OLSR: Optimized Link State Routing

Leonardo Maccari

leonardo.maccari@unitn.it

28/5/2012

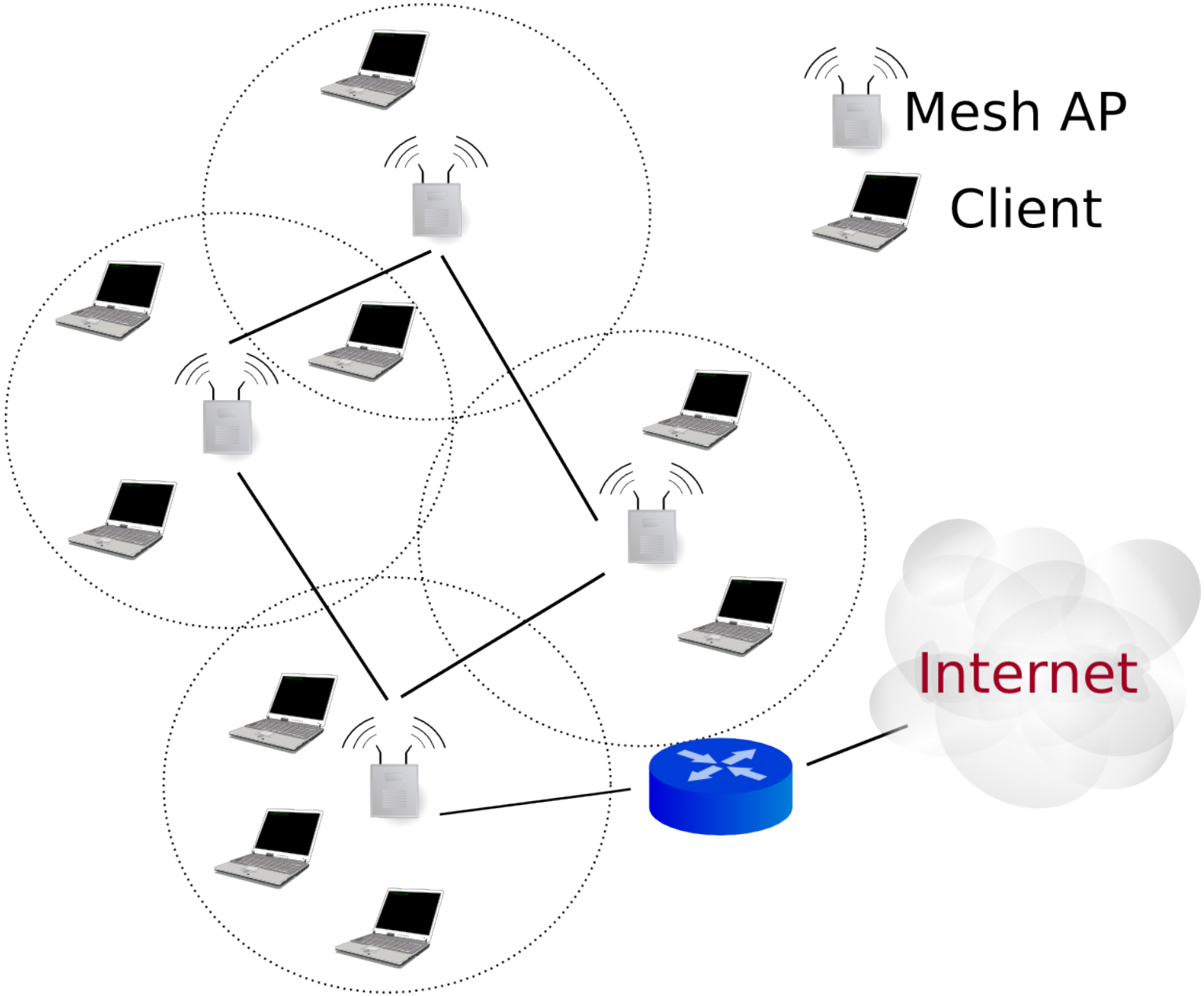
Contents

Wireless multi-hop distributed networks

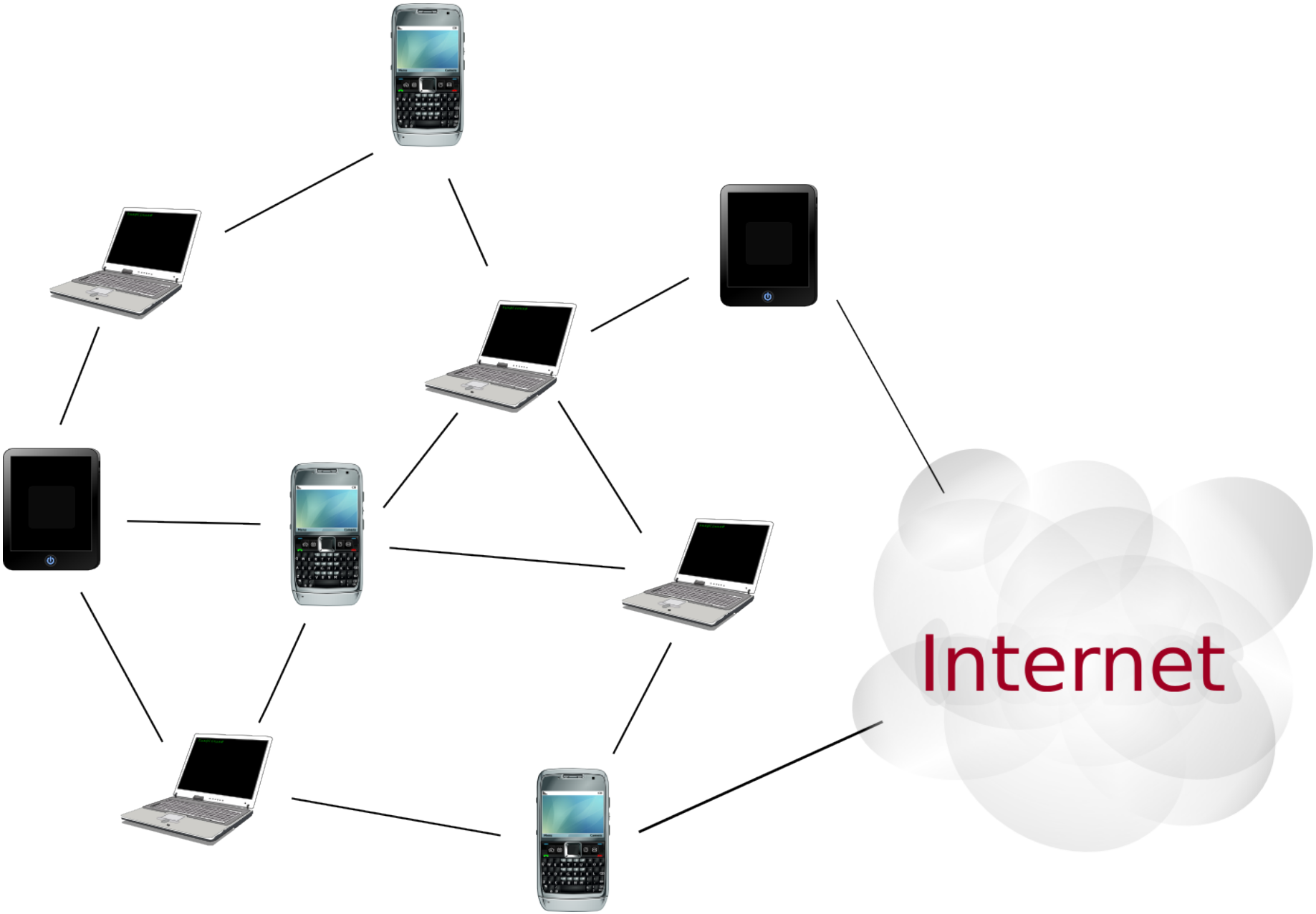
Generally two classes of networks are considered

- Wireless Mesh networks
- Wireless Ad-hoc networks

Wifi Mesh network



Wireless Ad-hoc networks



Features

Relevant features of Wireless Distributed Networks

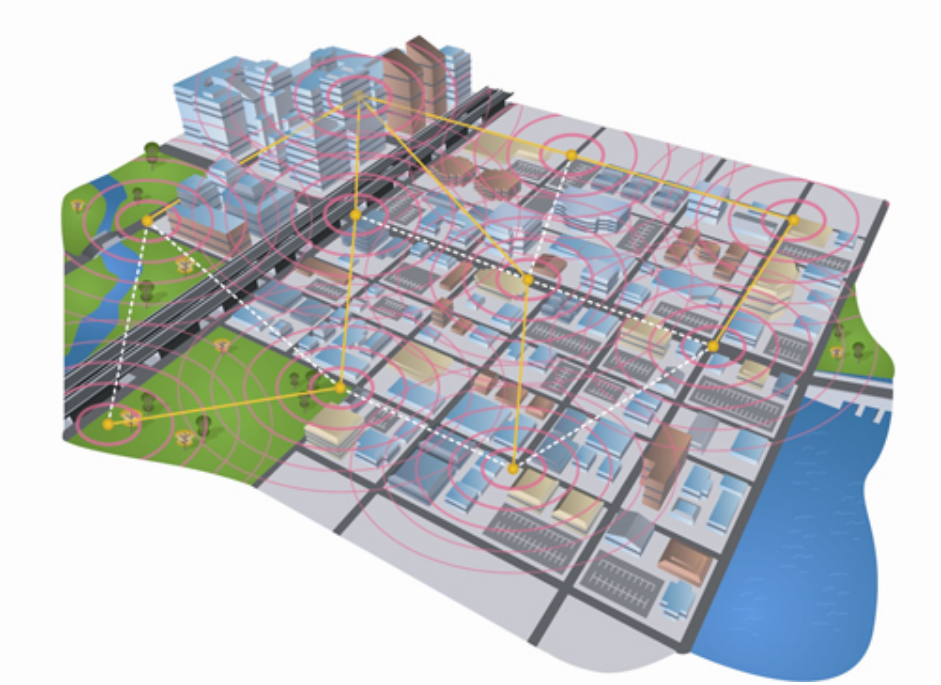
- Multi-hop: cooperate in routing to reach the destinations
- Scalability: size can be adjusted as needed
- Automatic organization: no need for pre-deployment work
- Support for mobility of nodes and clients
- Automatic reconfiguration: recovery from loss of links or failure of nodes
- Energy aware: try to limit energy consumption

When is a Wireless Distributed Networks useful?

- . . . basically, when there is nothing better to use . . .
- . . . but this happens in many circumstances.

1) Metropolitan Internet connectivity

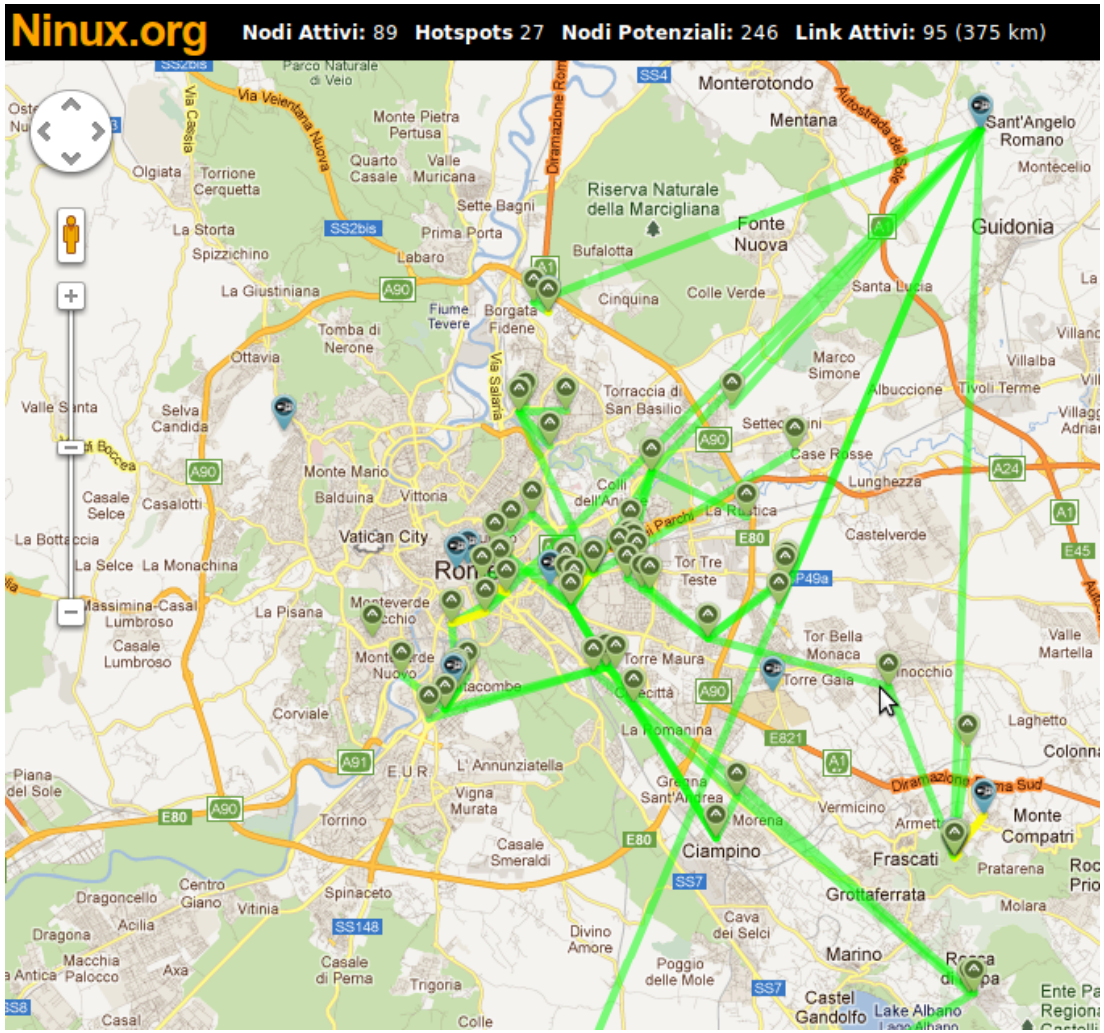
Bring connectivity when there is no other technology (Cisco does it).



- keep in mind the problems: capacity, traffic type, etc. . .
- it is quite hard to use a Wireless Distributed Networks as a generic connectivity carrier and you have to compete with more established approaches.

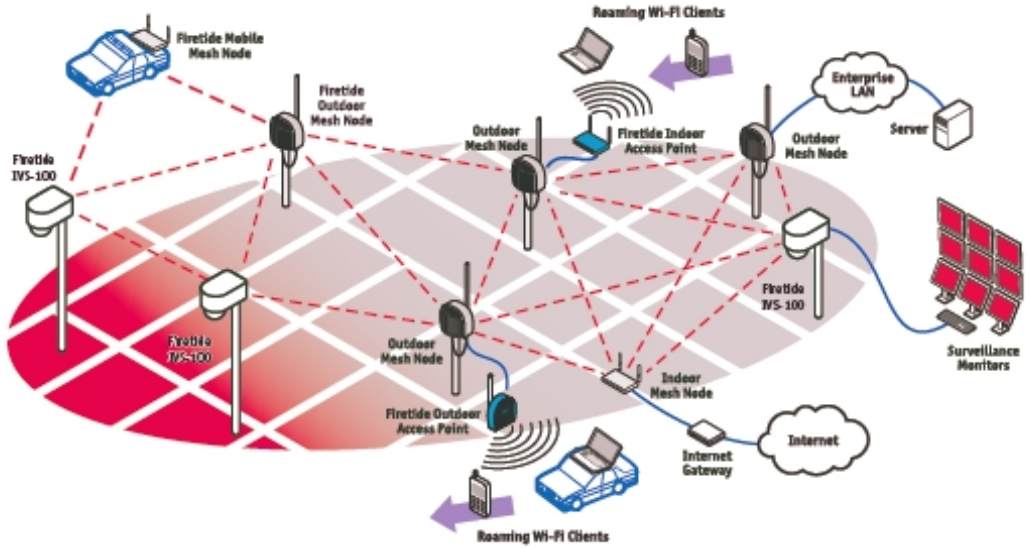
2) There's *nothing better*:

Community networks:



2a) More specific applications:

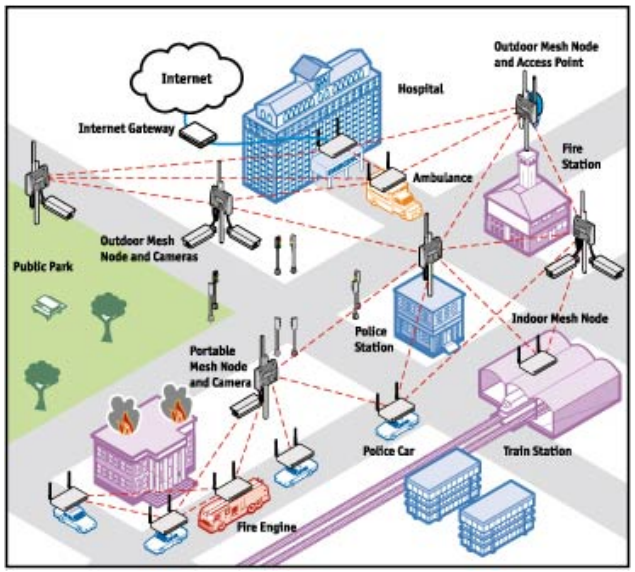
surveillance networks (Firetide does it).



Firetide IVS-100 wireless video surveillance network

3) There's simply *nothing else*: hostile scenarios

- An emergency network (a lot of people does this):

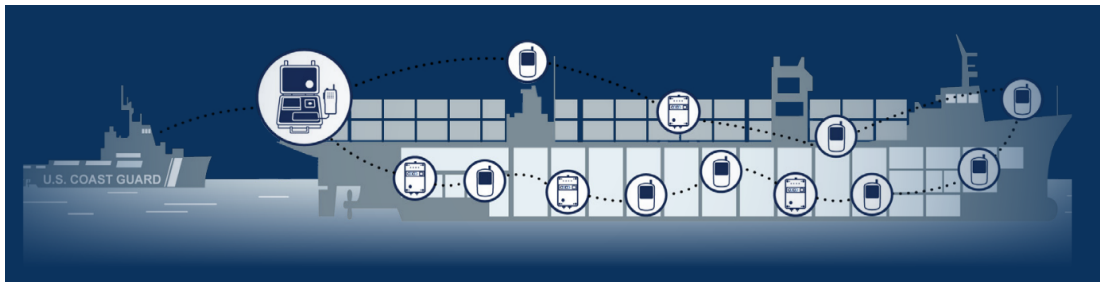


Hostile scenarios (2)

- Police/Safety networks (Selex Comms does this):

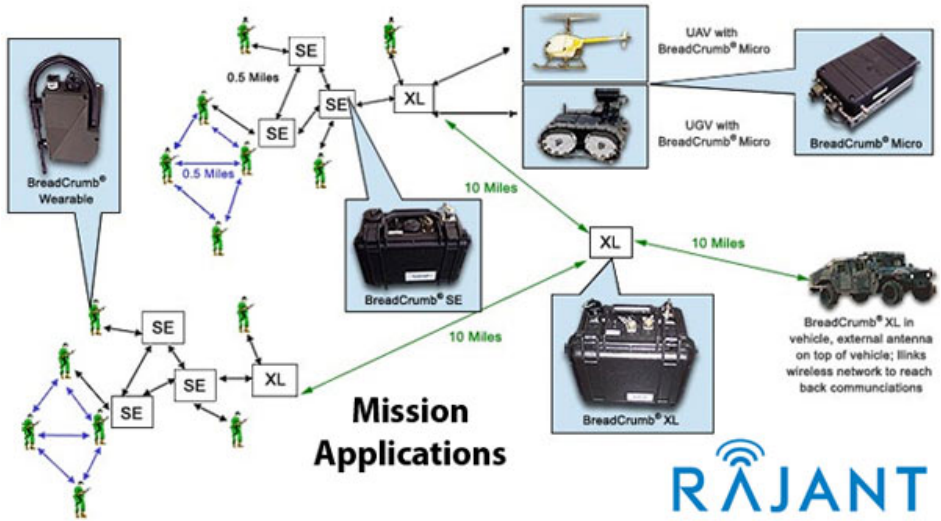
Hostile scenarios (3)

Military networks (Cococorp does it):



Hostile scenarios (4)

Military networks (Rajant does it):



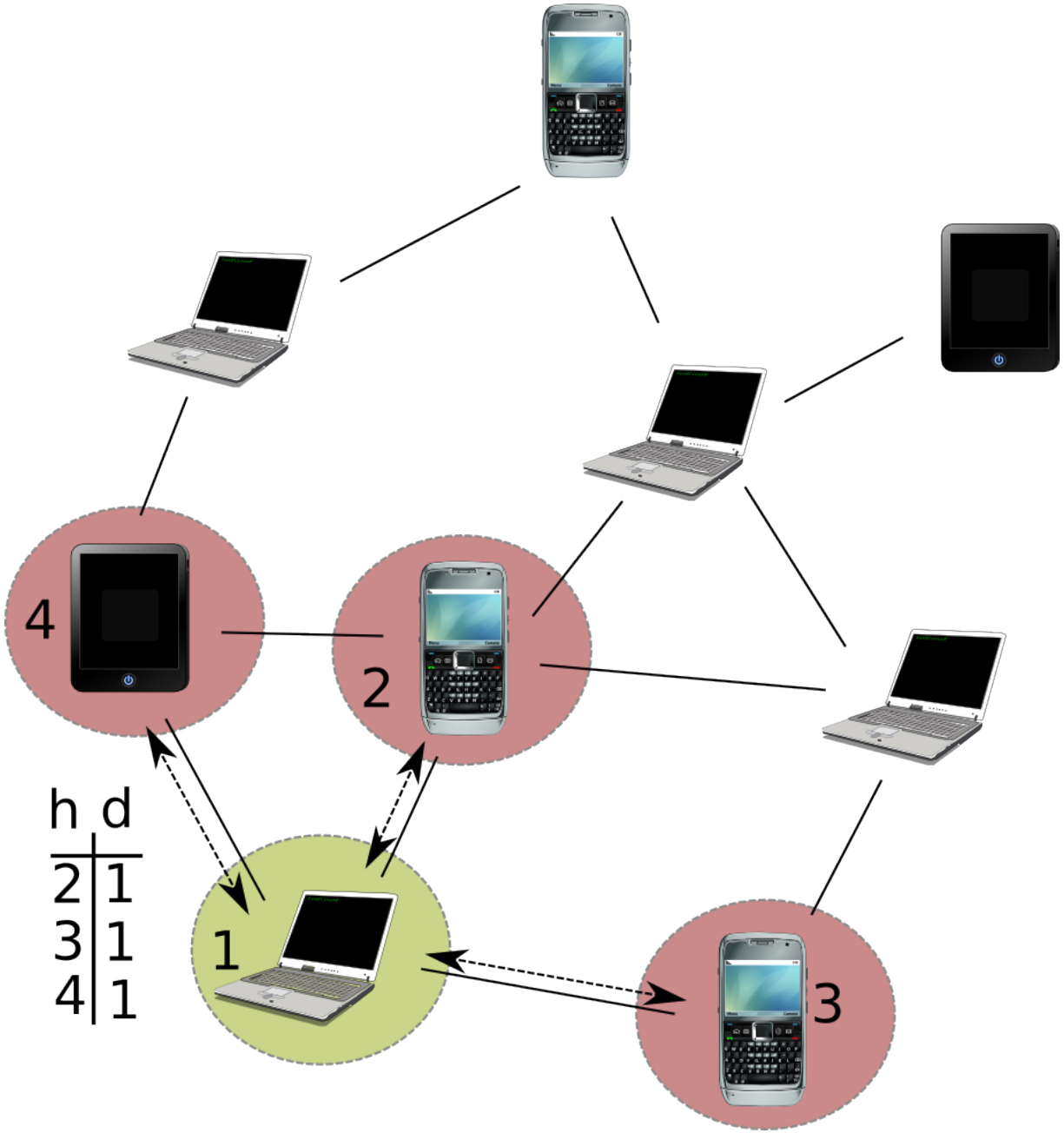
Proactive routing

Proactive routing protocols are protocols that distributes the knowledge necessary to build a RT before it needs to transmit any traffic. If you assume wireless links to be symmetric the following approach can be used.

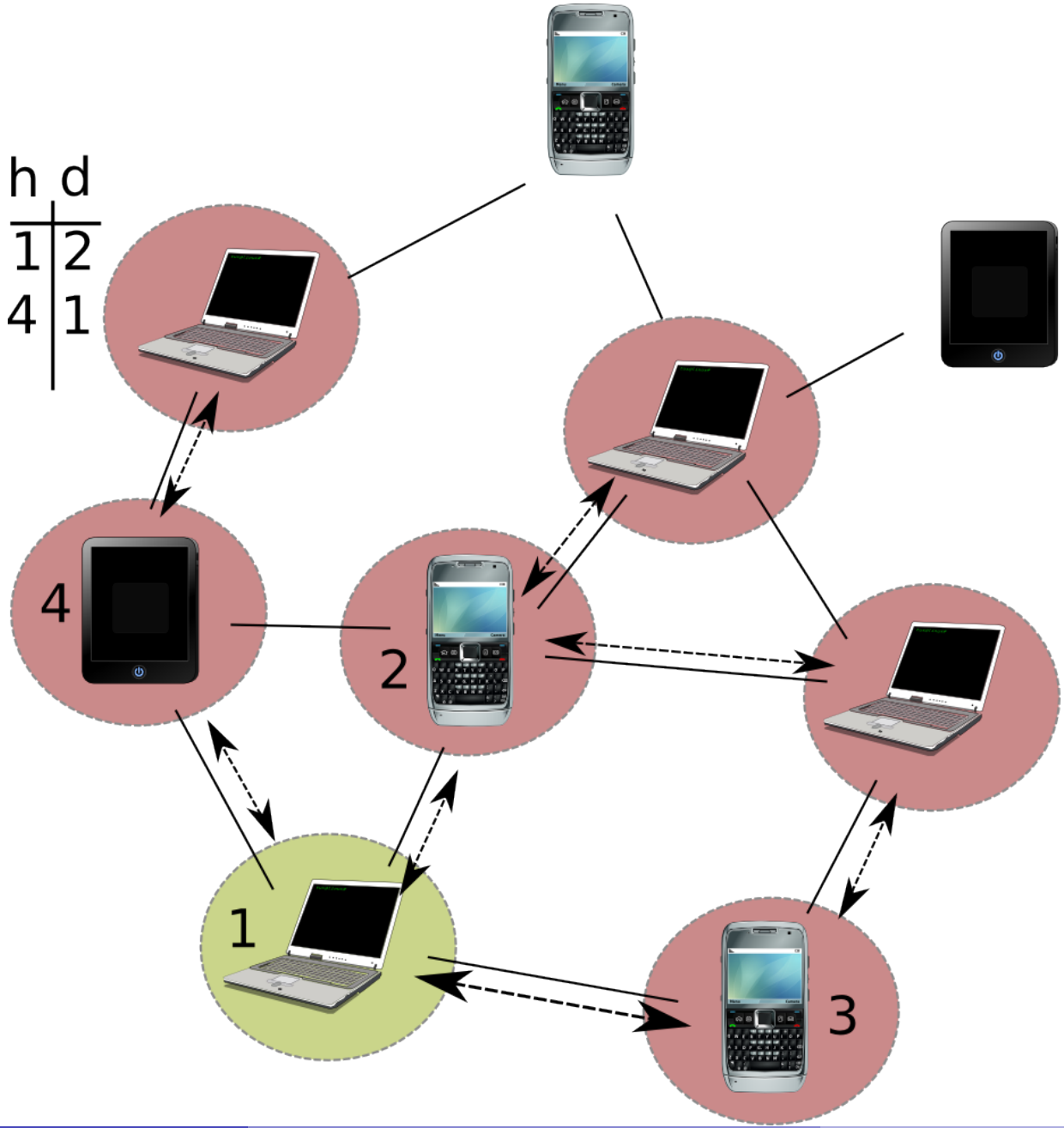
Proactive routing - Naive approach

- Each node at short intervals broadcasts its presence and address with TTL=1 (HELLO messages) $\rightarrow$ local neighborhood is known in advance to every node
- Each node at larger intervals broadcast its local topology (Topology Control (TC) messages) with large TTL, each topology packet is rebroadcasted by all the nodes in the network (with duplicate detection) $\rightarrow$ every node in the network knows all the other nodes and the links between them
- Every node can build an Adjacency Matrix and calculate a full RT to any destination

Ad-hoc proactive Routing: HELLO messages



Ad-hoc proactive Routing: TC messages



Critical Issues

Criticalities

- Wireless broadcast links are not symmetric
- The generated control traffic is high

Issues: traffic generation

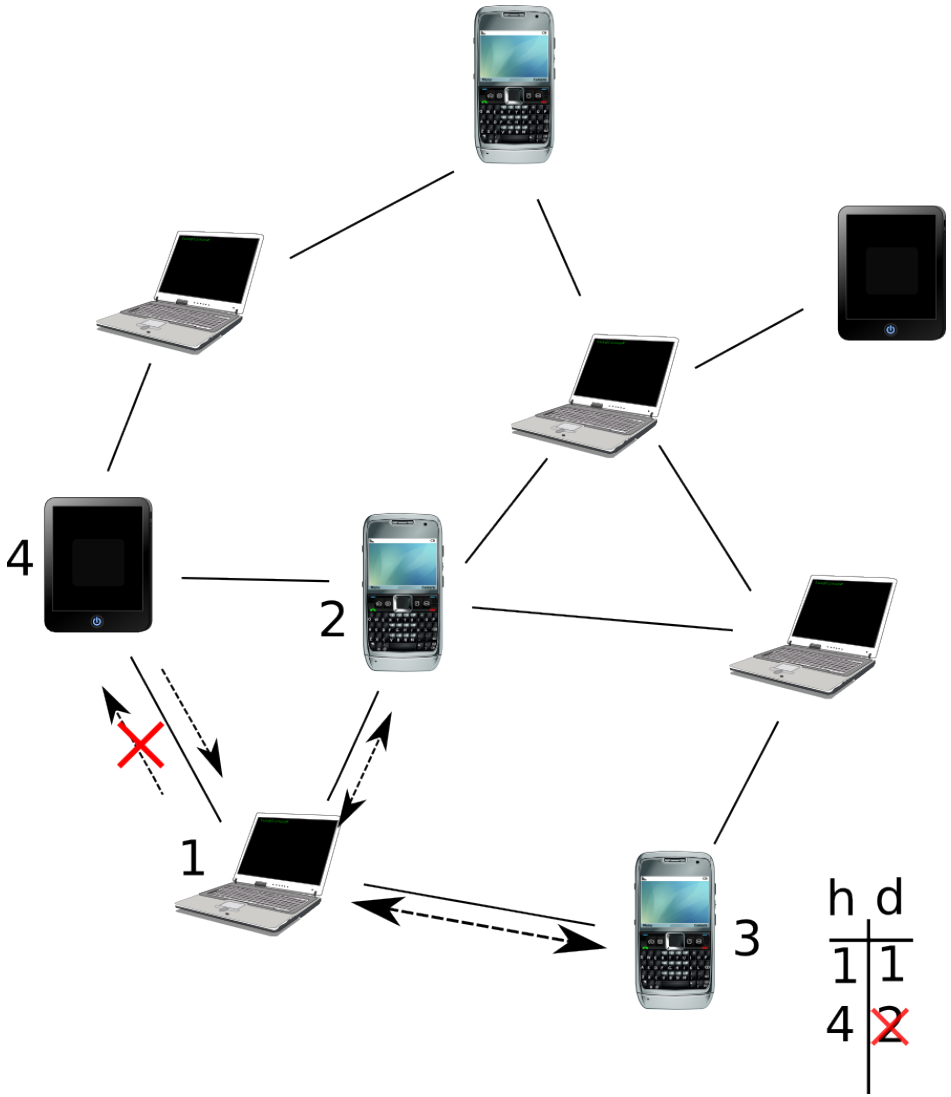
Each node of a network with N nodes with an average n neighbors periodically generates:

- Hello messages: few bytes containing only the interface address (size s) with interval t (defaults to 2 sec).
- TC messages: contains all the set of all the neighbors (size $s * n$) at interval T (defaults to 5 sec), reforwarded by every node in the network (multiply by N).

A total of $s * N$ bytes per each t plus $(s * n) * N^2$ bytes per each T. If $s = 4$, $N = 50$, $n = 5$ this generates approx 10kBs in the whole network plus protocol overhead, mostly generated by the second component. This happens even if there is no single application packet generated in the network.

Issues: Asymmetric links

Broadcast packets are not acknowledged (at least not in WiFi), this can generate wrong RT



Optimized Link State Routing: RFC 3626, October 2003

OLSR is a routing protocol that tries to solve the previous issues identified in the naive routing protocol described so far. Its two main features are:

- Link state detection using HELLO messages to avoid asymmetric links
- Proactive control message diffusion using MPR nodes to reduce control messages

Some definitions:

N: number of nodes in the network

n_i: set of 1-hop neighbors of node i

n_i^2: set of 2-hop neighbors of node i

m_i: set of MPR nodes chosen by node i

Link Sensing

- To avoid broken links each node will include in HELLO messages its full 1-hop neighbor set.
- For each neighbor in the HELLO a status is added that may be SYM/ASYM
- When node *i* receives an HELLO by neighbor node *j*, *i* will include *j* in its HELLO messages as an ASYM neigh.
- If *j* is able to receive HELLOs from *i* it will do the same
- When *i* receives HELLO messages from *j* containing address of *i* in the neighbors, *i* will sponsor *j* as a symmetric neighbor. *j* will do the same.
- There are hysteresis mechanisms in order to avoid wireless fluctuations.

Link Sensing

OLSR Step 1

Every node must sponsor in HELLO messages its complete neighborhood specifying the kind of neighbor (SYM/ASYM). From now on we will simply call neighbors only the SYM neighbors.

Two main consequences:

- The HELLO messages are bigger than before
- Using HELLO messages each node knows not only its 1-hop neighborhood but its 2-hop neighborhood

multipoint relays: MPR nodes

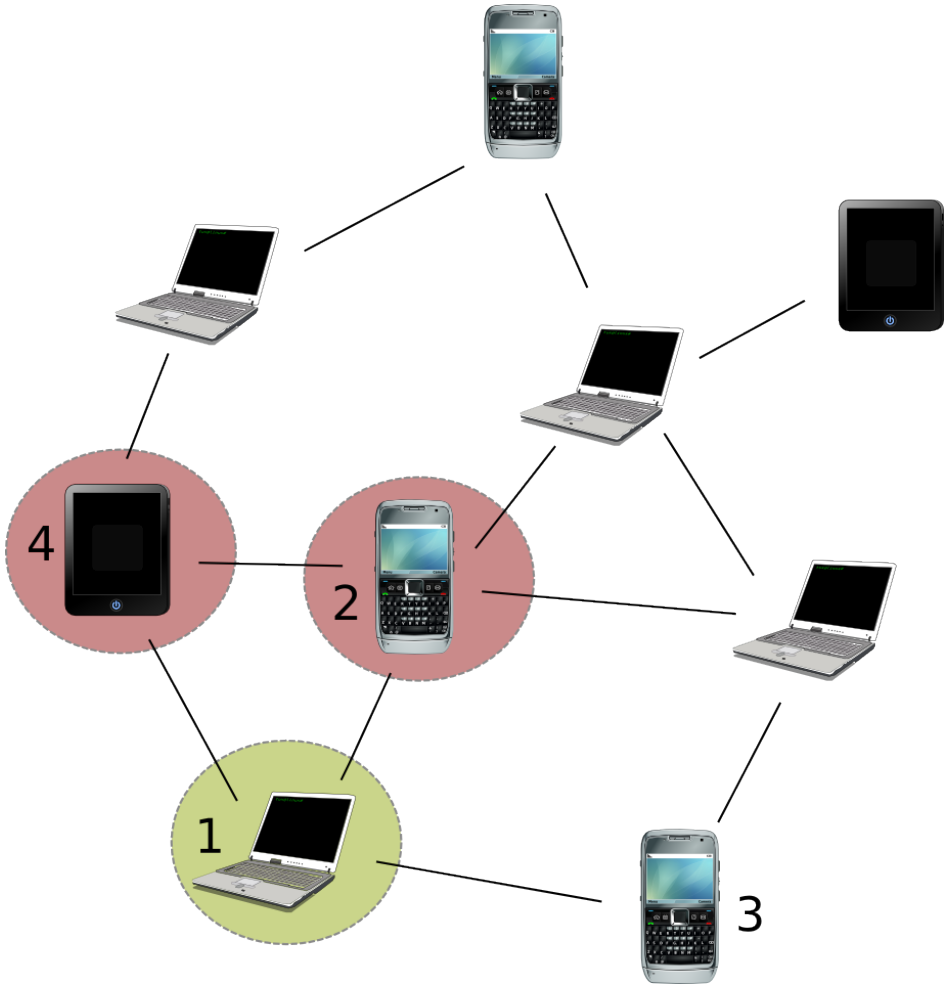
MPR set definition

The MPR set m_i of a node i is an arbitrary subset of its symmetric 1-hop neighborhood n_i which satisfies the following condition: every node in the 2-hop neighborhood n_i^2 of i must have at least a symmetric link towards a node in m_i

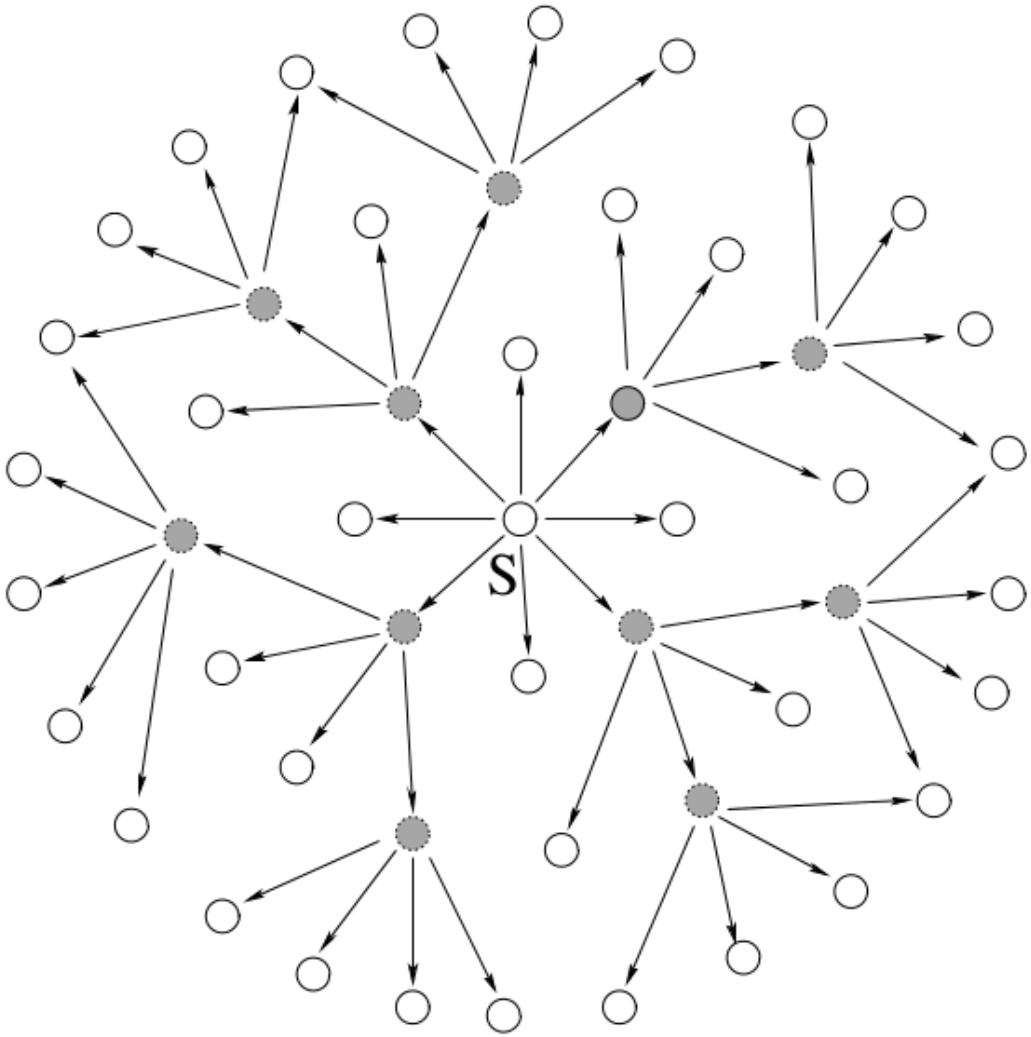
- Basically, m_i is a subset of the neighbors of i that can be used to reach any node in n_i^2 (i.e. they *cover* all the nodes in n_i^2)

MPR

In the example, node 2 and 4 are MPR for node 1 and node 1 is a *selector* for node 2 and 4. Node 3 is not necessary to cover n_1^2.



MPR



MPR properties: 2-hop broadcast

When node i wants to broadcast a message to all the nodes in n_i^2, not all his neighbors need to rebroadcast it. If only nodes in m_i rebroadcast the message the effect is the same with less retransmissions. The smaller m_i compared to n_i^2 the less packets are sent.

OLSR Step 2

Each node must select a subset of N_i^2, as small as possible. In the HELLO messages, the state of each neighbor contains another bit that is MPR/NOT_MPR so each MPR knows its selector set. Only the MPR nodes participate in flooding messages rebroadcasting the packets from their selectors.

MPR properties: network-wide broadcast

- It is easy to show that if every node follows step 2 then if node i sends a network-wide broadcast control packet M (i.e. a TC message), if there are no losses every node j receives M:
  - By contradiction, if node j did not receive a packet, none of n_j has retransmitted it. This implies that any node k did not select as MPR a node in $n_j \cap n_k$. This, in turn implies that no node is a 2-hop neighbor of j, so j is out of the network or it is a 1-hop neighbor of all the other nodes, so he must have received the packet.

  Step 2 guarantees broadcast communication to all the network using only a subset of nodes, the traffic sent each interval T is reduced (approximately) from $(s * n) * N^2$ to $(s * n) * M * N$ where M is the number of MPR nodes in the network ($M \leq N$).

MPR properties: shortest paths to any MPR

Shortest paths across MPR nodes.

- Choosing MPRs does not change the connectivity of the network. If j is in n_i^2 its distance from i remains the 2 for any choice of m_i. Every node in the network is a selector of at least one MPR (excluding trivial topologies)
- Consequently if a generic node k receives a TC message from a node i on a path longer than 2 hops, it will receive it at least once on the shortest path from i to k. Since only MPR nodes forward TC messages, there exist a shortest path from i to k made of MPR nodes.

MPR properties: shortest paths to any node

Imagine that only MPR nodes generate TC messages. Each TC message contains all the neighborhood of the MPR that generated it, so if a node k receives a TC from an MPR x somewhere in the network:

- k then knows the topology of the neighborhood of x
- k receives the TC since it has been forwarded from other MPR nodes on the shortest path from x to k
- By definition k receives TC messages also from the MPR in shortest path from x to k, with information on their neighborhood.
- As a consequence, k receives all the necessary information to compute the shortest path to any MPR.

MPR property 3, continued

Recall: every node must be a selector of at least one MPR.

- if every k in the network knows the shortest path to any MPR x in the network and also the neighborhood of x, then k knows the shortest path to any node in the network.
- Not only, there is no need for a MPR to include in TC messages its full neighborhood, it is sufficient to include only its selector set.

Step 3

Only MPR nodes will generate TC messages. Each TC message includes the selector set of the MPR that generated it

OLSR: summing up

OLSR Step 1

Every node must sponsor in HELLO messages its complete neighborhood specifying the kind of neighbor (SYM/ASYM). From now on we will simply call neighbors only the SYM neighbors.

OLSR Step 2

Each node must select a subset of N_i^2, as small as possible. In the HELLO messages, the state of each neighbor contains another bit that is MPR/NOT_MPR so each MPR knows its selector set. Only the MPR nodes participate in flooding messages rebroadcasting the packets from their selectors.

Step 3

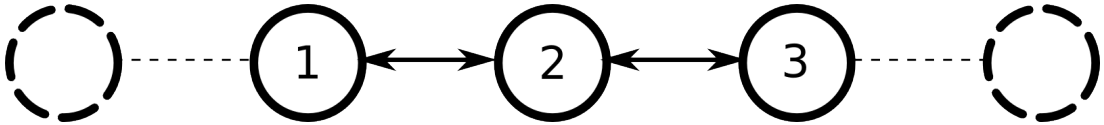
Only MPR nodes will generate TC messages. Each TC message includes the selector set of the MPR that generated it

OLSR: summing up

- If all the nodes apply step 1,2,3 and there is no loss, than all the nodes have enough information to compute the the shortest path to any node in the network.
- Since only the MPR generate TC messages and each TC includes only a fraction of the neighborhood of the MPR that generated it the overall control traffic is reduced to $(s * (n * \frac{M}{N})) * M^2$ from the initial $s * n * N^2$ with naive mode[1].
- It is very important that each node select a minimal MPR set, since M depends on that.

[1] 2 approximations I've done: the average number of selector set size for each MPR is $n * \frac{M}{N}$, and for each generated TC this will be forwarded by all the MPR. Both things are not true for all the topologies

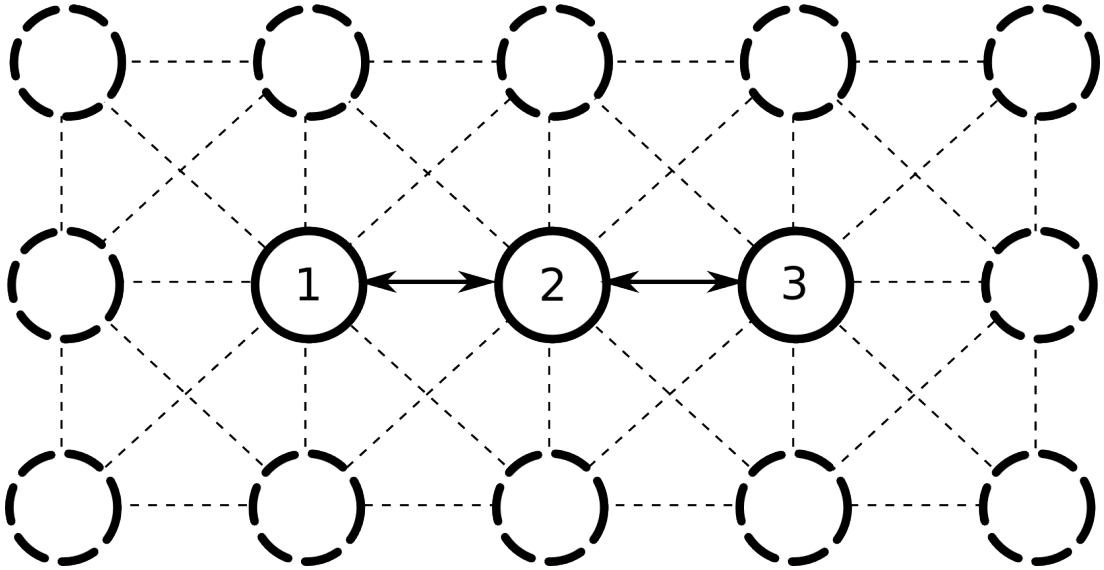
Examples[2]



- Only 3 nodes over 5 in the network generate TC messages
- Each TC message is forwarded exactly 2 times

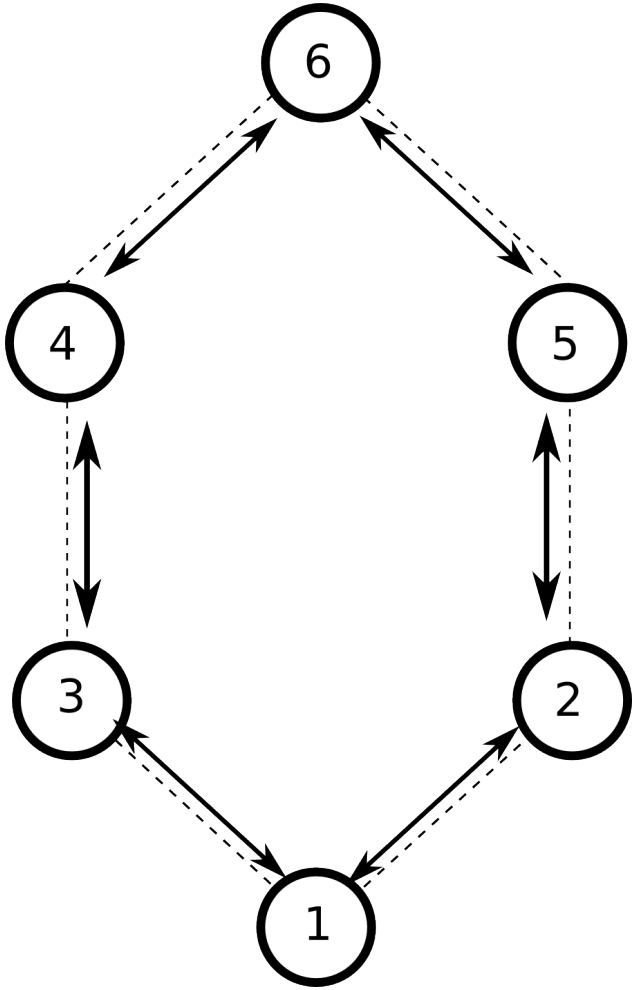
[2] dashed edges are wireless links, arrows go from a selector to the MPR (only the most important ones). Solid vertexes are MPRs.

Examples



- Only 3 nodes over 15 in the network generate TC messages
- Each TC message is forwarded exactly 2 times

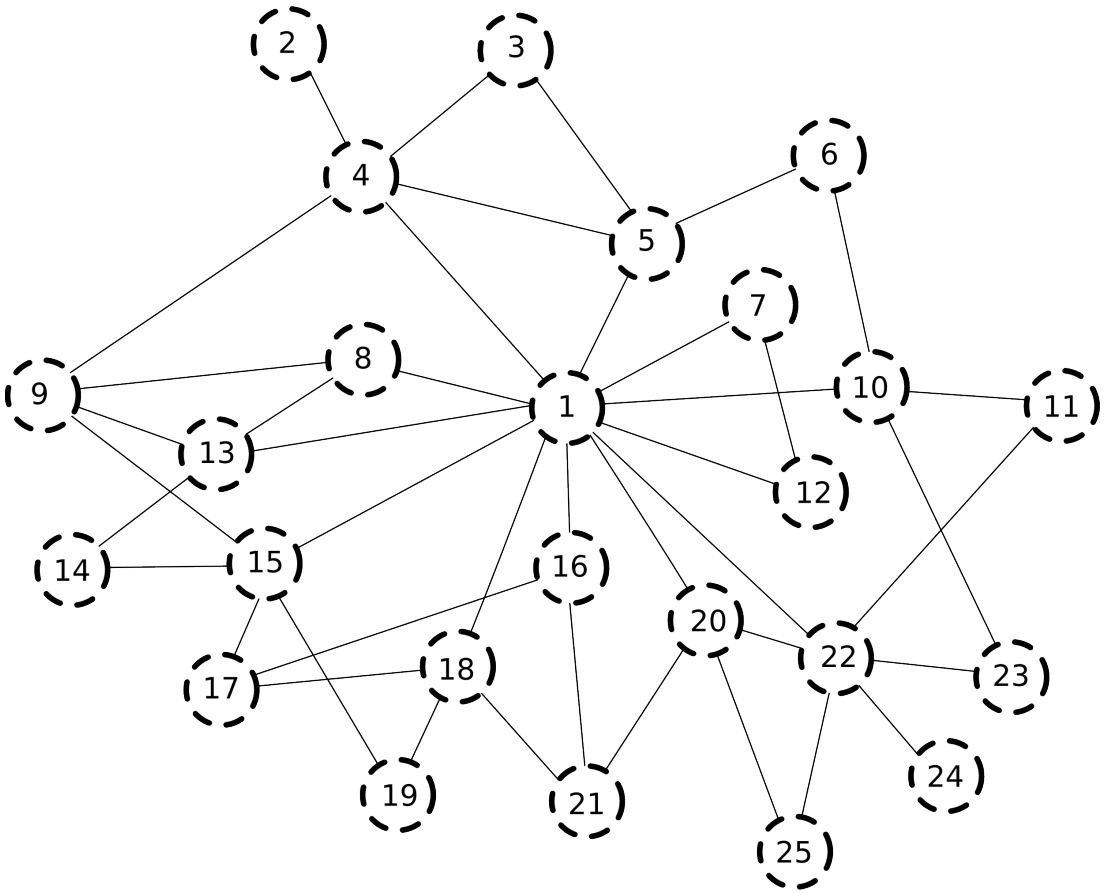
Examples



- All the nodes in the network generate TC messages
- Each TC message is forwarded exactly 5 times

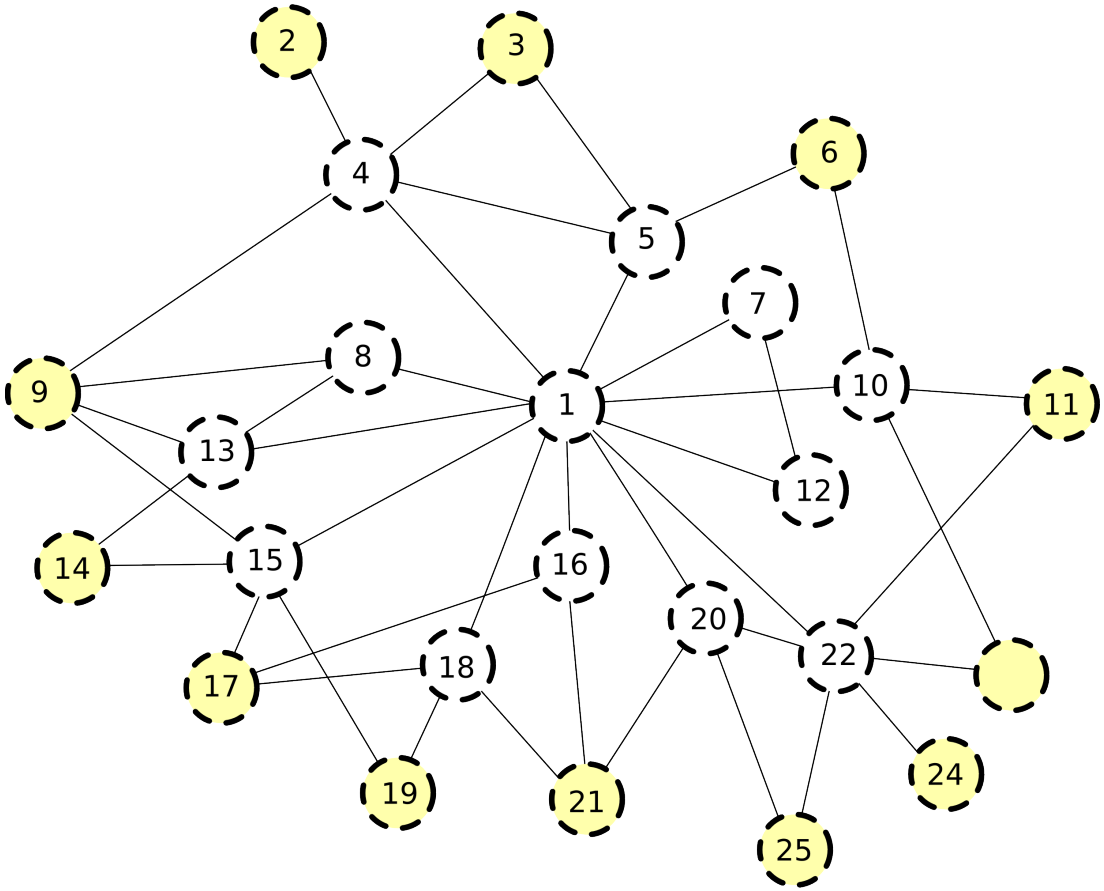
Exercise:

We have seen that the number of MPR in the network has a critical impact on control traffic, each node must try to find the smallest MPR set. Consider this network and try to compute the minimal MPR set for node 1



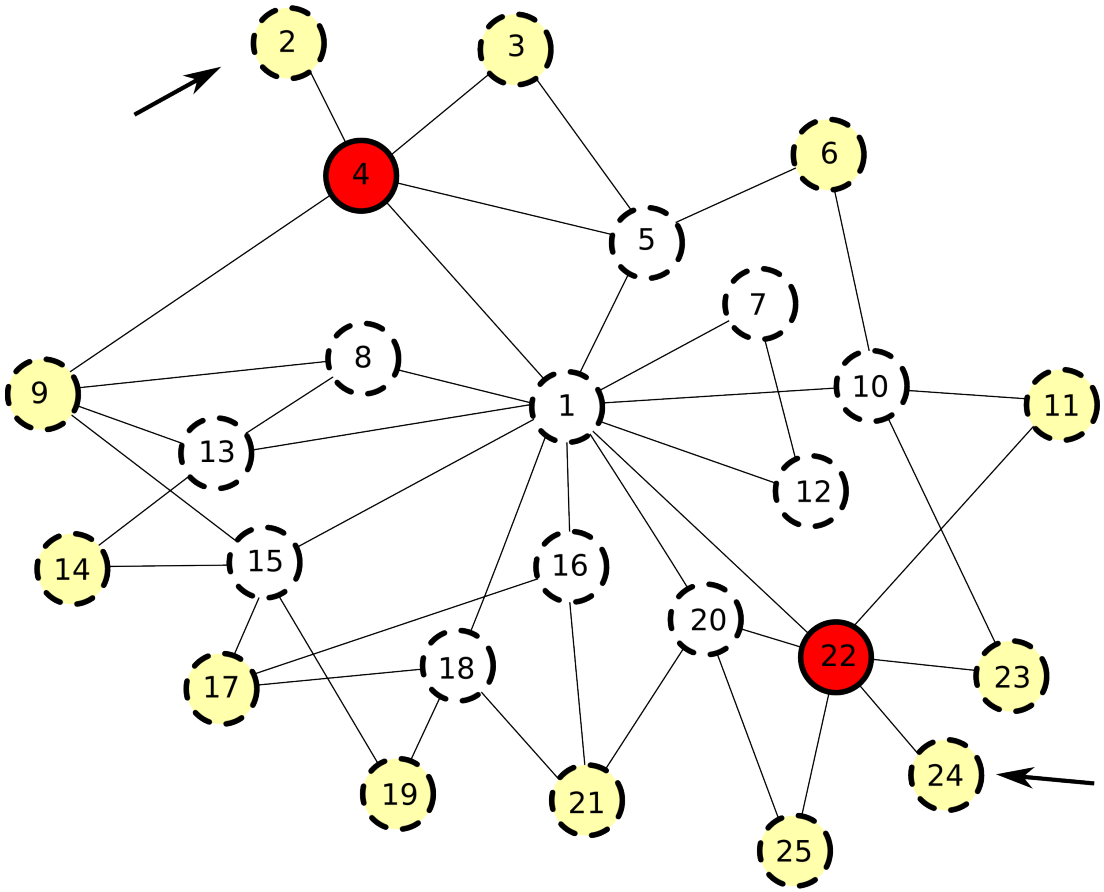
One possible solution: step 1

Identify the 2-hop neighbors



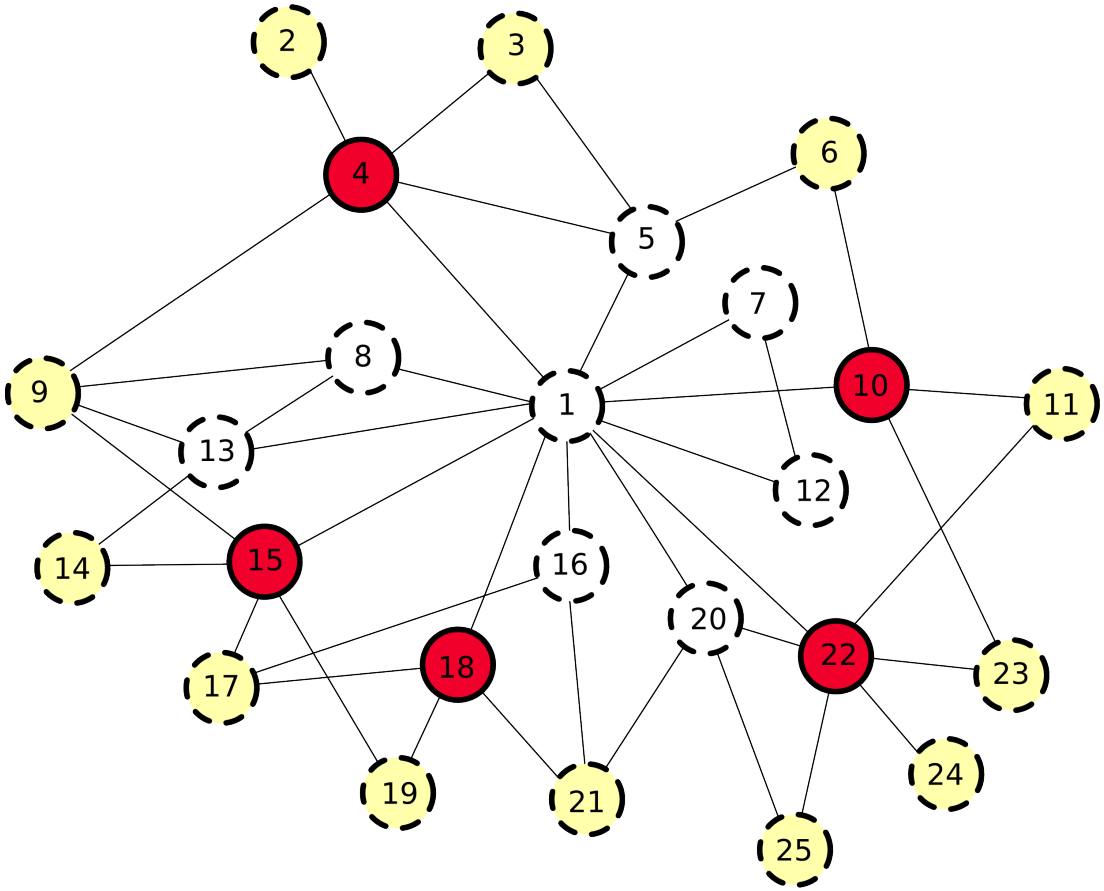
One possible solution: step 2

Identify the forced MPR nodes

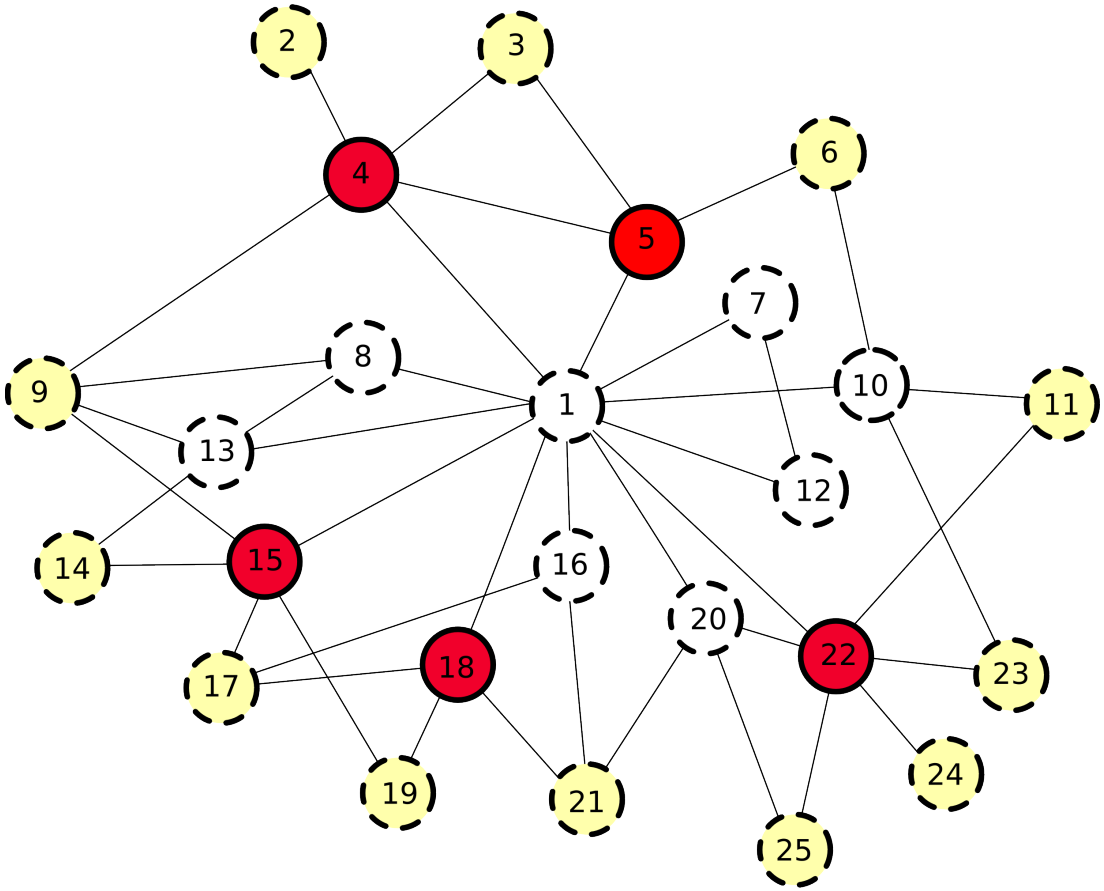


One possible solution: step 3

Try to select the smallest MPR set among the remaining 1-hop neighbors



Another possible solution: step 3



MPR Selection

Two issues emerge from this example

- In some topologies there are nodes that must be included in the MPR set
- There can be more than one minimal MPR set

MPR Selection: complexity

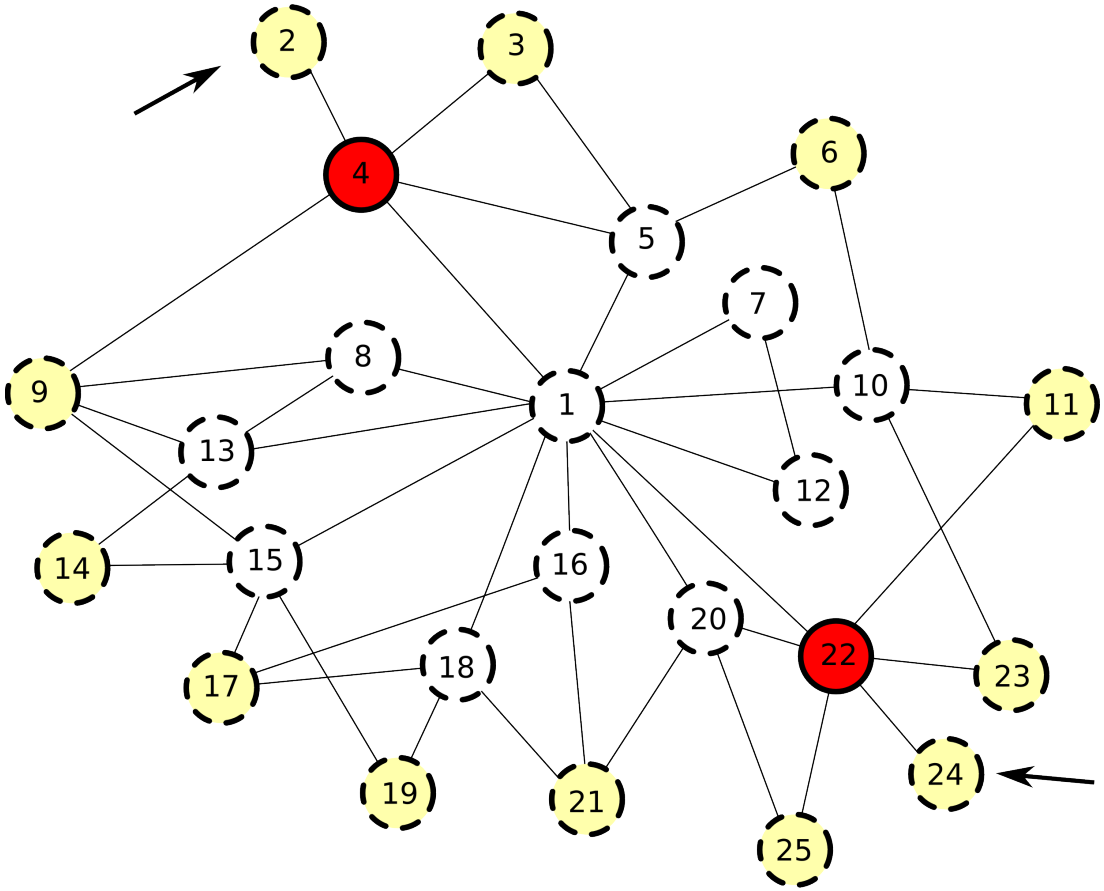
- It can be shown that finding the optimal MPR set is NP-complete.
  - It takes polynomial time to verify a given solution
  - No polynomial time algorithm is known to find a solution
- For large neighborhood it is inconvenient to try to find the optimal solution

MPR Selection: heuristic

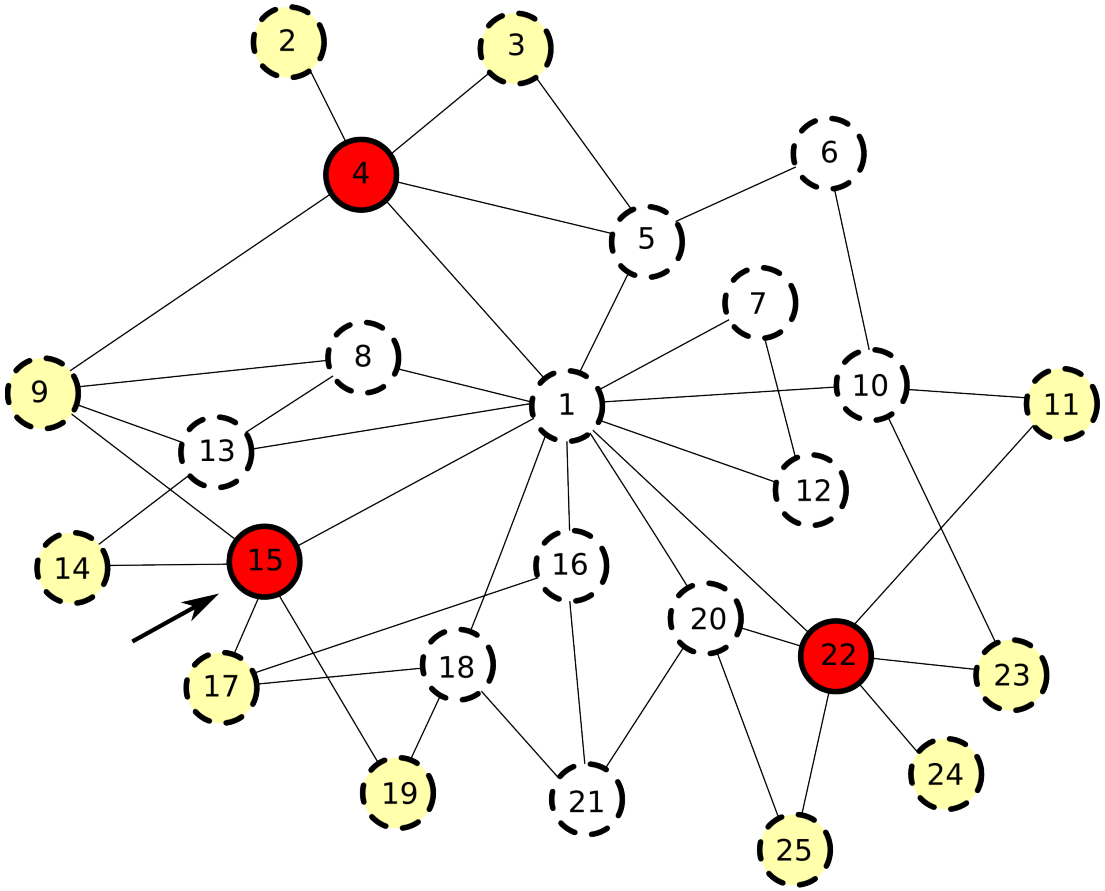
OLSR introduces the following heuristic for a node x:

1. Find all nodes j in n_x^2 that have only a neighbor k in n_x^1, insert k in m_x. Those nodes must be inserted to guarantee full 2-hop connectivity.
2. Repeat the following until any node in n_x^2 is reachable using nodes in m_x:
   1. Order every node u in $n_x \setminus m_x$ for their reachability, i.e. the number of nodes in n_x^2 that are covered by u and are not covered by any other node already in m_x. Insert in m_x the one with highest reachability.
   2. In case of a tie, insert the one with the highest willingness
   3. In case of a further tie compute the degree of u, that is $||n_u^1 \setminus n_x^1||$ and insert the one with higher degree. In practice, get the one with the highest number of 1-hop neighbors that are not shared with x

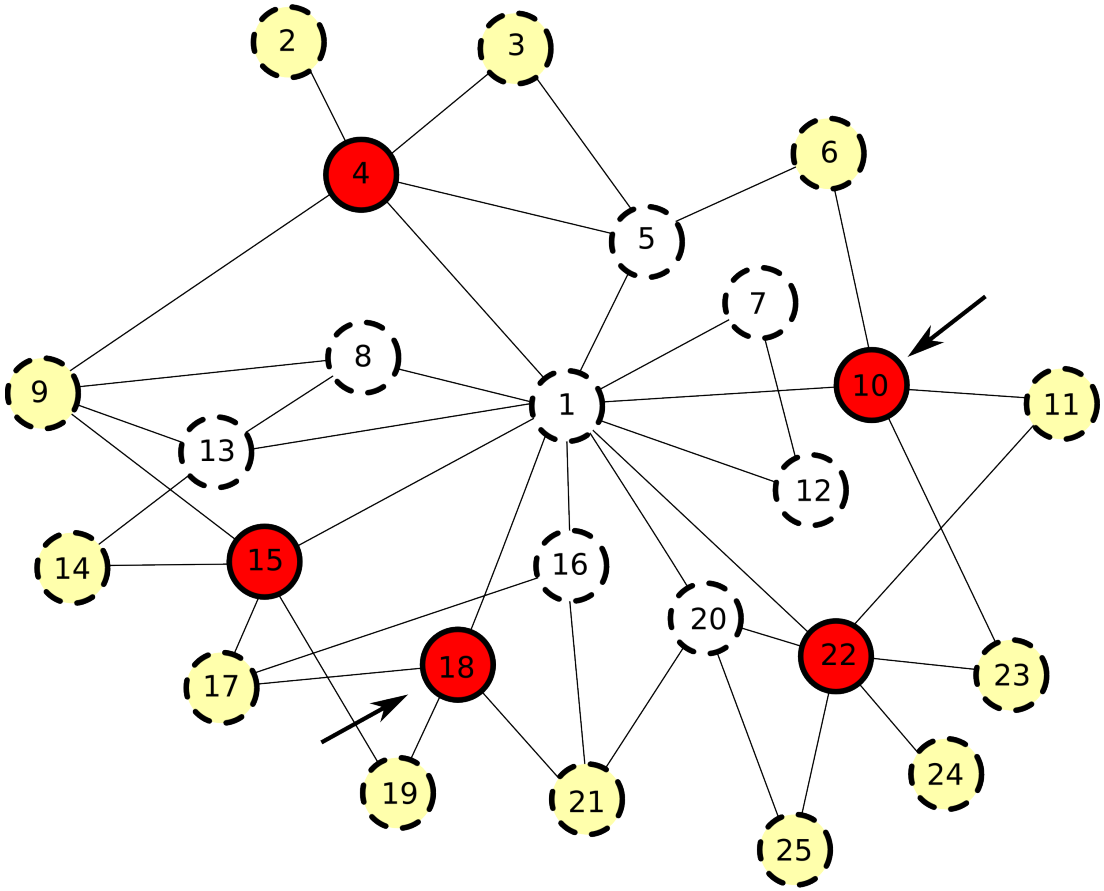
Step 1



Step 2



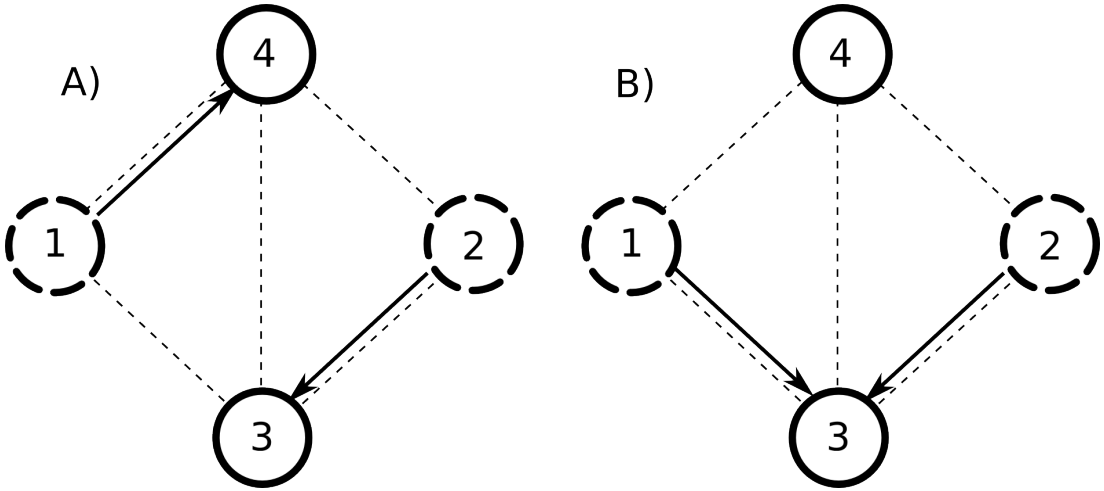
Step 3



MPR Selection: heuristic

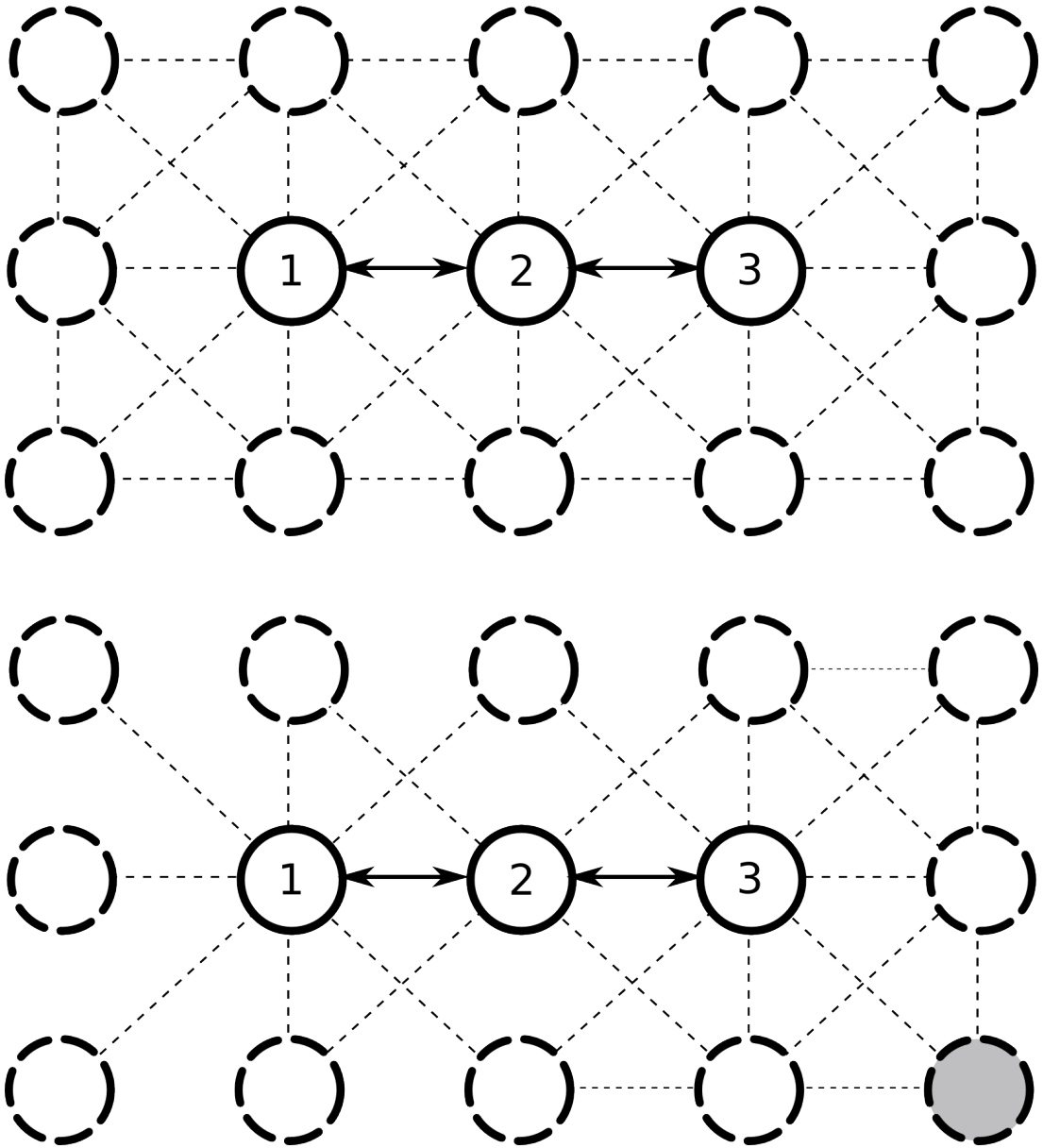
- It can be shown that this heuristic produces an MPR set of size S that compares to the size of the optimal set $S^{\star}$ with the following relation:
  - $S \leq log(\Delta)S^{\star}$
  - Where Δ is the maximum number of nodes in n_x^2 a node in n_x can cover.

Diamond network



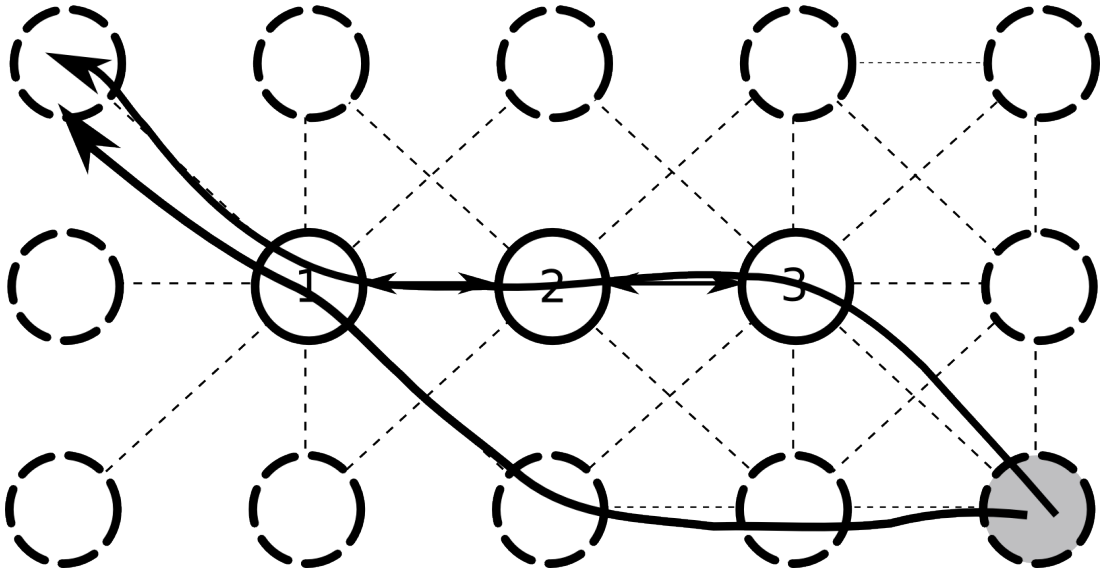
- Both these scenarios are perfectly possible but one generates twice the TC messages of the other
- This is the price to pay to have a completely distributed MPR selection algorithm

Network approximation



Equivalent routes

- Once you know an approximated topology, you can compute shortest-path routes
- Depending on the algorithm you chose you can have various paths



RT building

Note that:

- Note that the routes are decided hop-by-hop so any node can decide only the first hop
- In the following hops the route can be changed
- The closer you get to the destination, the more precise is the knowledge of the topology
- OLSR suggests an incremental algorithm for route formation

Fish-eye

- A node needs only the necessary information to decide what is the next hop on the shortest path to the destination.
- We have seen that nodes already have a limited view of the topology outside their 2-hop neighborhood
- This approach can be enhanced with fish-eye strategies

Fisheye

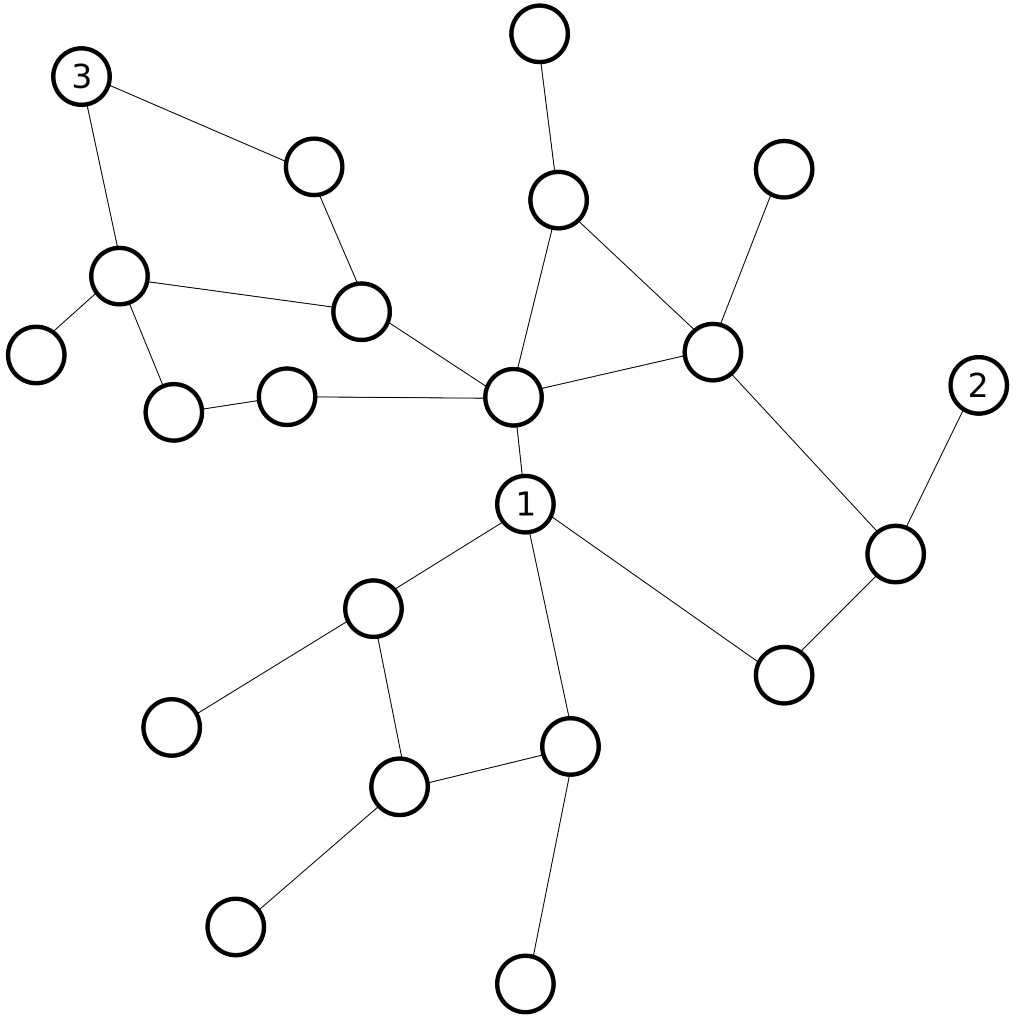
- The word fisheye refers to the view of the world a fish has under the water
- If you look up from below the water you see the whole 180 degrees horizon compressed in a cone with a smaller angle
- Basically, you have better vision on the close thinks and a fading compressed vision on objects far away. The same happens if you use a *fisheye* lens on a camera

Fisheye [3]

[3] http://en.wikipedia.org/wiki/File:Museumfisheye.jpg

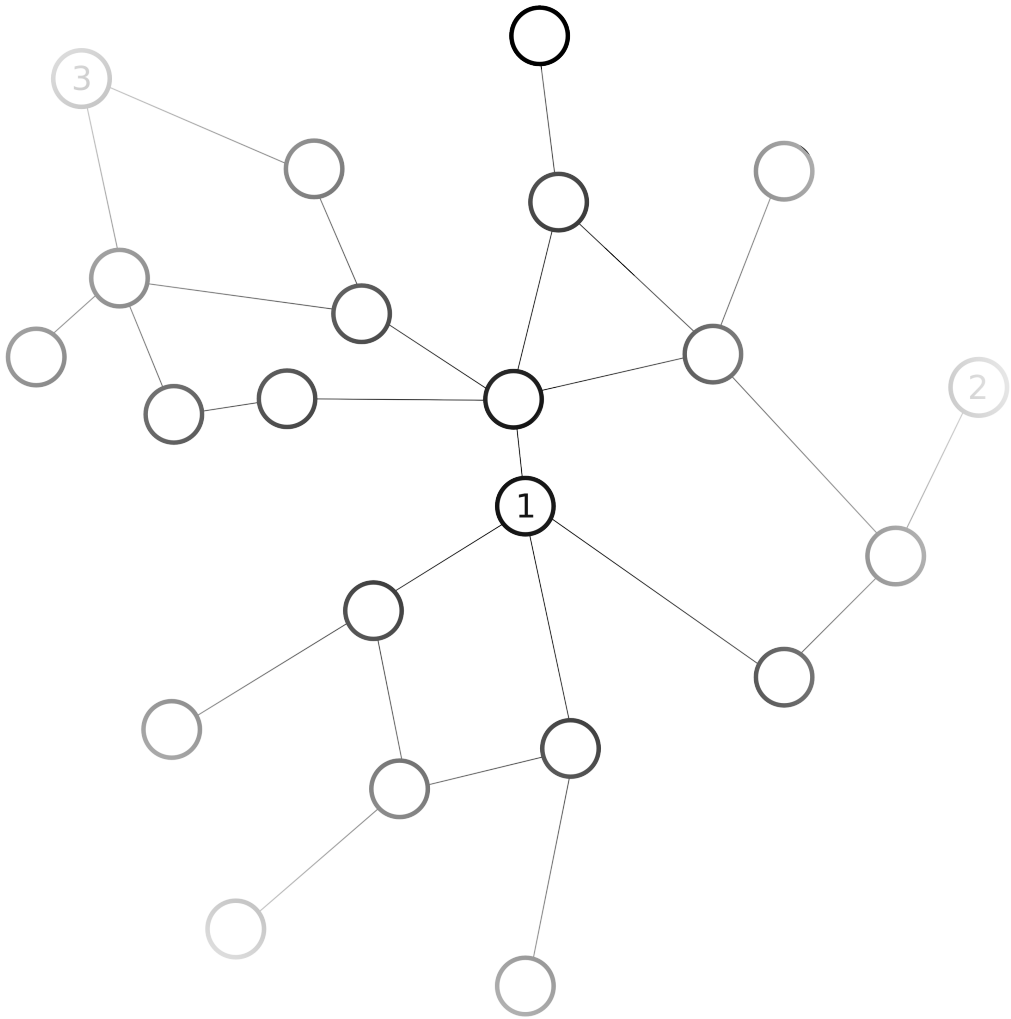
Fisheye OLSR

Fisheye transforms the network from this . . .



Fisheye OLSR

. . . to this



Fisheye

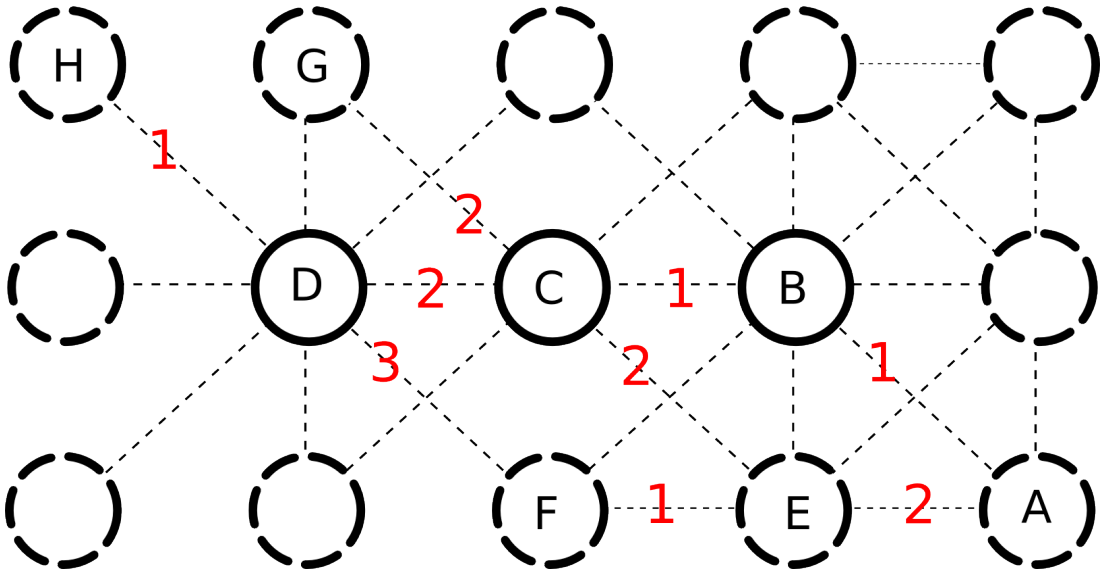
- to obtain this effect, the TTL of the TC messages is not always to the maximum.
- a sequence of TTL can be used to control the diffusion of TC messages, for instance {2,4,16}
- doing this, the first TC message will be sent only to nodes that are 2-hops away, the second to nodes that are 4-hops away, the third to all the nodes in the network
- periodically every node receives enough information to build all the routing table, but the information is more accurate (it is fresher) for closer nodes
- With Fisheye, the number of TC messages forwarded is not any more linear with the number of MPR nodes
- How to define the TTL sequence is an open issue

QOLSR

Can we add QoS to the routing tables?

- Each node makes an esteem of the link quality for its neighbors (a numeric value).
- Each node includes this value in its HELLO and possibly TC messages.
- The adjacency matrix now has weights on the edges and smarter algorithms can be used for routing

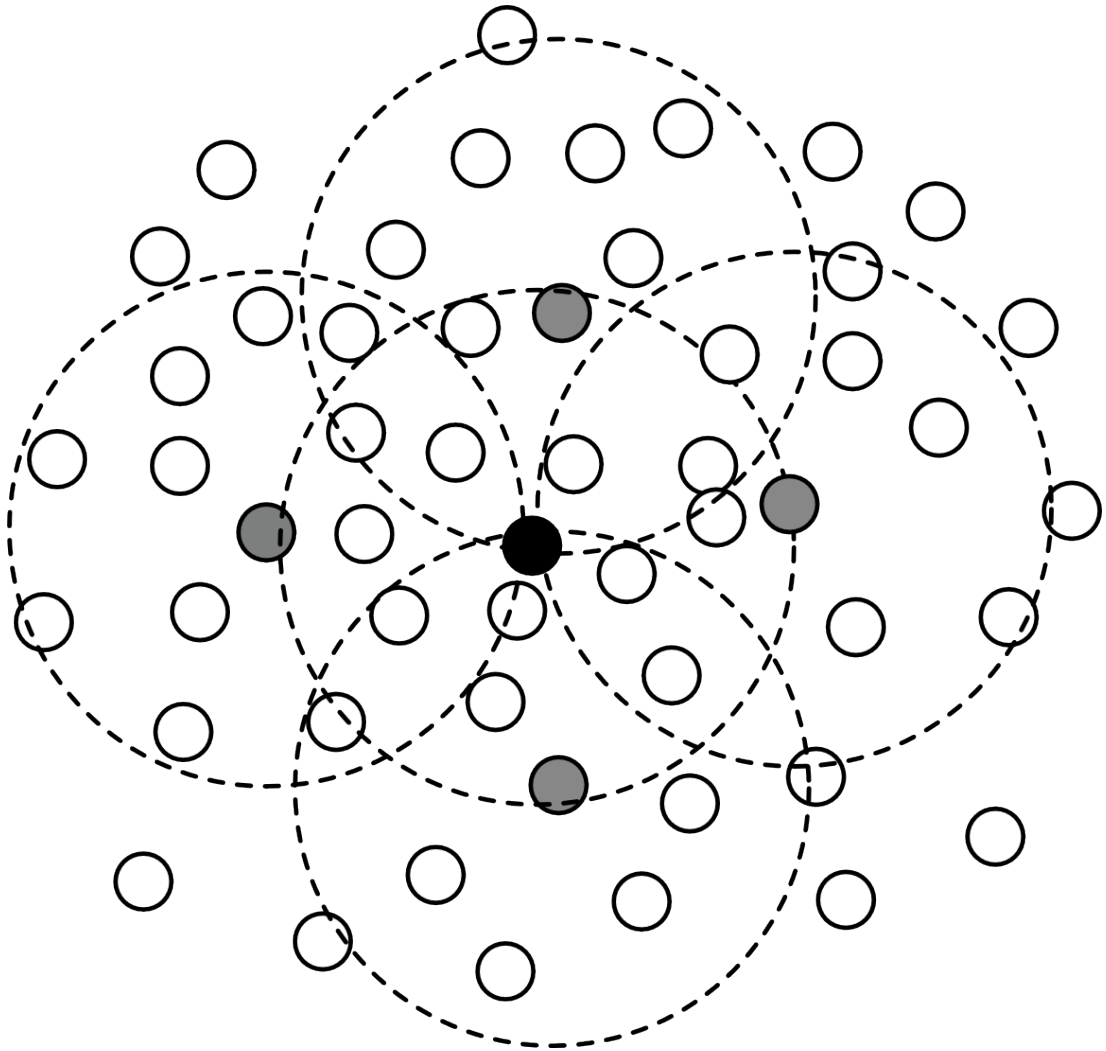
QOLSR



- From A to H, route A-B-C-D-H has weight 5 while route A-E-C-D-H has weight 7
- Node A will chose as next hop node E or node B depending on the quality required for the packet
- Note however that to have control all along the route, only the MPR nodes must be involved in packet routing
- How to define and measure the metrics, and how to avoid oscillation of routes are open issues.

QOLSR

- By design, MPR nodes are chosen on the border of the coverage area of the selector
- This doesn't make them a good candidate to have high quality links



QOLSR

Electing MPR with QOLSR:

- MPR nodes in QOLSR are used not only for broadcast messages, but to route unicast messages, so they must be chosen for the quality of their link with the selectors.
- A node may choose a set of MPR only for broadcast flooding trying to minimize the n^2 coverage, and a distinct set of MPR nodes trying to maximize the quality of the links.
- This will increase the control traffic but will also provide better routes.

OLSRv2

- OLSR is under revision to produce a new RFC including:
  - alignment with other MANET RFC
  - QoS features
- see draft-ietf-manet-olsrv2-14

Contact me!

- leonardo.maccari@unitn.it
- www.pervacy.eu

OLSR: Optimized Link State Routing

Leonardo Maccari

leonardo.maccari@unitn.it

28/5/2012