Ad hoc Network Routing

Wireless Systems, a.a 2015/2016

Un. of Rome “La Sapienza”

Chiara Petrioli[†]

[†] *Department of Computer Science – University of Rome “Sapienza” – Italy*

SAPIENZA
UNIVERSITÀ DI ROMA

What is an ad hoc network

- A wireless multi-hop infrastructure-less network whose devices act as source/ destination of messages & as relay for packets generated by a node s and addressed to a node z (iff they are on a s-z route)
- Pros: No need for infrastructure → low cost, enables communication where it is usually not needed or it is not viable
- Must be: Self-organizing, self-configuring, self-maintaining

Application scenarios

- Disaster recovery applications
- Military networks
- Personal Area Networks
- Home Networking
- Wireless Sensor Networks (WSNs)
- Inter-vehicular communication
- Mesh Networks (extension of WiFi standard)

Features of ad hoc networks

- Highly dynamic networks → device mobility, energy saving sleep/awake modes
- Need for low energy, low overhead, simple protocols
- Traffic:
  - All-pairs in general ad hoc networks
  - Can be low or high (multimedia) traffic
- Scale: Application dependent
  - 10-100 nodes in traditional ad hoc networks

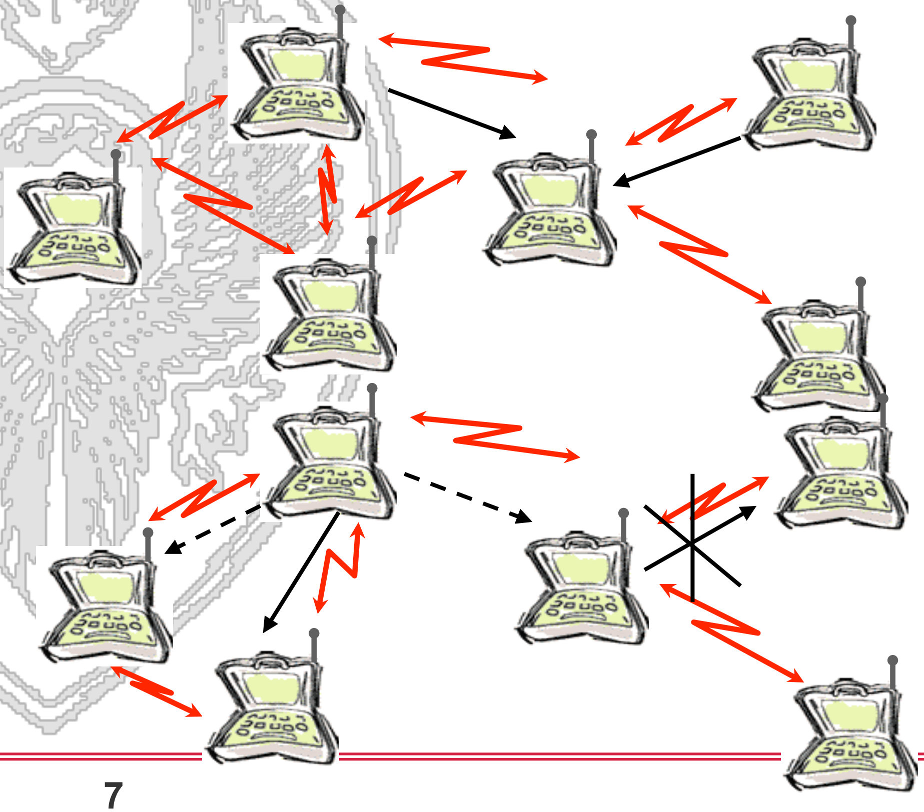
SAPIENZA
UNIVERSITÀ DI ROMA

Features of highly resource constrained ad hoc networks (WSNs)

- Highly dynamic networks → due to device mobility (only in some specific applications), to the fact the active node set changes in time for sake of energy saving (always to be considered)
- Need to design low energy protocols → very critical, energy consumption a real bottleneck, memory also a bottleneck
- Traffic from sensors to sink(s)
  - Even if opportunistic communication is also increasingly considered
- Scalability is a major issue (could go up to 1000 or 10000 nodes)
- Code must be simple (small storage capability, very simple, inexpensive, resource constrained devices)
- First solutions we will see for traditional ad hoc networks do not scale to high numbers and are not energy-saving, and are typically too complex for resource constrained embedded systems devices

Medium Access Control in Ad Hoc Networks

- CSMA-like approach, e.g. CSMA/CA
  - Why not TDMA like?
- Why not CSMA/CD? Nodes cannot received and transmit simultaneously
  - Hidden terminal
  - Exposed terminal



SAPIENZA
UNIVERSITÀ DI ROMA

802.11 Distributed Coordination Function

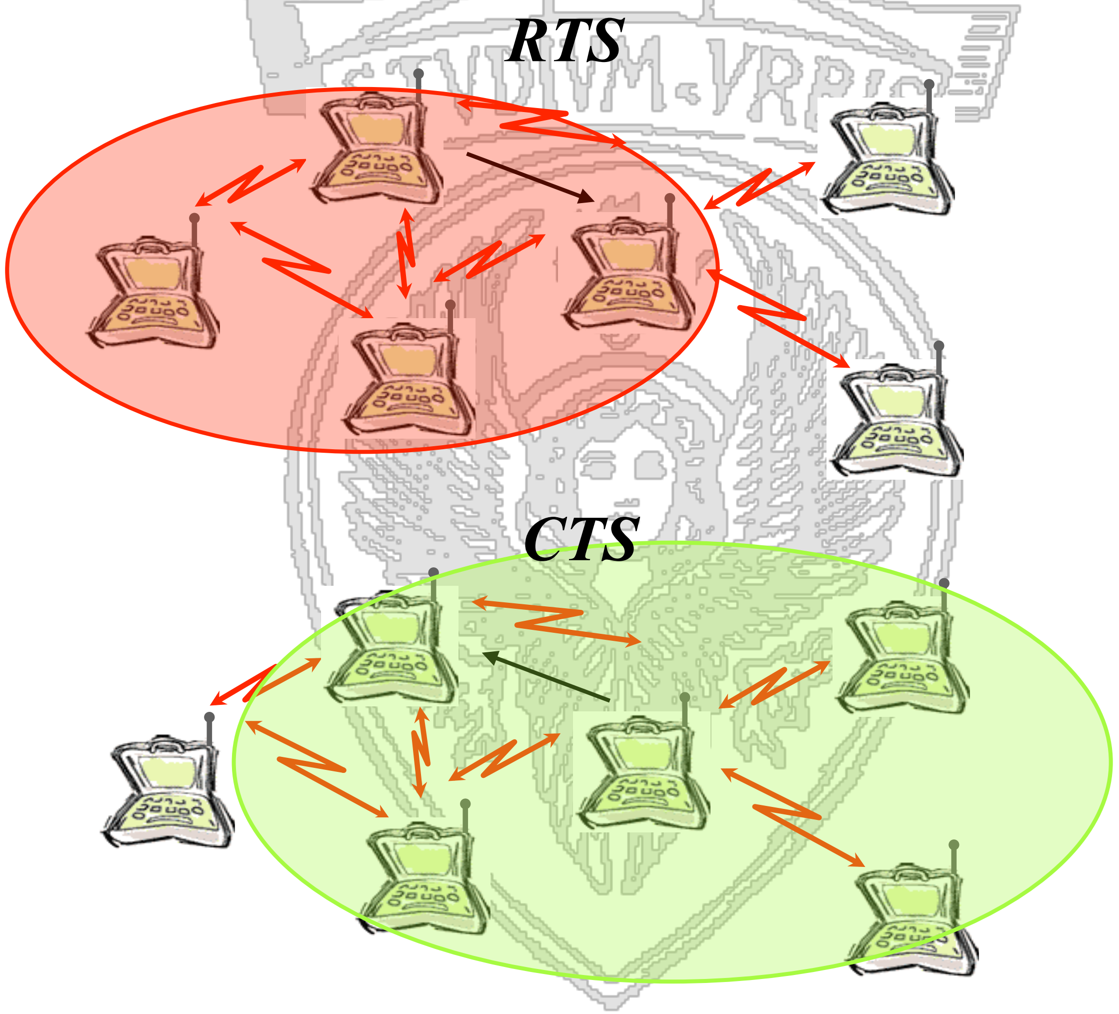
- Based on CSMA/CA
- Before transmitting a frame the sender node x performs carrier sensing
- If the channel is free for a time called Distributed InterFrame Space (DIFS) node x transmits the packet
- Otherwise (being the channel in use) node x waits for the end of current transmission + a random backoff time.
  - The backoff timer count down is frozen whenever the node senses the channel busy (only when the channel is free for a DIFS the counter value is decreased) WHY?
  - When the backoff timer value is zero node x transmits the packet.
  - The backoff timer value is randomly picked within a window interval of CW slots. At the first transmission attempt CW is set to the minimum value (16 slots). An exponential backoff is used. At each retranmission attemp CW is doubled till a maximum value equal to 1024 slots.
- How can node x decide whether the transmitted packet has been successfully received? By means of an explicit ACK the receiving nodes transmits (if the packet is correctly received after a Short InterFrame Space (SIFS) time, SIFS<DIFS)

802.11 DCF
Virtual carrier sensing

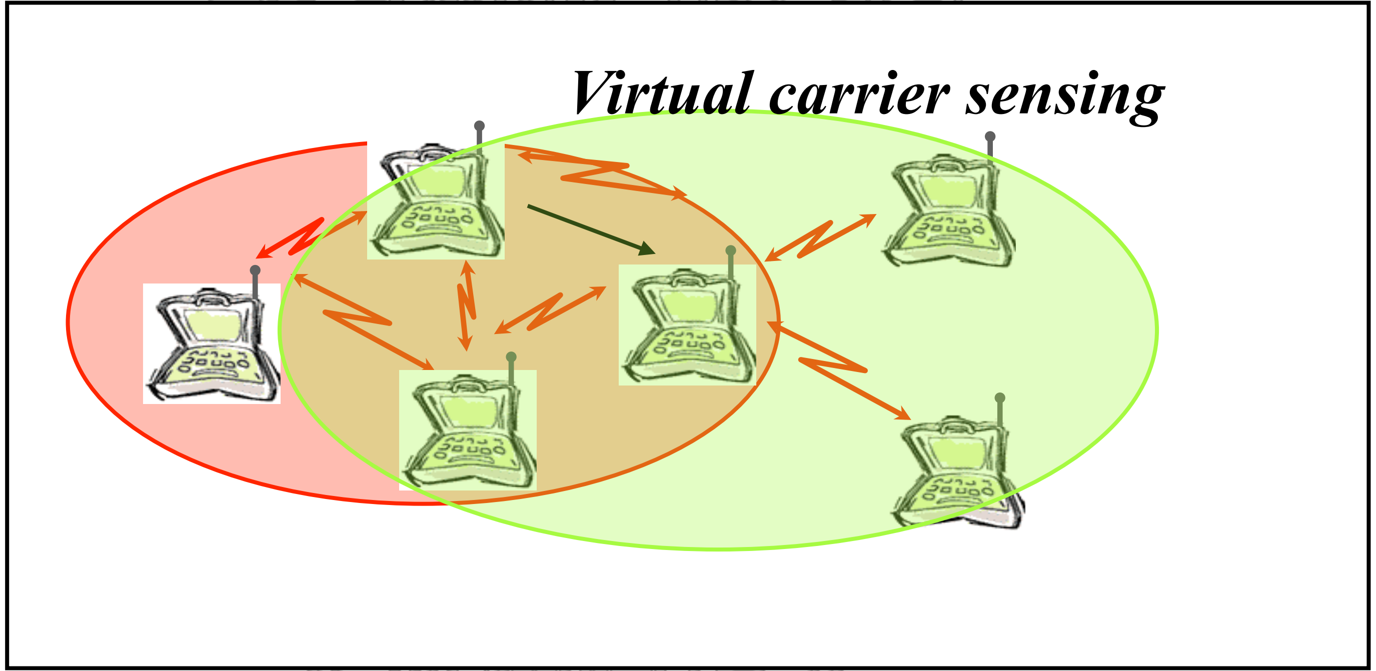
- To mitigate performance impairments due to the hidden terminal phenomenon DCF uses virtual carrier sensing
- Before transmitting a frame the sender node performs carrier sensing, waits for a time called Distributed InterFrame Space (DIFS) and then transmits a short control packet named Request To Send (RTS), informing its neighbors that it is going to transmit a packet towards the destination
- RTS includes a NAV (Network Allocation Vector) field whose value expresses the time from the RTS reception to the end of the handshake (ACK reception)
  - By receiving the RTS and looking at the NAV value all nodes can estimate when not to transmit in order not to interfere with the on-going transmission
- If the receiver correctly receives an RTS it waits for a SIFS and then transmits a Clear To Send (CTS) message
  - All destination neighbors will know that the channel is busy, and for how long
  - CTS also includes a NAV field
- Upon receiving a CTS, the sender transmits the DATA packet (after SIFS)
- Upon receiving the DATA packet, the destination waits for a SIFS and then sends the ACK
- If the handshake is not correctly completed the node performs a retransmission attempt after an exponential backoff

SAPIENZA
UNIVERSITÀ DI ROMA

Virtual Carrier Sensing



Virtual Carrier Sensing



Other approaches?

- Can TDMA be an option?
- Or an hybrid solution?

- Several proposals in the literature
  - Hybrid or TDMA approaches used in sensor networks
  - Demonstrated promising for MANET based on simulations

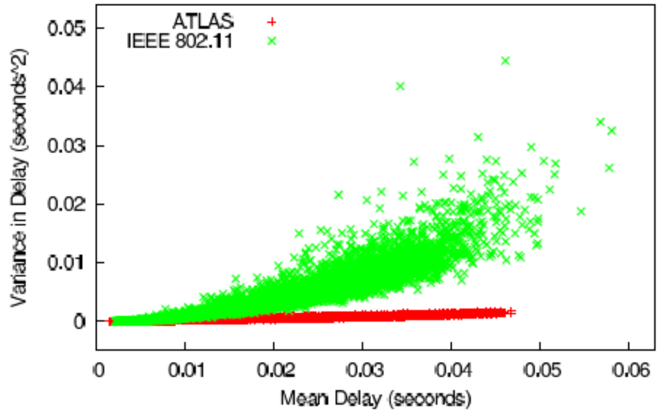
Other approaches?

- Can TDMA be an option?
  - Synchronization in higly dynamic environments
  - Static allocation of resources is a limit
  - Efficiency depends on traffic (control overhead balanced by increased efficiency in high traffic scenarios);
- Or an hybrid solution?

- Several proposals in the literature
  - Hybrid or TDMA approaches used in sensor networks
  - Demonstrated promising for MANET based on simulations, in case of adaptive schemes, also ensuring some level of fairness

Other approaches?

- Can TDMA be an option?
- Or an hybrid solution?
- Several proposals in the literature
  - Hybrid or TDMA approaches used in sensor networks
  - Demonstrated promising for MANET based on simulations
- Jonathan Lutz, Charles J. Colbourn, Violet R. Syrotiuk: ATLAS: Adaptive Topology- andLoad-Aware Scheduling. IEEE Trans. Mob. Comput. 13(10): 2255-2268 (2014)



Routing-Background

Routing-Traditional approach (from Reti 1)

- Intra-AS routing in the Internet
  - Link State Approaches

    (info on the topology graph gathered at nodes which run shortest path algorithms-Dijkstra- to decide the routes to the different destinations –e.g. OSPF routing protocol)
  - Distance Vector approaches (e.g. RIP)

Bellman-Ford

Given a graph G=(N,A) and a node *s* find the shortest path from *s* to every node in N.

A shortest walk from *s* to i subject to the constraint that the walk contains at most h arcs and goes through node *s* only once, is denoted **shortest(<=h) walk and its length is D^h_i.**

<u>Bellman-Ford rule:</u>

Initiatilization $D^h_s=0$, for all h; $c_{i,k}$ = infinity if (i,k) NOT in A; $c_{k,k}$ =0; **D^0_i=infinity for all i!=s.**

<u>Iteration:</u>

$$D^{h+1}_i=\min_k [c_{i,k} + D^h_k]$$

Assumption: non negative cycles **(this is the case in a network!!)**

The Bellman-Ford algorithm first finds the one-arc shortest walk lengths, then the two-arc shortest walk length, then the three-arc…etc. →distributed version used for routing

Bellman-Ford

$$D^{h+1}_i = \min_k [c_{i,k} + D^h_k]$$

Can be computed locally.
What do I need?

For each neighbor *k*, I need to know
-the cost of the link to it (known info)
-The cost of the best route from the neighbor *k* to the destination (←this is an info that each of my neighbor has to send to me via messages)

In the real world: I need to know the best routes among each pair of nodes → we apply distributed Bellman Ford to get the best route for each of the possible destinations

Distance Vector Routing Algorithm -Distributed Bellman Ford

iterative:

- continues until no nodes exchange info.
- *self-terminating*: no "signal" to stop

asynchronous:

- nodes need *not* exchange info/iterate in lock step!

Distributed, based on local info:

- each node communicates *only* with directly-attached neighbors

Distance Table data structure

each node has its own

- row for each possible destination
- column for each directly-attached neighbor to node
- example: in node X, for dest. Y via neighbor Z:

Cost associated to the (X,Z) link

$D^X(Y,Z)$ = distance *from* X *to* Y, *via* Z as next hop

$= c(X,Z) + \min_w \{D^Z(Y,w)\}$

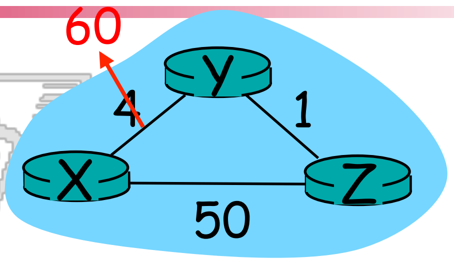
Info maintained at Z. Min must be communicated

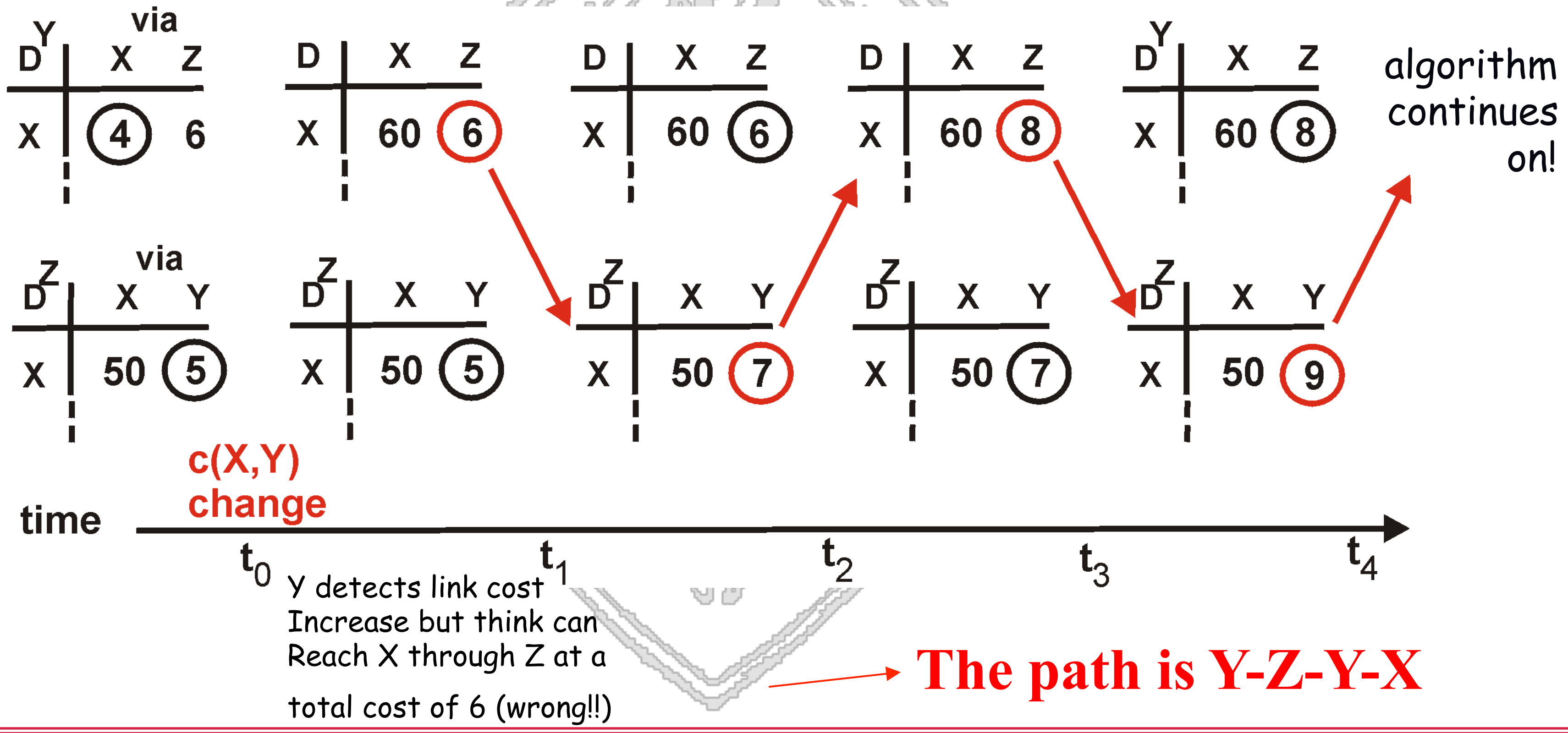
SAPIENZA
UNIVERSITÀ DI ROMA

Distance Vector: link cost changes

Link cost changes:

- good news travels fast
- ***bad news travels slow*** - "count to infinity" problem!





D^Y	X	Z
X	4	6

D^Z	X	Y
X	50	5

Time	D^Y: via X	D^Y: via Z	D^Z: via X	D^Z: via Y
t_0	4	6	50	5
t_1	60	6	50	5
t_2	60	6	50	7
t_3	60	8	50	7
t_4	60	8	50	9

Y detects link cost Increase but think can Reach X through Z at a total cost of 6 (wrong!!)

The path is Y-Z-Y-X

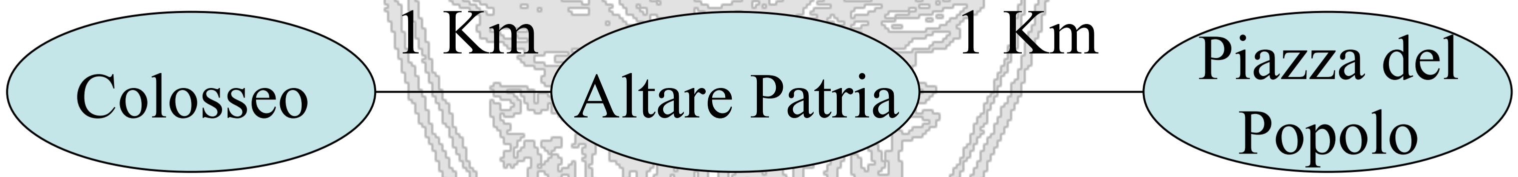
Count-to-infinity – an everyday life example

Which is the problem here?

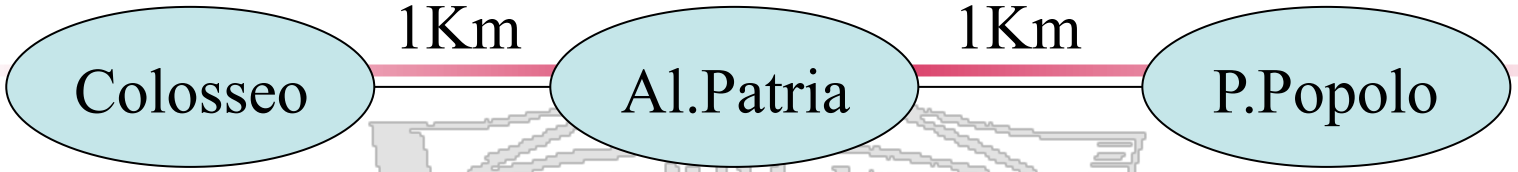
the info exchanged by the protocol!! 'the best route to X I have has the following cost…' (no additional info on the route)

A Roman example…

-assumption: there is only one route going from Colosseo to Altare della Patria: Via dei Fori Imperiali. Let us now consider a network, whose nodes are Colosseo., Altare della Patria, Piazza del Popolo



Count-to-infinity – everyday life example (2/2)



The Colosseo. and Alt. Patria nodes exchange the following info
• Colosseo says ‘the shortest route from me to P. Popolo is 2 Km’
• Alt. Patria says ‘the shortest path from me to P. Popolo is 1Km’
Based on this exchange from Colosseo you go to Al. Patria, and from there to
Piazza del Popolo OK **Now due to the big dig they close Via del Corso (Al. Patria—P.Popolo)**
• Al. Patria thinks ‘I have to find another route from me to P.Popolo. Look there is a route from Colosseo to P.Popolo that takes 2Km, I can be at Colosseo in 1Km → I have found a 3Km route from me to P.Popolo!!’ Communicates the new cost to Colosseo that updates ‘OK I can go to P.Popolo via Al. Patria in 4Km’
VERY WRONG!! Why is it so? I didn’t know that the route from Colosseo to P.Popolo was going through Via del Corso from Al.Patria to P.Popolo (which is closed)!!

SAPIENZA
UNIVERSITÀ DI ROMA

Count to infinity-solutions

- Bounded network diameter (RIP)
  - It is possible to use a TTL and discard all packets that have traversed more than x hops.
  - If network diameter is limited (15 in RIP networks) convergence in case of count to infinity is fast.
- Split horizon with poison reverse
  - Limits transmitted information. If A uses information received by B to select the route towards D (in other words if A's next hop relay to reach D is B), A will not communicate valid route lengths to B, or it will communicate infinity
    - ✓ Broadcast cannot be used to send updates
    - ✓ Does not solve all loop situations
- Trigger Updates (to fasten convergence)
  - Instead of sending periodic updates, updates can be transmitted immediately in case route lengths change

Link State Approach

- Each node periodically sends information on its neighbors (and the associated cost on the links to them) to all other nodes in the network
  - Via flooding or a variant of flooding
- Updates can be sent also in case changes are detected
- As each node has a complete view of the network topology it can locally compute the best route towards the destination
  - Running Dijkstra
- The node then populates its rouing table accordingly

Ad Hoc Networks Routing

Routing in ad hoc networks- Goals

- Multi-hop path routing capability
- Dynamic topology maintenance
- “No loops”
- Minimal control overhead
- Low processing overhead
- Self-starting

Primary Approaches

- Proactive
  - Based on traditional distance-vector and link-state protocols
  - Each node maintains route to each other network node
  - Periodic and/or event triggered routing update exchange
  - Higher overhead in most scenarios
  - Longer route convergence time
  - Examples: DSDV, OLSR

Highly Dynamic Destination-Sequenced Distance-Vector (DSDV) Routing

- Proactive, distance vector approach (uses distributed asynchronous Bellman Ford). Updates on route costs transmitted periodically or when significant new information is available.
- Difference wrt Bellman Ford: in ad hoc networks there are frequent changes in the topology, solutions must try to avoid loops (approaches such as Poison reverse are not effective in broadcast channels, we seek solutions which are simple and fully distributed)
- Metrics: fresh routes better than stale routes, number of hops used to select among the fresh routes
- How to identify fresh routes? By means of sequence numbers identifying the freshness of the communicated information. When changes occur, the sequence number increase.

Highly Dynamic Destination-Sequenced Distance-Vector (DSDV) Routing

- Periodically destination nodes transmit updates with a new sequence number (and such updates are propagated by the other nodes).
- Updates periodically sent by nodes contain information on the costs to achieve the different destinations and the freshness of the route
- Data broadcast include multiple entries each with:
  - Destination address
  - Number of hops required to reach the destination
  - Sequence number of the information received regarding that destination as originally stamped by the destination
- In the header the data broadcast also include:
  - Address (HW address/Net address) of the sender of the message
  - Sequence number created by the transmitter
- Two types of updates (full dump or incremental-only changes- to decrease bandwidth consumption.

Highly Dynamic Destination-Sequenced Distance-Vector (DSDV) Routing

- How can the costs be modified? Cost=number of hops, target: using fresh routes as short as possible → a link cost changes from 1 to inf and from inf to 1
- How do we detect that a link is 'broken'? At layer 2 (no hello messages received for some time, or attempts to retransmit a frame exceeds the MAC protocol threshold) or at layer 3 (do not receive periodic updates by a neighbor)
- Link cost increase (1→ inf):
  - The nodes incident to that link (A,B) discover it (see above)
  - Routes going through that link get assigned an inf cost in nodes A and B routing tables
  - A new sequence number is generated by the mobile node. Mobile nodes different from the destination use odd SN, the destination even SN.
  - Updates with routes with infinite cost are **immediately transmitted** by nodes
- Link cost decrease (inf→1):
  - Immediately transmits updates

Highly Dynamic Destination-Sequenced Distance-Vector (DSDV) Routing

- When a node receives updates it sees if costs to reach the different destinations can be improved:
  - routes with more recent sequence numbers to a given destination are used
  - if more routes available with the same SN the shortest is used
- Newly recorded routes are scheduled for immediate advertisement (inf→ finite value)
- Routes with improved metric are scheduled for advertisement at a time which depends on the estimated average settling time for routes to that particular destination (based on previous history) → delayed advertisements to decrease the overall overhead
- As soon as a route cost changes the node may delay informing its neighbors but immediately starts using the new information for its forwarding

Highly Dynamic Destination-Sequenced Distance-Vector (DSDV) Routing-Correctness

- Assuming routing tables are stable and a change occurs
  - let G(x) denotes the routes graph from the sources to x BEFORE the change (assume no loop)
  - change occurs at *i* when 1) the link from *i* to its parent p(*i*) in G(x) breaks → *i* sets to inf that route (no loop can occur) 2) node *i* receives from one of its neighbors k a route to x with sequence number SN^x_k and metric m which is selected to replace the current metric *i* is using to select the route to x (this occurs only if SN^x_k greater than the previous SN I had stored Sn^x_i or if the two SN are equal but the new route has a lower hop cost → in the first case if selecting k leads to a loop then $SN^x_k <= Sn^x_i$ which is a contradiction, in the second case the claim comes from the observation that in presence of static or decreasing link weights distance- vector algorithms always maintain loop-free paths).

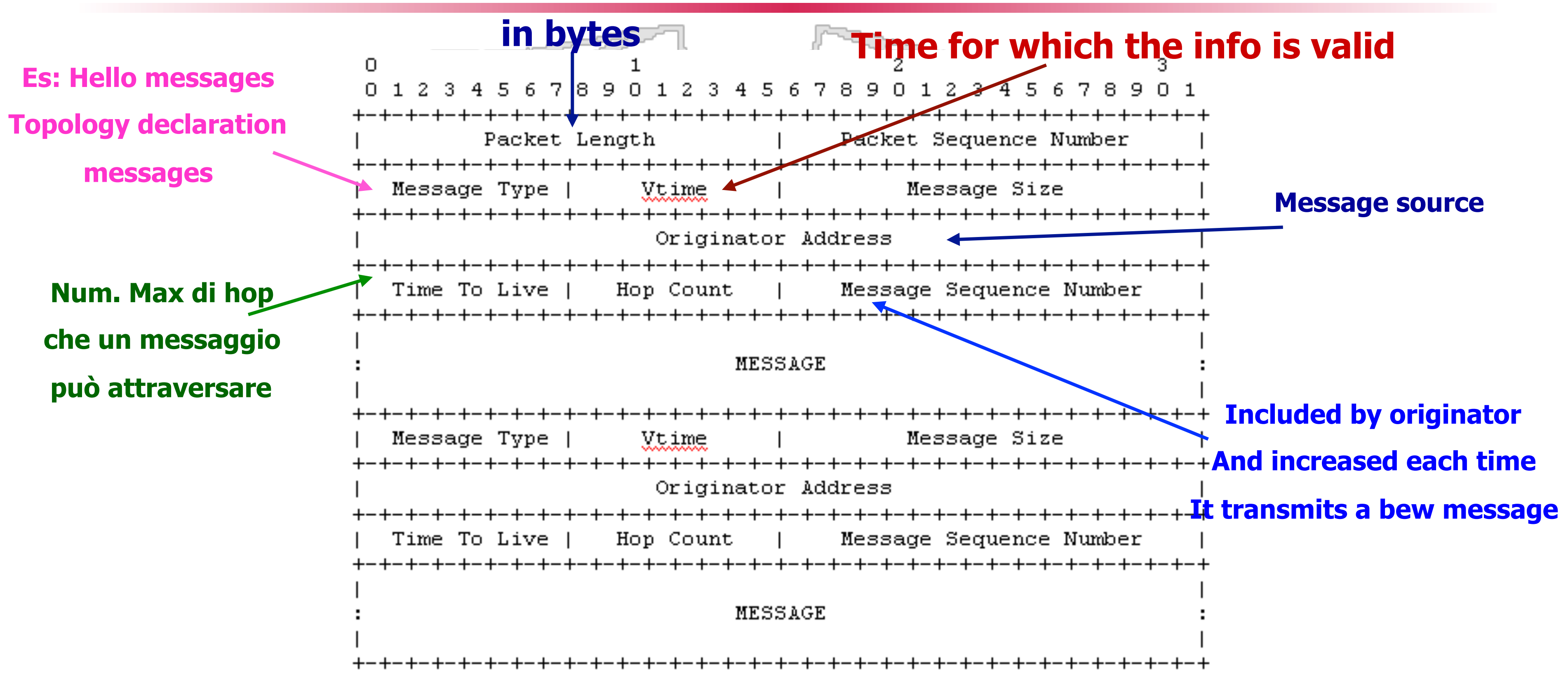
OLSR

- Optimized Link State Routing (OLSR) is a link state protocol for MANETs
  - suited for large and dense ad hoc networks
- The key concept is to decrease the overhead of flooding by means of identifying a subset of nodes (multipoint relays) in charge of forwarding the information during the flooding process
  - **Multipoint relay Y**: a node selected by at least one of its 1-hop neighbors (say node X) to relay all valid broadcast information it receives from X (the broadcast information is valid if it is not expired and not duplicate).
    - ✓ MPR(X)=set of multipoint relays of node X
    - ✓ neighbors of node X which are not in MPR(X) receive and store the broadcast messages transmitted by X but DO NOT retransmit them
  - A node X which has selected a neighbor Y as multi-point relay is called a **multipoint relay selector** of node Y
- Requires only partial link state to be flooded
  - links from MPR to their selectors must be declared
    - ✓ enough to ensure routes to each destination can be found
  - additional link state information MAYBE advertised

OLSR- other features

- The protocol is fully distributed
- Proactive approach: routes are always available when needed
- Other features:
  - Time between updates can be tuned to increase reactivity to topological changes
  - does not require reliable transmission
    - ✓ some losses are tolerated ← needed info are periodically transmitted
      - OLSR control packets are embedded in UDP datagrams
    - ✓ sequenced delivery of the messages is not needed ← proper recostruction of the sequence is possible due to use of sequence numbers
  - support to other MANET related issues
    - ✓ Sleep mode operation
    - ✓ Multicasting

Packet format



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Packet Length         |    Packet Sequence Number     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Message Type |     Vtime     |         Message Size          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Originator Address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time To Live |   Hop Count   |    Message Sequence Number    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                            MESSAGE                            :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Message Type |     Vtime     |         Message Size          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Originator Address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time To Live |   Hop Count   |    Message Sequence Number    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                            MESSAGE                            :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Each node maintains triples <originator address, sequence number, if the message has already been transmitted> for messages recently received. This allows to discard duplicates.

Packets with TTL=0 or inconsistent with OLSR specifications are also discarded.

Hello messages

- Hello messages are used to
  - verify if links are up and running (link sensing)
    - ✓ if no hello message is received by a neighbor in a given interval a timeout occurs and the link is assumed down
    - ✓ to exchange with neighbors neighborhood information (piggybacked in the hello messages)
      - allows to compute two hop neighborhood
        - » Which in turn is needed to select multipoint relays

How MPRs are selected

- Each node X selects its MPRs among its one hop neighbors
- The set is selected to cover node X 2-hop neighborhood
  - MPR(X) is an arbitrary subset of node X one hop neighbors such that each node z in node X' s two hop neighborhood have a neighbor in MPR(X)
    - ✓ can be selected with a greedy protocol
      - MPR(X)=null, C(X)= two hop neighborhoos of X
      - For each neighbor Y of X, its degree D (Y) is computed without considering X and its neighbors
      - Y is included in MPR(X) if its the only neighbor of X able to cover a two hop neighbor
        - » C(X)=C(X) \ {nodes covered by Y}
      - till (X)=null
        - » Include in MPR(X) the neighbor of X which allows to cover more uncovered nodes in in C(X) (ties broken based on degree D)
        - » C(X)=C(X) \ {nodes covered by selected neighbor}
  - The smaller MPR(X) the less control overhead exchanged

How are control packets disseminated

Upon receiving a message *m* at node Y

- If the received message is not a duplicate, is valid and has a non zero TTL
  - if it is received by an MPR selector of Y
    - ✓ retransmit ***m***
      - ***reduce by one the message TTL***
      - ***increase by one the message hop count***
      - ***broadcast on all node Y interfaces***
    - ✓ update or create the entry for the message in the duplicate set (→upon receiving the same message the node can identify it was already received and retransmitted→ can be discarded)

SAPIENZA
UNIVERSITÀ DI ROMA

How are data packets forwarded

- Information on topology are disseminated to all the network nodes
  - in an efficient way
    - ✓ exploiting the backbone of multipoint relays
    - ✓ limiting as much as possible topology information
- Each node then locally runs a shortest path algorithm to determine paths to the different destination
  - fills a routing table
  - upon reception of a data packet forwarding is performed according to the routing table