR-3D works:

- Approximate 3D Convex Hull with 2 x 2D Convex Hull
- Use two-hop greedy forwarding

3D Convex Hull

- Projection onto orthogonal planes
- Use two 2D convex hulls to approximate the 3D convex hull



Geographic Protocols - GDSTR-3D – Greedy Distributed Spanning Tree Routing (4)

GDSTR-3D in UAVNET:

Pros:

- Take in account 3D position of nodes
- Packet delivery guaranteed
- Hop stretch close to 1

Cons:

- Assumes static topology

Energy Efficiency in UAVNET (1)

Energy aspects in UAV:

- energy consumption can limit the flying time
- large consumption even when there is no transmission, i.e. Power rating of Wi-Fi 802.11n interface, in non-MIMO single antennal mode:
  - 1280 mA in Transmission
  - 940 mA in Reception
  - 820 mA in idle
  - 100 mA in sleep
- battery capacity of typical small drone is 5200mAh, 11.1V. Drone draws about 12.5A with a flying time of about 25 minutes
- communication equipment of the UAV in a mesh network normally receives and transmits continuously

Scenarios of energy for communication:

- same source for powering UAV
- different source (battery pack)

Energy Efficiency in UAVNET (2)

Same source for powering UAV

- flight time reduced by 16% of the rated value
- together with GPS and a couple of sensors, it can easily go beyond 20%

If battery voltage drops down below 11.1V even before the power is fully drained, normal function of the UAV are compromised.

Different source (battery pack)

- battery weight needs to be taken into account in the limited payload capacity of the UAV
- experiment:
  - router required 8 AAA batteries and each weigh about 11.5 g and capacity 860mAh
  - flight duration 25 minutes and consumption 740mAh, with continuous transmission and reception
  - fully charged alkaline cells enough voltage for 8 hours
  - eight cells weigh 92 g. UAV used is about 1 kg and it could carry about 300 grams of payload. The batteries constituted about 30% of the payload that could be put on the UAV!

Energy Efficiency in UAVNET (3)

Because of these problems, different methods of energy conservation are used:

- **Energy Conservation in the Network Layer:**
  - measuring and controlling power utilization
  - distributing load fairly based on some energy metric
  - make some of the nodes to sleep
  - navigate a low energy path
- **Energy Conservation in the Data Link Layer:** method to avoid wastage of energy in the MAC layer due to:
  - collisions
  - overheads
  - listening for potential traffic
  - overhearing traffic meant for other nodes.
- **Energy Conservation in the Physical Layer:**
  - hardware implementation
  - connectivity to neighbors
  - implements encoding and signalling
  - moves the bits over the physical medium

Energy Efficiency in UAVNET - Energy Conservation in the Network Layer

Energy efficient routing important for networks with energy constraints, four categories:

- **Path Selection Based:** *aim to select paths that minimize total source to destination energy requirement*
- **Node Selection Based:** *aim to select nodes that preserve battery life of each node or exclude nodes with low energy*
- **Cluster Head or Coordinator Based:** *selection of a cluster head or a coordinator that will remain awake while the other nodes can sleep to conserve power. Protocols suitable for sensor networks.*
- ***Sleep Based:*** *conserve energy mainly by making as many nodes sleep for as long as possible*

Energy Efficiency in UAVNET - Energy Conservation in the Data Link Layer

Responsibility of the MAC sublayer of the data link Layer to ensure a fair mechanism to share access to the medium among nodes, four categories:

- **Duty Cycle With Single Radio:** *involve use of sleep-activity schedules, only a single radio is used for signaling and data traffic*
- **Duty Cycle With Dual Radio:** *involve sleep-activity schedules. They use separate radios for signalling and data traffic*
- **Topology/Power Control Based:** *adjust transmit power levels to control the connectivity to other nodes or select topology by deciding which nodes to keep on*
- ***Cluster based MAC Protocols:*** *divide nodes into groups and select a cluster head or a coordinator for inter-area communication and data aggregation. Non-cluster nodes can then have power saving sleep schedules.*

Energy Efficiency in UAVNET - Energy Conservation in the Physical Layer

The physical layer deals with modulation and signal coding and related signal transmission technologies, four categories:

- **Dynamic Voltage Control:** *adjust circuit bias voltage levels, depending on the traffic, to optimize energy usage.*
- **Node Level Power Scheduling:** *node level optimizations to achieve energy efficiency and enhance network lifetime*
- **Choosing Minimum Subset:** *choosing a subset of nodes to switch on and keep the other nodes switched off in order to save energy*
- ***Sleeping Schedules With Buffering:*** *aim to prolong sleep times by buffering packets that arrive while a node is inactive*

References

- “Survey of Important Issues in UAV Communication Networks” Lav Gupta, Senior Member, IEEE, Raj Jain, Fellow, IEEE, and Gabor Vaszkun, Second Quarter 2016
- Zone Routing Protocol (ZRP) - Nicklas Beijar Networking Laboratory, Helsinki University of Technology P.O. Box 3000, FIN-02015 HUT, Finland Nicklas.Beijar@hut.fi