Mobile Systems M

Alma Mater Studiorum – University of Bologna
CdS Laurea Magistrale (MSc) in
Computer Science Engineering

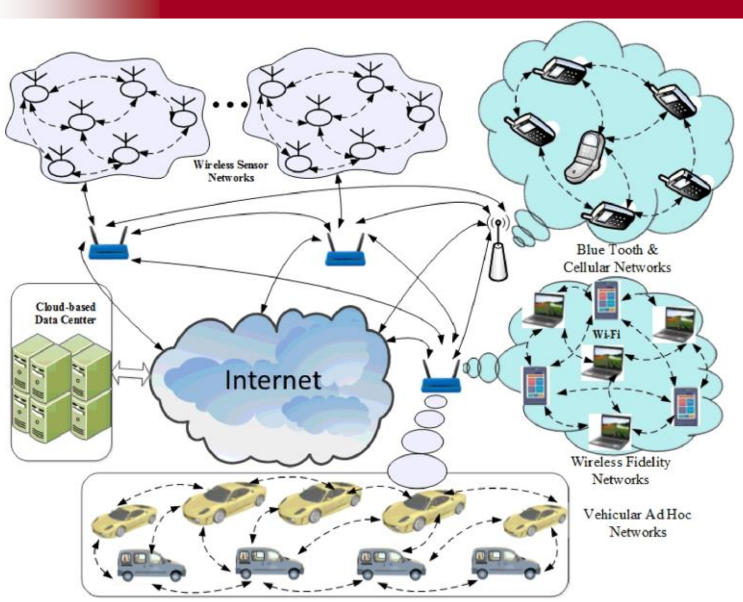
Mobile Systems M course (8 ECTS)
II Term – Academic Year 2021/2022

02 – Mobile Ad Hoc Network (MANET) and Routing

Paolo Bellavista
paolo.bellavista@unibo.it

http://lia.disi.unibo.it/Courses/sm2122-info/

What is exactly an Ad Hoc Network?



Primary Features of Wireless Ad Hoc Networks

- ***Created dynamically*** (on-the-fly) to satisfy needs and reqs that are typically ***temporary***
- ***Immediate and highly reconfigurable deployment*** (NO fixed infrastructure)
- High "volatility"
  - ***Mobility, failures/faults***, node resources that vary over time
- Nodes with very differentiated features (***heterogeneity***)
- Nodes with ***limited energy (battery-operated)***
- Any node can play the ***role of potential router***
  - ***Multi-hop communications***

Tech Challenges for Ad Hoc Networks

- Limited transmission range
- Broadcast nature of the wireless medium (e.g., hidden terminal)
- Packet loss due to transmission errors
- ***Mobility***
  - ***Modifications to routing and established paths due to mobility***
  - ***Packet loss*** induced by mobility
  - ***Network partitioning*** is possibly frequent
- Energy constraints
- Easy "snooping" of wireless transmissions (associated security issues)

Possible Application Areas for MANETs

But a ***vast spectrum*** of possible application areas :

- Personal Area Networking
  - Cellphones, laptops, wrist watches, human body sensors, …
- Civil environments
  - Meeting rooms, stadiums, ships/planes groups, …
- Military environments
  - War scenarios, realization of dynamic coalitions while in the war field, lack of infrastructure in enemy fields/areas
- Rescue/emergency operations
  - Search&rescue, police actions, firemen, …
- Sensor and actuator networks
  - Groups of sensors/actuators embedded in the environment (e.g., smart home) or "scattered" in geographical wide area

Several Variants are Possible…

- Fully ***symmetric*** environments
  - Any node has the ***same capabilities and responsibilities***
- ***Asymmetric capabilities***
  - Different coverage ranges and differentiated ***wireless*** transmission techniques
  - Different ***battery life***
  - Different ***computing capabilities***
  - Different ***mobility degrees*** (e.g., speed ranges)
- ***Asymmetric responsibilities***
  - Only some nodes can perform ***packet routing***
  - Only some nodes play the role of ***leader*** for their neighbors (e.g., clusterheads)
- Differentiated ***traffic characteristics***
  - Bandwidth, latency, reliability; unicast/broadcast/multicast/geocast

Several Variants are Possible…

- They can also ***co-exist and cooperate*** with infrastructure-based networks
- Different ***mobility patterns***
  - People seated in waiting rooms (*limited mobility*)
  - Taxi cabs (*high mobility*)
  - Military movements (most of them are *clustered*?)
  - Personal area networks (also in this case, most movements are *clustered*?)
- ***Mobility features***
  - Speed
  - ***Predictability*** (direction, pattern, triggers, …)
  - ***Uniformity or lack of uniformity*** in the mobility of different cooperating nodes

Routing in MANETs: Overview

First issue: ROUTING

- Why MANET routing is specifically hard and challenging?
  The answer to you ☺…
- ***3 routing protocols***, described below
  - Dynamic Source Routing (***DSR***)
  - Ad hoc On-demand Distance Vector routing (***AODV***)
  - Greedy Perimeter Stateless Routing (***GPSR***)

And, in addition, some elements of ***the more sophisticated TORA***

How to Properly Perform Routing in MANETs?

- Usually ad hoc networks ***involve mobile nodes***
  - Most relevant exception (only partial): Wireless Sensor Networks (WSN)
  - Thus, mainly Mobile Ad hoc NETworks (MANETs)
- ***Several routing protocol proposals in the related literature***
  - Some of them specifically designed for MANETs
  - Other ones adapted from existing protocols, previously proposed for usage in wired networks
- ***No single protocol has demonstrated to be optimal*** in any possible deployment environment and scenario
  - Some proposals also towards the development of ***adaptive protocols***

Why Routing is Different in MANETs?

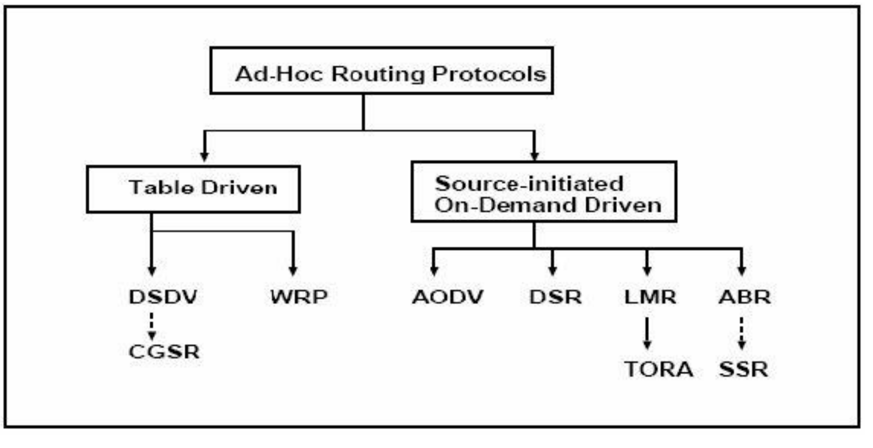
- ***Host mobility***
  - Link failure/repair operations in response to mobility may have different characteristics if compared with management operations reacting to other problems
- ***Frequency*** (*rate*) of ***link failure/repair operations*** may be high in the case of high mobility
- Need of exploiting ***new criteria for performance evaluation***, for example
  - ***Stability*** of routing paths ***depending on mobility***
  - ***Energy consumption***

MANET Routing Protocols

- ***Proactive protocols***
  - ***Maintain valid routes independently on ongoing traffic***
  - Generally, minor latency and greater overhead
  - Traditional routing solutions such as link-state and distance-vector are proactive
- ***Reactive protocols***
  - ***Maintain valid routes only if needed*** (on-demand)
- ***Geographic protocols***
  - Usage of knowledge of destination ***location*** to perform forwarding
- ***Hybrid protocols***

Which is the best approach? ***It depends on traffic and mobility patterns***

Just for curiosity, other taxonomies are more than possible



Trivial Solution: Flooding

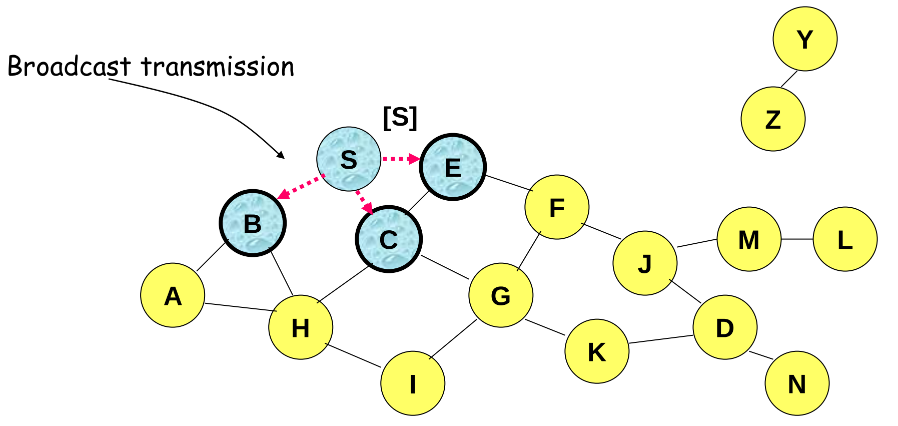
- ***Advantages***
  - Simplicity
  - More efficient when ***transmission frequency is very low*** (no need of discovery/maintaining valid routes or paths)
  - ***Potentially higher reliability*** (exploitation of *multiple paths*)
  - More suitable for ***high mobility patterns***
- ***Disadvantages***
  - Potentially high overhead
  - Potentially low reliability (broadcast exploitation, ***no reliable broadcast*** always available at low-layers of the employed wireless connectivity protocol)

Some protocols use ***flooding for control packets***, typically for routing discovery (overhead mortgaged over the successive longer sequence of data transmissions)

Dynamic Source Routing (DSR) (Johnson&Maltz, CMU, 1996)

- ***Source routing***: it is the source that tries to establish and embeds ***the whole path*** (from source to destination) in the exchanged packets
- How does the source determine the valid path in DSR?
  - When a node S is willing to send a packet to node D, but it does not know yet a valid route to D, ***S starts an operation of route discovery***
    - S performs ***flooding of a Route Request*** (RREQ) packet
    - Any node ***appends its own identifier*** to the packet header when forwarding the received RREQ packet

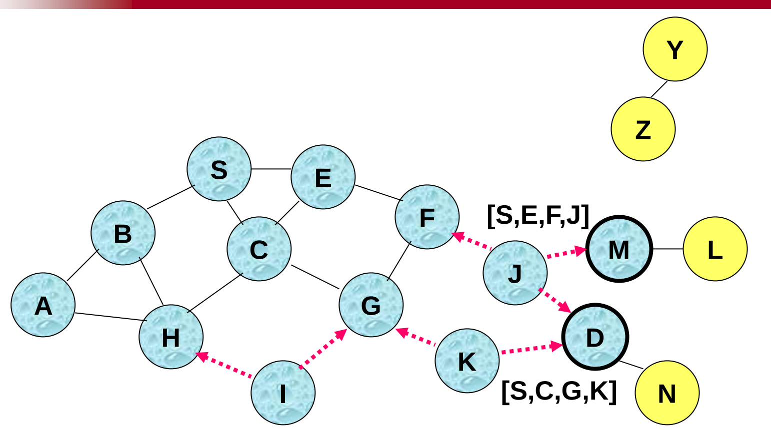
Route Discovery in DSR (1)



Represents RREQ transmission

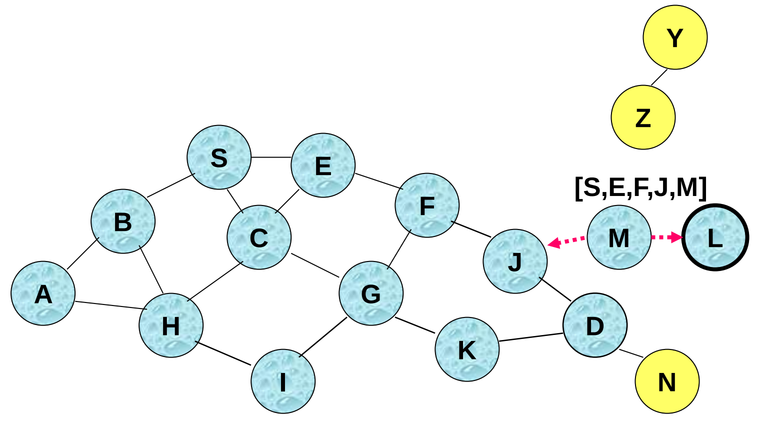
[X,Y] Represents the list of identifiers appended to RREQ

Route Discovery in DSR (2)



- Nodes J and K both perform RREQ broadcast to node D
- Since nodes J and K may be ***hidden nodes*** the one of the other, their ***trasmissions may be colliding***

Route Discovery in DSR (3)

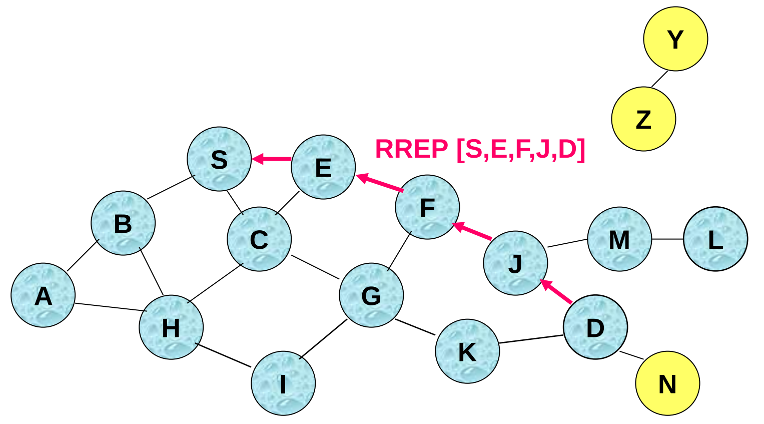


Node D ***does not perform forwarding*** of the RREQ packet because it realizes to be the ***desired destination*** for the route discovery operation

Route Reply in DSR

- Destination D, once received the first RREQ packet, sends a reply packet called ***Route Reply (RREP)***
- RREP is sent on the ***inverse path*** wrt the one contained in the received RREQ packet
- RREP ***includes data about the path*** from S to D, i.e., the one used by RREQ to reach D

Route Reply in DSR: Example

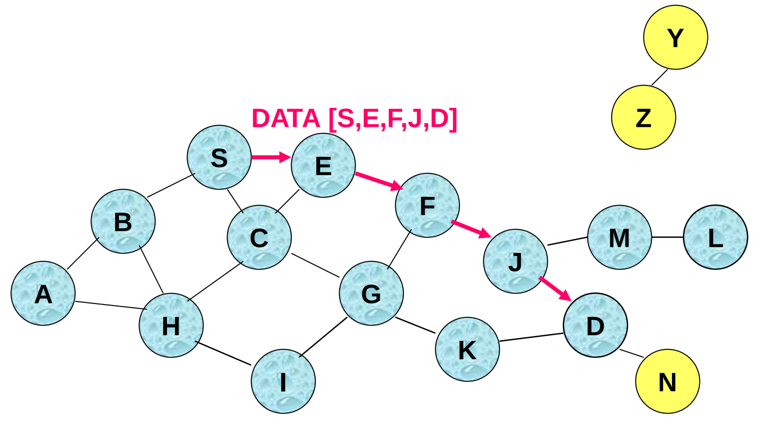


Represents the RREP control message

How to Perform Data Routing in DSR?

- Node S, after receiving RREP, can ***cache the path*** included in the RREP message
- When node S is willing to send a data packet to D, ***the whole routing path is included in the packet header*** (this is the reason why this is called source routing)
- ***Intermediary nodes use the source route*** included in the data packet to ***determine to which node the packet has to be forwarded***

Data Messages in DSR

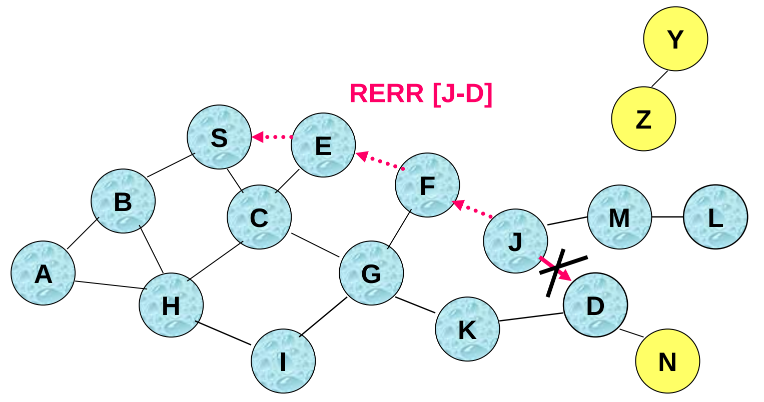


The packet header size grows with the path length

Path Caching in DSR

- ***Path caching*** (or *route caching*) is an add-on optimization
- Any node can perform caching of new paths that it happens to discover, in any possible way
- Advantages
  - ***Accelerates*** the route discovery process
  - ***Reduces*** the ***RREQ propagation*** process
  - Helps the exploitation of additional alternate paths
- Disadvantages
  - ***Invalid caches*** (*stale caches*) may negatively affect on the overall performance
    - How to invalidate the distributed caches?

Route Error (RERR)



- ***J sends an RERR packet to S*** along the JFES path when its forwarding of a data packet from S to D fails, e.g., due to node mobility
- Nodes that listen to the RERR packet can update their path cache and ***remove the JD link***

DSR: Pros and Cons

- ***Advantages***
  - Paths are maintained only among nodes that need to communicate (reduced overhead)
  - Caching can reduce the overhead associated with routing discovery
  - Each discovery can lead to the determination of ***multiple paths*** to destination because of intermediaries that reply based on local caches
- ***Disadvantages***
  - ***Growth of packet header size***
  - ***RREQ flooding***
  - Necessary mechanisms to avoid RREQ collisions among neighbors
  - Increase of channel conflicts when sending RREP (*RREP storm issue*; overhearing and local decision based on shortest path)
  - RREPs that use ***stale cache*** (affecting other caches in cascading)
    - Static timeout for caching, or
    - Adaptive timeout based on expected mobility, statistics about link usage, probability of link failure

Ad hoc On-demand Distance Vector (AODV)

(Perkins&Royer, Sun&UCSB, 1999)

DSR may lead also to ***large-size headers*** and consequent performance degradation

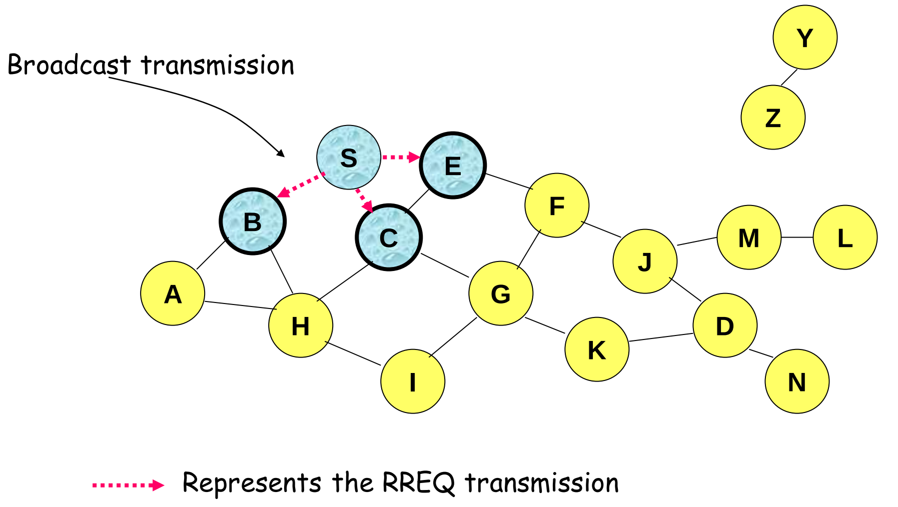
- In particular, when typical payloads are small

- ***AODV*** tries to improve the DSR efficiency ***by maintaining lightweight routing tables, suitable for MANET nodes***
  - Data packets do not include path info at all
- AODV maintains the positive feature of DSR that ***paths are stored only on the nodes that need to communicate*** (by need)

AODV: Basic Idea

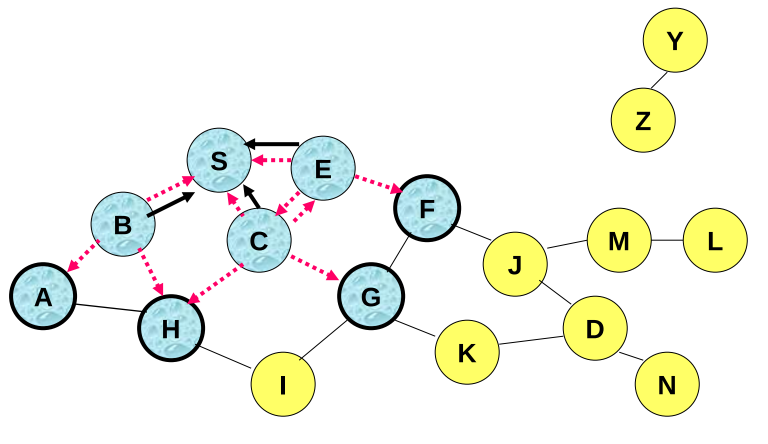
- ***Route requests (RREQ) are forwarded similarly*** to analogous packets in ***DSR***
- When a node performs re-broadcasting of a RREQ packet, it initializes and starts an ***inverse path that is directed*** to the source node
- When the target destination receives an RREQ, it replies with a ***Route Reply (RREP) packet***
- ***RREP travels along the inverse path that is configured during the forwarding chain*** of RREQ and ***consequently configures the entries of the routing tables*** only of the ***traversed nodes***

RREQ/Reverse Path Setup in AODV (1)



27

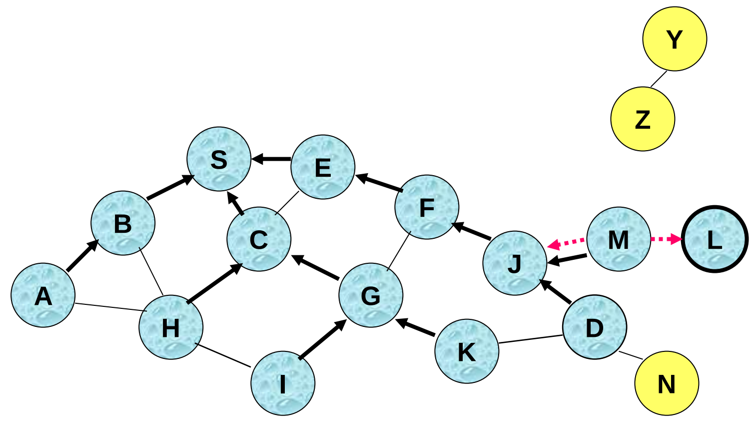
RREQ/Reverse Path Setup in AODV (2)



Represents the links for the inverse path
Backpointers are stored over the path nodes

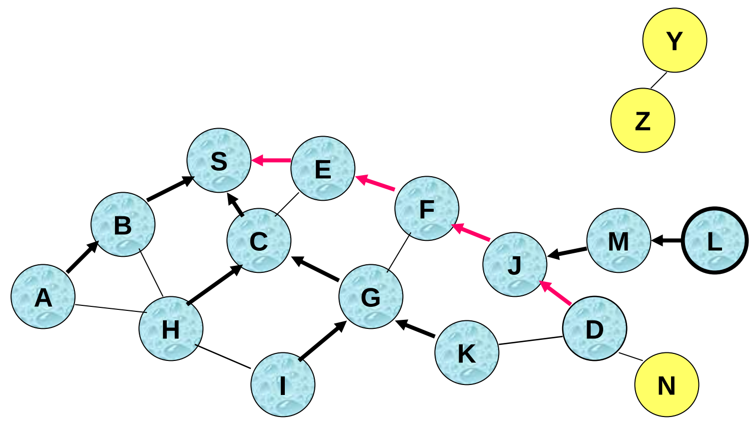
28

RREQ/Reverse Path Setup in AODV (3)



D does NOT perform RREQ forwarding because it is THE destination of RREQ

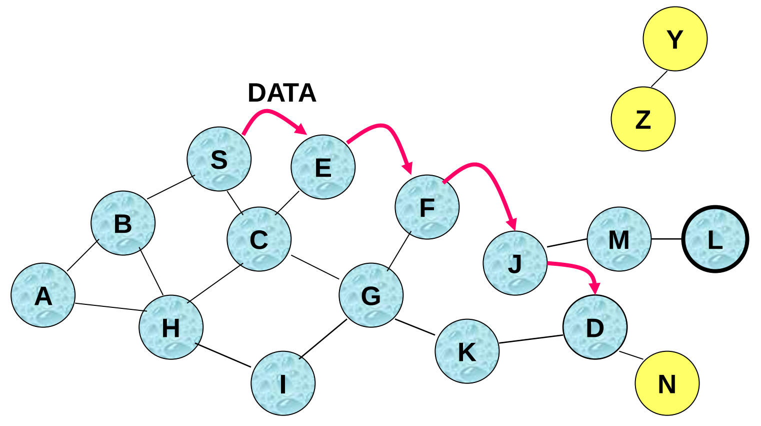
Route Reply in AODV



← Represents the link on the path used by RREP

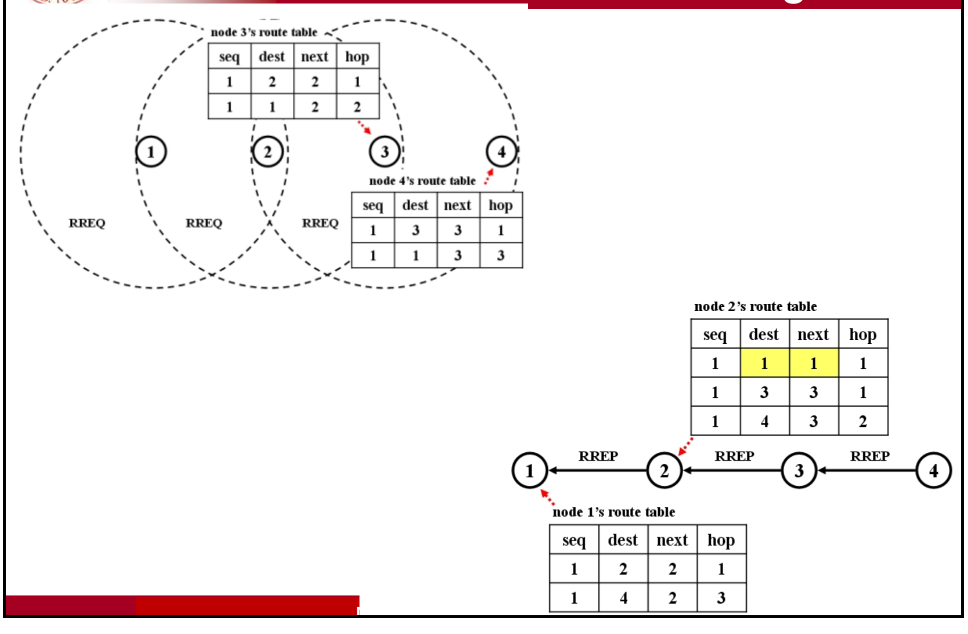
Forward links are configured when the RREP packet passes through the inverse path

Data Transmission in AODV



- The entries of the local routing tables are used to perform forwarding of data packets
- Differently from DSR, the path is not included in the header

Examples of AODV Routing Tables



node 3's route table

seq	dest	next	hop
1	2	2	1
1	1	2	2

node 4's route table

seq	dest	next	hop
1	3	3	1
1	1	3	3

node 2's route table

seq	dest	next	hop
1	1	1	1
1	3	3	1
1	4	3	2

node 1's route table

seq	dest	next	hop
1	2	2	1
1	4	2	3

Timeout

- Any entry of a routing table that includes an ***inverse path is discarded after a given timeout***
  - Why? If ***RREQ did NOT get to reach its destination, or if RREP did NOT correctly return back***, the related entry would occupy local memory in a completely useless way
  - ***Timeout must be sufficiently long*** to allow ***RREP packets to return back***
- Any entry of a routing table that includes a ***forward path*** is removed if not used for a given interval called active_route_timeout (longer than the timeout for inverse paths)
  - Why? The path may ***become invalid in short time*** in highly mobile networks

Reporting of non-usable Links

- ***A neighbor node is considered active*** for one entry in the routing table if ***one of its packets has been forwarded by using that entry*** in the last active_route_timeout time interval
- When a ***link*** towards a next node included in the routing table ***fails***, all ***active neighbors are informed***
- A node generates ***RERR in response to a broken path*** to destination D
  - When S receives RERR, it starts a ***new route discovery process*** towards D

In addition: Link Failure Detection

- ***Hello messages***: ***neighbor nodes periodically exchange alive messages***
- ***Lack of hello messages*** is used as an indication of possible ***fault/failure of a link***
- Alternatively, ***the lack of a series of received ACKs at the MAC layer*** can be used as an indication of probable link failure *(cross-layer monitoring)*

How to Limit Flooding during the Phase of Route Discovery?

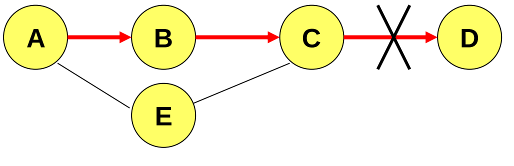
- Optimization: ***gradual expansion of the search, ring shaped***
- RREQ messages are sent initially with ***limited TTL***, in order to limit their propagation
  - DSR also may exploit (and several versions of it do that) a similar optimization
- If no RREP message is received, then the approach is to ***try again with larger TTL***
  - Sending of a new RREQ

Therefore, we are looking for a more balanced ***tradeoff among which factors***?

AODV – Optimization

- ***Possible additional optimization***: an intermediate node with a route to D can reply to RREQ
  - Faster operation
  - Decreases the issue of route request flood
- This optimization can cause loops in presence of link failures

AODV: Routing Loops

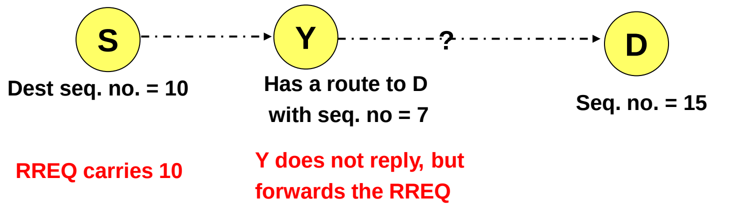


- Assume that link C-D fails and node A does not know about it (RERR packet from C is lost)
- C performs a route discovery for D
- Node A receives the route request (via path C-E-A)
- Node A replies, since A knows a route to D via node B
- ***Results in a loop: C-E-A-B-C***

AODV: Sequence Numbers

- Each node X maintains a ***sequence number***
  - acts as a time stamp
  - incremented every time X sends any message
- Each route to X (at any node Y) also has X's sequence number associated with it, which is Y's latest knowledge of X's sequence number
- ***Sequence number relates to 'freshness' of the route*** – higher the number, more up to date is the route

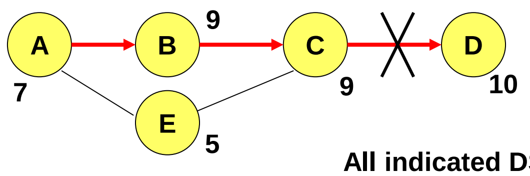
Use of Sequence Numbers in AODV



Loop freedom: intermediate node replies with a route (instead of forwarding request) only if it has a route with a higher associated sequence number

AODV: Avoidance of Loops

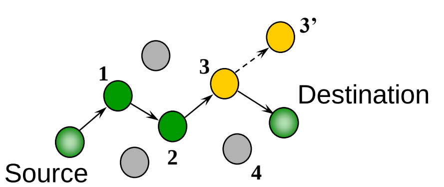
DSN = Destination Sequence Number

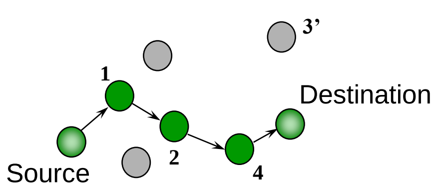


All indicated DSNs are for D

- Link failure increments the DSN at C (now is 10)
- If C needs route to D, RREQ carries the DSN (10)
- A does not reply as its own DSN is less than 10

Mobility-related Path Maintenance





- Movement not along the active path triggers no action
  - If source moves, reinitiate route discovery
- When destination or intermediate node moves
  - upstream node of break broadcasts RERR messages
  - RERR contains list of all destinations no longer reachable due to link break
  - RERR propagated until node with no precursors for destination is reached

Greedy Perimeter Stateless Routing (GPSR; Karp&Kung, Harvard, 2000)

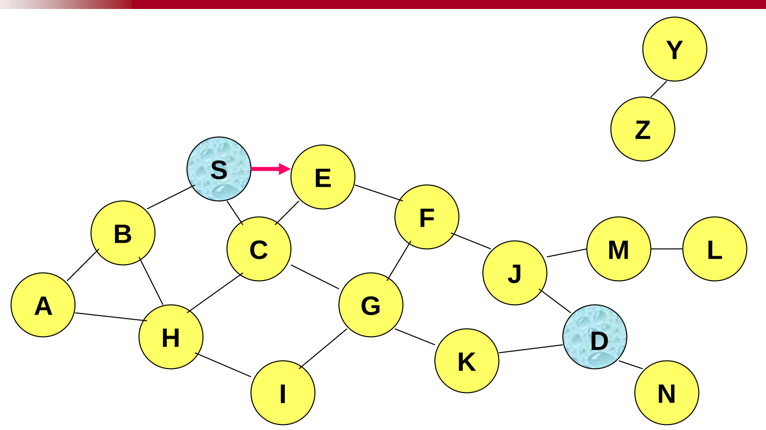
Geographic routing exploits location information to facilitate reaching the destination

- Assumption#1: ***source node knows the destination location***
- Assumption#2: nodes maintain ***lists of neighbor nodes and their locations***
    - Need to include ***location info in hello messages*** (beacons) that are periodically exchanged

Two schemes for data forwarding:

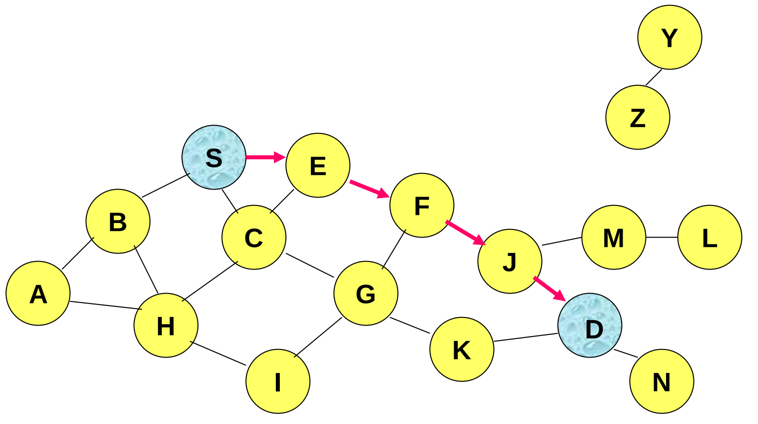
- ***Greedy forwarding***: data are sent to the neighbor node that is estimated as the closest one towards the destination (usage of only the location info of neighbor nodes for data forwarding)
- If ***greedy forwarding fails***, switch to a different scheme, i.e., ***perimeter forwarding***

Greedy Forwarding (1)



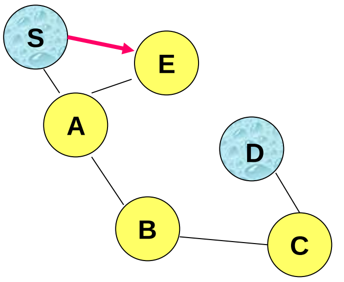
E is the S's neighbor that is closest to D
("closest" in terms of Euclidean distance)

Greedy Forwarding (2)



F is the E's neighbor node closest to D
J is the F's neighbor node closest to D

Possible Failures in Greedy Forwarding



In the case that E is not in the coverage range of D (assumption of the figure)

No node among the E's neighbors is closer to D than E
Forwarding failure!
But a useful path would exist: [S, A, B, C, D]

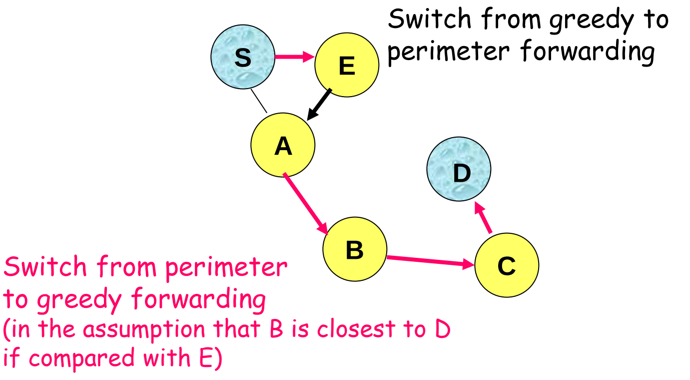
Perimeter (Face) Forwarding

- It can ***always reach a destination if a useful valid path exists***
  - Route ***around the so-called "holes"***
- Each node calculates ***Relative Neighborhood Graph (RNG) or Gabriel Graph (GG)***
  - RNG is a ***non-directed graph*** defined on a set of points in the Euclidean plane that are compliant with this constraint: connecting two points A and B with an arc ***if and only if there is no point C that is closest to both A and B (C-to-A and C-to-B distances minor than A-to-B distance)*** - G. Toussaint, 1980
- ***RNG is traversed*** by using the right-hand rule
  - Basically, the idea is of ***visiting the nodes that determine the perimeter*** around a hole

Perimeter (Face) Forwarding

- During graph traversing, if a packet meets a node that is closest to destination if compared with the node where greedy forwarding had failed, ***the decision is to operate a new switch towards greedy forwarding***
- ***We can have loops*** if perimeter forwarding is used and whenever the destination is not reachable
  - GPSR is capable of detecting the situation and of discarding the involved packet

Example



Temporally Ordered Routing Algorithm (TORA)

TORA is proposed to operate in a ***highly dynamic*** mobile networking environment

- ***Highly adaptive, loop-free, highly distributed***
- Based on the concept of ***link reversal***

Key design concepts of TORA:

- ***Localization of control messages*** to a very small set of nodes ***near the occurrence of a topological change***
- To this purpose, nodes need to maintain ***routing info about neighbors***
- The ***height metric*** is used to model the routing state of the network

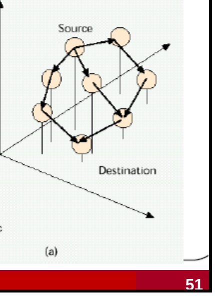
Three basic functions:

- ***route creation***
- ***route maintenance***
- ***route erasure***

Temporally Ordered Routing Algorithm (TORA)

During route creation and maintenance, nodes establish a Directed Acyclic Graph (DAG)

- ***A logical direction*** is imposed on links towards destination
- ***Source-initiated***
- Provides ***multiple routes*** for any desired source/destination pair
- Starting from any node in the graph, a destination can be reached by ***following the directed links***
- ***Highly adaptive, efficient, scalable, distributed algorithm***
- ***Multiple routes from source to destination***



TORA Major Tasks

Three major tasks

- Route creation – query (QRY) and update (UPD) packets
- Route maintenance
- Route erasure – broadcast of clear packet (CLR)

Using ***unique node ID and unique reference ID for packets***

1) Route creation: demand-driven «query/reply»

Performed only when a node requires a path to a destination but does not have any directed link

- A QRY packet is flooded
- An UPD packet propagates back if routes exist

2) Route maintenance: «link reversal» algorithm

- React only when necessary
- Reaction to link failure is ***localized in scope***

3) Route erasure

A CLR packet is flooded to erase invalid routes

TORA Metrics

- ***Assigns a reference level (height) to each node***
- ***A local DAG is maintained for each destination***
- ***Synchronized clock*** is relevant, accomplished via GPS or a dedicated protocol such as Network Time Protocol (NTP)

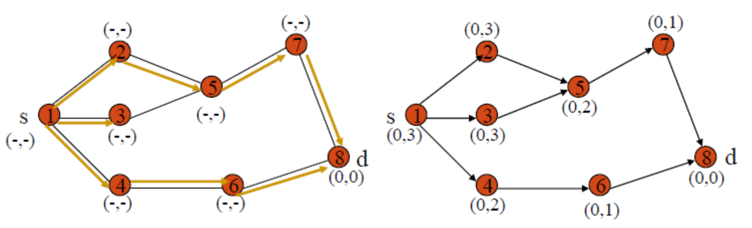
Timing is an important factor in TORA because the «height» metric is dependent on the logical time of a link failure

- Logical time of a link failure
- The unique ID of the node that defined the new reference level
- A reflection indicator bit
- A propagation ordering parameter
- The unique ID of the involved node

Adjust reference level to restore routes on link failure

Temporally Ordered Routing Algorithm (TORA)

- Route creation of TORA

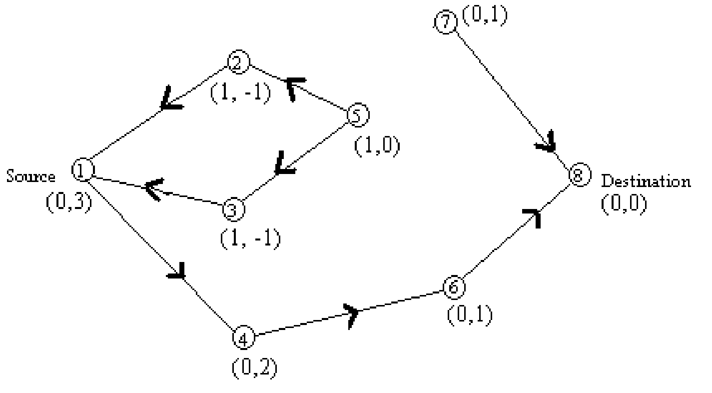


Propogation of QRY (reference level, height)

Height of each node updated by UPD

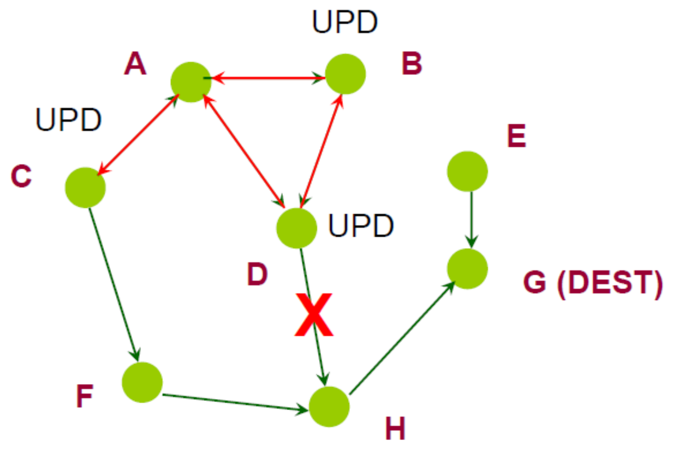
Temporally Ordered Routing Algorithm (TORA)

Route creation

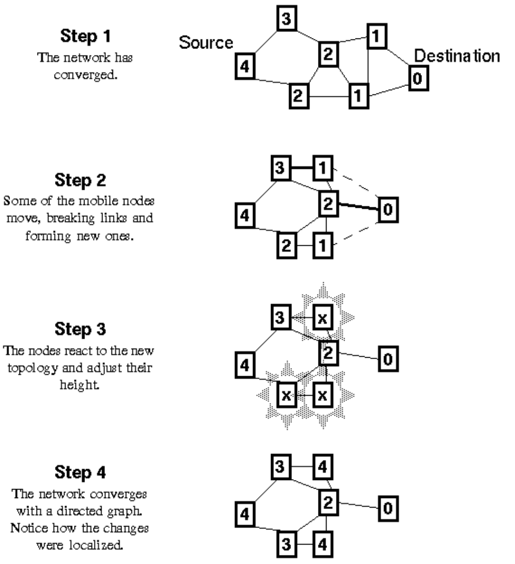


Temporally Ordered Routing Algorithm (TORA)

- Route maintenance



Temporally Ordered Routing Algorithm (TORA)



Temporally Ordered Routing Algorithm (TORA)

In summary…

Advantages:

- ***Less control overload*** – by limiting the control packets for route ***reconfiguration to a small region***

Disadvantages:

- Local reconfiguration of paths ***results in non-optimal routes***
- Concurrent deduction of partitions and subsequent deletion of routes could ***result in temporary oscillations and transient loops***

Moreover, many other MANET routing algos in the literature…

Think about optimizations that stem from

- Application requirements
- Most probable characteristics of deployment scenarios
- Rate between mobility&dynamicity vs communication rate
- Which information assumed to be known at participating nodes?
- Which node coordination and associated overhead?
- How much proactive? How much reactive?
- How much optimistic? How much pessimistic?

For instance: Multi-hop Routing vs. Energy Consumption

- Energy consumption to transmit a packet:
  - Constant cost to power on the circuitry
  - Proportional to packet size
  - Proportional to distance * distance

- Multi-hop routing can ***reduce the consumption*** of energy (the consumed energy is basically proportional to distance * distance) but this can generate non-negligible ***latencies***

- Which per-hop distance?
  - Too short => the dominant part of the energy cost is for powering on the circuitry
  - Too large => the dominant part is for packet transmission; ***reduction of re-usability of bandwidth in space***; overhead for scheduling because the number of nodes at 1-hop-distance grows

A short Parenthesis on Clustering

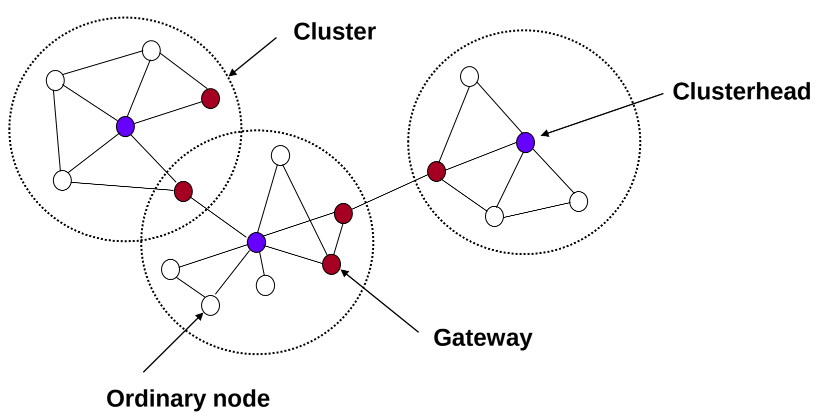
Clustering (grouping) to ***decrease resource consumption***

- Split the network in ***clusters (groups)***, each of them including an "analogous" number of nodes
- ***Clusterheads*** are the ***natural backbone*** also in order to perform routing
- ***Optimal clustering is an NP-complete problem***
- Very relevant: anyway mobility tends to ***degrade the optimality of the determined clustering***

Specific usefulness for sensor networks: to combine "cluster-level readings" into a single data packet (*data aggregation*)

61

Clustering as a Routing Backbone



62

Very shortly: Clustering Examples

LEACH

- ***Local decision*** if a node should serve as clusterhead or not (random number choice and completely local election)
- Any non-clusterhead node performs overhearing and ***selects the closest clusterhead***
- The clusterhead role ***is periodically re-assigned (node rotation)*** to balance energy consumption
- ***Communication is first to clusterhead, then to cluster members***
- ***No guarantee of optimality in clustering determination***

HEED

- ***Residual energy*** to consider in the clusterhead election
- Clusterheads are elected after an iterative protocol:
  - A node announces its ***intention and cost*** as a clusterhead
  - Any non-clusterhead node selects its candidate with minor cost by following a probabilistic metric, possibly choosing itself

Weighted Clustering Algorithm (WCA)
(Chatterjee et al., 2002)

- A clusterhead can ***ideally support n nodes***
  - Ensures efficient MAC functioning
  - Minimizes delay and maximizes throughput
- A clusterhead uses ***more battery power***
  - Does extra work due to packet forwarding
  - Communicates with more nodes
- A clusterhead should be ***less mobile***
  - Helps to maintain same configuration
  - Avoids frequent WCA invocation
- A ***better power usage with physically closer nodes***
  - More power for distant nodes due to signal attenuation

Is Wi-Fi Direct a MANET technology?

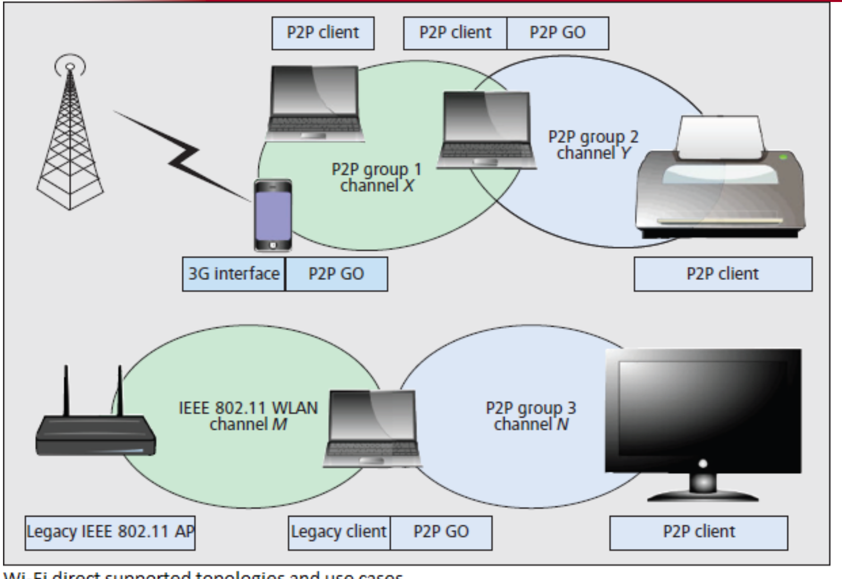
Given our current understanding of MANET, let us go back to Wi-Fi Direct…

- In a typical Wi-Fi network, client scans and associate to wireless networks available, which are created and announced by Access Points (AP)
- ***Wi-Fi Direct allows specifying these roles as dynamic***, and hence a Wi-Fi Direct device has to implement both the role of a client and the role of an AP
- These roles are therefore ***logical roles that could even be executed simultaneously by the same device***, this type of operation is called ***Concurrent mode***

Wi-Fi Direct Architecture

- Wi-Fi Direct devices communicate by ***establishing a P2P group***
- The ***device implementing AP-like functionality*** in P2P group is referred to as the ***P2P Group Owner (P2P GO)***, and device acting as client are known as P2P clients
- Once P2P group is established, other P2P clients can join the group as in a traditional Wi-Fi network
- When the device acts as ***both as P2P client and as P2P GO, the device will typically alternate between the two roles by time-sharing the Wi-Fi interface***
- Like a traditional AP, a P2P GO announces itself through beacons, and has to support power saving for its associated clients

Wi-Fi Direct Architecture



Wi-Fi direct supported topologies and use cases.

Wi-Fi Direct Architecture

- Only ***the P2P GO is allowed to cross-connect*** the devices in its P2P group to ***an external network*** (e.g., mobile in the previous figure)
- ***This connection must be done at network layer***, typically implemented using Network Address Translation (NAT)
- ***Wi-Fi direct does not allow transferring the role of P2P GO within the group***
- If P2P GO leaves the P2P group then the group is broken down and has to re-established

Parallel and comparison with Bluetooth scatternets?

Wi-Fi Direct Group Formation

Three types of group formation techniques: ***Standard, Autonomous, and Persistent cases***

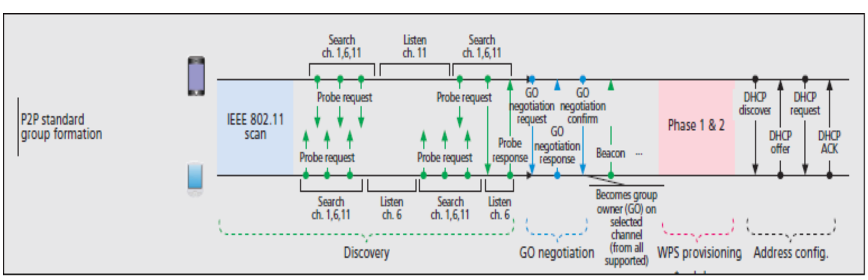
Group Formation procedure involves two phases:

- Determination of P2P GO
  - ***Negotiated*** - Two P2P devices negotiate for P2P GO based on desires and capabilities
  - ***Selected*** - P2P GO role established at formation or at an application level
- Provisioning of P2P Group
  - Establishment of ***P2P group session*** using appropriate credentials
  - Using Wi-Fi simple configuration to exchange credentials

Wi-Fi Direct Group Formation

Standard: P2P devices have to discover each other, and then negotiate which device will act as P2P GO

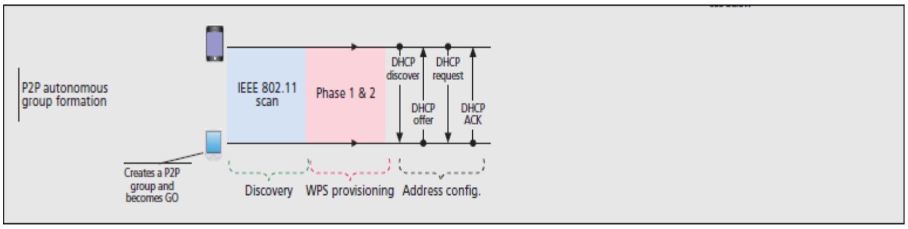
- Its starts by performing a traditional Wi-Fi scan, by means of which they can discover existing groups and Wi-Fi networks



- To prevent conflicts when two devices declare the same GO Intent, a tie-breaker bit is included in the GO Negotiation Request, which is randomly set every time a GO Negotiation Request is sent

Wi-Fi Direct Group Formation

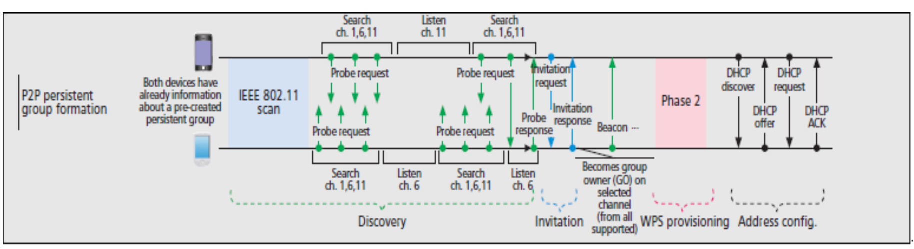
Autonomous: a P2P device may autonomously create a P2P group, where it immediately becomes the P2P GO, by sitting on a channel and starting a beacon



- ***Other devices can discover the established group*** using traditional scanning mechanisms
- As compared to previous case, ***discovery phase is simplified*** in this case as the device establishing the group does not alternate between states, and indeed no GO negotiation phase is required

Wi-Fi Direct Group Formation

Persistent: a P2P device can declare a group as persistent, by using flag in the P2P capabilities attribute present in beacon frames



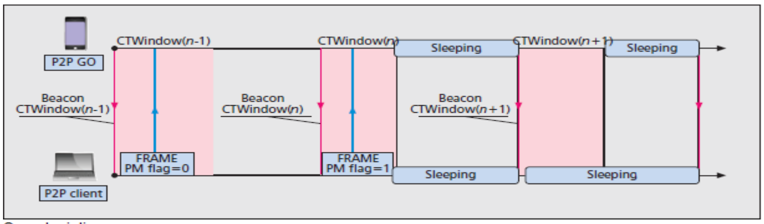
After the discovery phase, if a P2P device recognizes to have formed a persistent group with the corresponding peer in the past

any of the two P2P devices can use the ***Invitation Procedure to quickly re-instantiate the group***

Wi-Fi Direct Power Saving

Wi-Fi Direct defines ***two new power saving mechanisms***: the Opportunistic Power Save protocol and the Notice of Absence (NoA) protocol

Opportunistic Power Save protocol (OPS) allows a P2P GO to save power when all its associated clients are sleeping

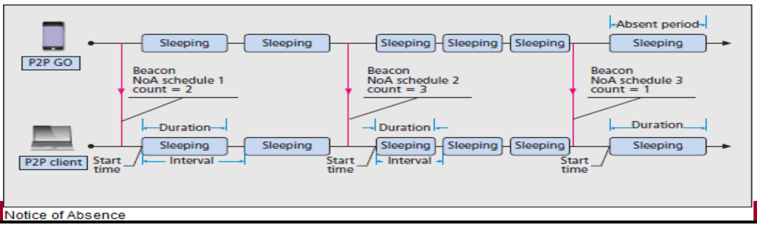


Opportunistic power save

Wi-Fi Direct Power Saving

Notice of absence protocol (NoA) allows a P2P GO to announce time intervals, referred to as ***absence periods***, where P2P Clients are not allowed to access the channel

- P2P GO defines a NoA schedule using four parameters:
  - Duration that specifies the length of each absence period
  - Interval specifying time between consec absence periods
  - Time that specifies the start time of the first absence period after the current Beacon frame
  - Count that specifies how many absence periods will be scheduled during the current NoA schedule



Notice of Absence

Wi-Fi Direct Security

- Wi-Fi Direct devices are required to implement ***Wi-Fi Protected Setup (WPS)*** to support a secure connection with ***minimal user intervention***
- WPS allows establishing a ***secure connection by introducing a PIN*** in the P2P Client, or ***pushing a button in the two P2P devices***
- Following WPS terminology, P2P GO is required to implement an internal Registrar, and the P2P Client is required to implement an Enrollee
- WPS operations consist of two parts
  - In the first part, the internal Registrar is in charge of generating and issuing the network credentials, i.e., security keys, to the Enrollee
  - In the second part, the Enrollee (P2P Client) disassociates and reconnects using its new authentication credentials

Wi-Fi Direct: some Refs to Additional Material

Optional additional readings:

- IEEE 802.11-2013 Standard, Device-To-Device communication with Wi-Fi direct: Overview and experimentation, 2007
- Wi-Fi Alliance, P2P Technical Group, Wi-Fi Peer-to-Peer (P2P) Technical Specification v1.0, December 2009
- Wi-Fi Alliance, Wi-Fi Protected Setup Specification v1.0h, Dec. 2006
- IEEE 802.11z-2010 - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 7: Extensions to Direct-Link Setup (DLS)

Other more Innovative Routing/Clustering Modes?

Under some ***simplifying assumptions***, which can significantly facilitate how to solve the problem

Again ***cross-layer*** or ***possible static assumptions*** about given and determined deployment environments

For example, ***content sharing*** scenarios in sport events with very large pubic of attendants (Olympic stadium in Turin 2006) and widespread distribution of pictures/videos recorded by spectators

- to provide an ***entertainment service***, e.g., small multimedia contents, dynamically discovered, to a ***large public*** of users ***concentrated in space and in time***
- to maintain ***content availability*** notwithstanding ingress/exit of spectators from the targeted physical locality

Dense MANET Assumption and Interaction with Application Layer

- Assumptions
  - ***Dense MANET***
    - ***Large number of devices co-located*** in a physical area that is relatively small
    - ***Node density*** that is almost ***invariant*** over relatively long time intervals
  - ***Replication and read-only replicas***

- Non-functional requirements
  - Low ***overhead*** → Lightweight and approximated protocols
  - High ***scalability*** → Complete decentralization
  - Sufficient ***accuracy*** → Protocol ending based on heuristics

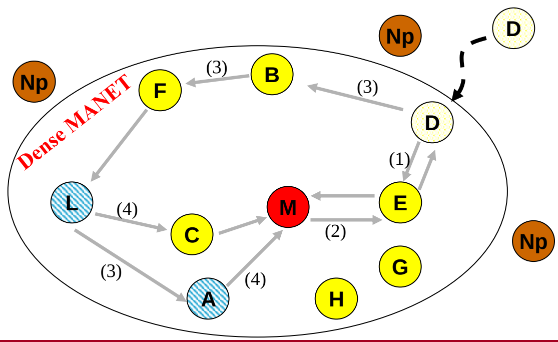
Replication in Dense MANETs (REDMAN)

Basic idea: to ***disseminate replicas*** of resources of common interest and to ***maintain the desired replication degree*** independently from node mobility (unpredictable) in/out the targeted dense area

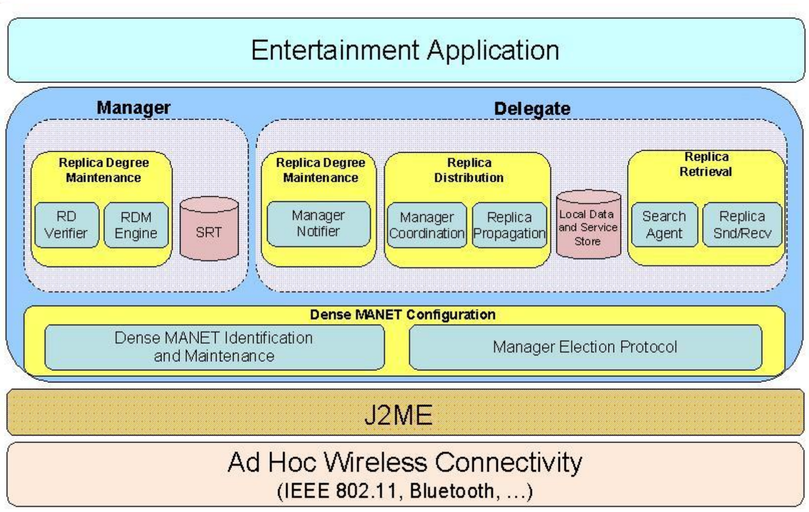
- ***Delegates host replicas***, reply to retrieval requests, participate to dissemination
- ***Managers***: responsible for maintaining the proper and desired replication degree

Shared Resource Table

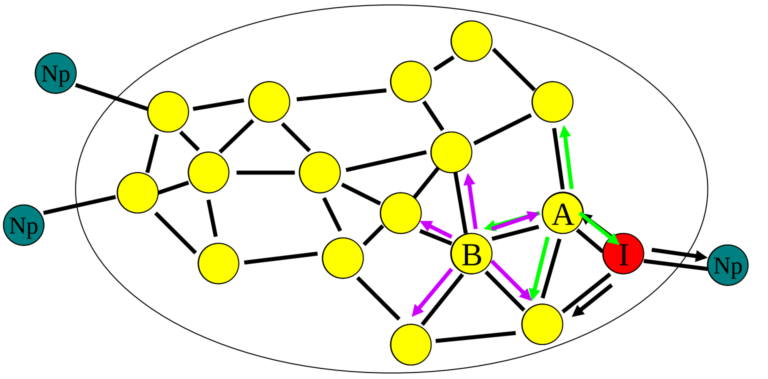
Resource Name	Replication Degree	Probable Replica Placement
Alberto Tomba's Picture1	3	D, L, A



Middleware Approach: Application-layer Management

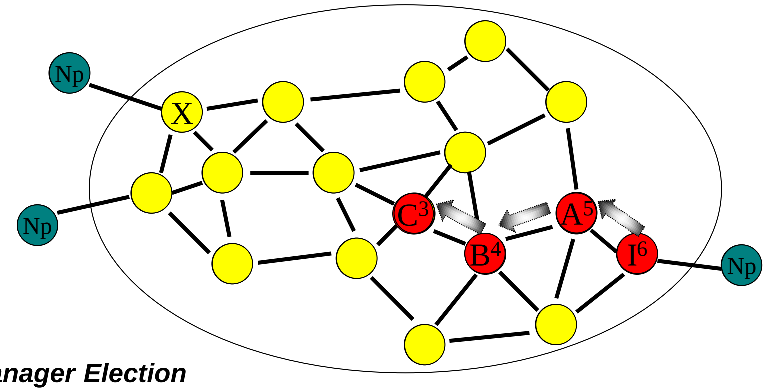


Issue: Identifying a Dense MANET



- Dense MANET if and only if #*Neighbors > Threshold*
- ***Decentralized and lightweight protocol*** in which any node ***autonomously decides*** its own belonging condition
- ***Dynamicity***: lazy updates based on hello messages

Issue: Manager Election



Manager Election

- Role is assigned to a node that is ***topologically central***
- Lightweight solution, ***no optimal placement*** (priority is avoiding exhaustive search)
- Exploration strategy based on heuristics

Dynamicity

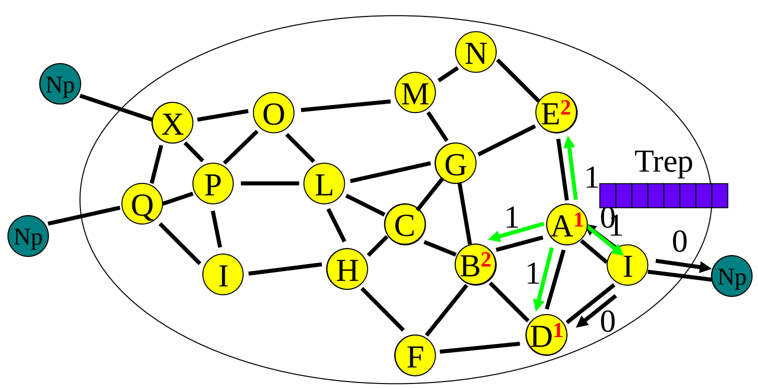
- Reactive response: new determination of farthest nodes every *Tr*
- Proactive response: new election every *Tp* >> *Tr*

Ending the Election Process

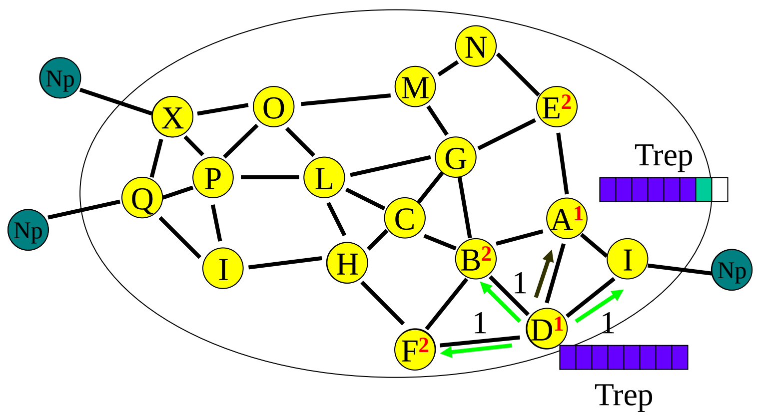
- ***Optimal solution is considered to be found iff***:
  1. *currentINvalue* = $\lceil$*worstExploredValue* / 2$\rceil$
- Alternatively, ***heuristics***:
  2. *currentINvalue ≤ worstExploredValue* * ***DesiredAccuracy***
  3. ***maxConsecutiveEqualSolutions*** have been explored without improving the current *bestValue*

Of course, ***DesiredAccuracy*** and ***maxConsecutiveEquals*** determine (approximatively) the quality of the solution achieved (quantitatve indicator)

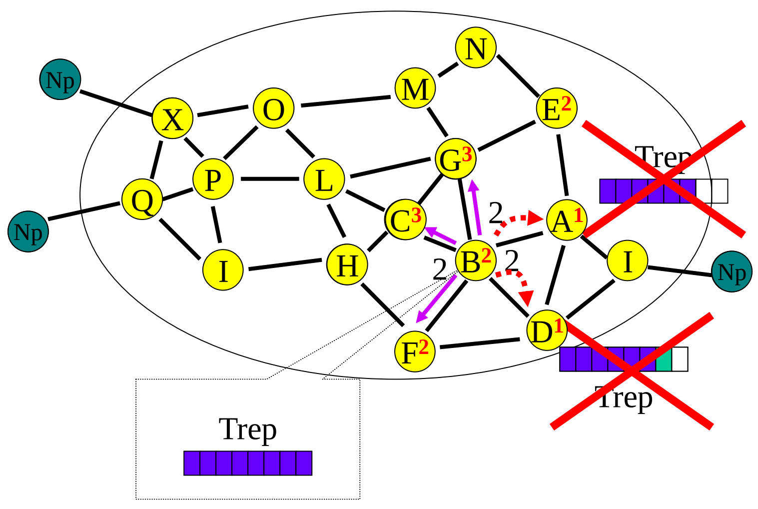
Identifying the Farthest Nodes



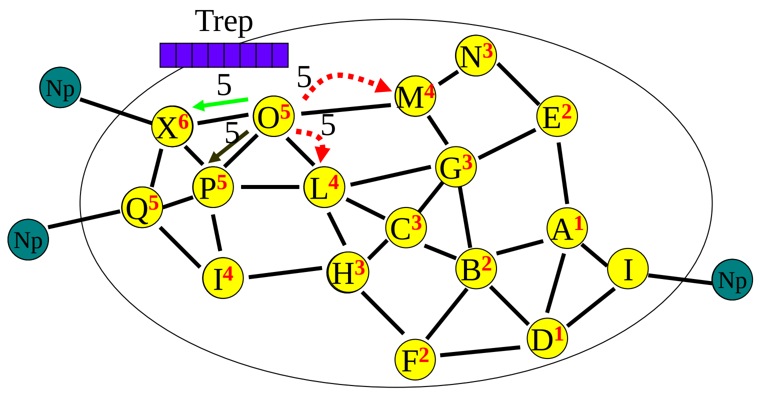
Identifying the Farthest Nodes



Identifying the Farthest Nodes

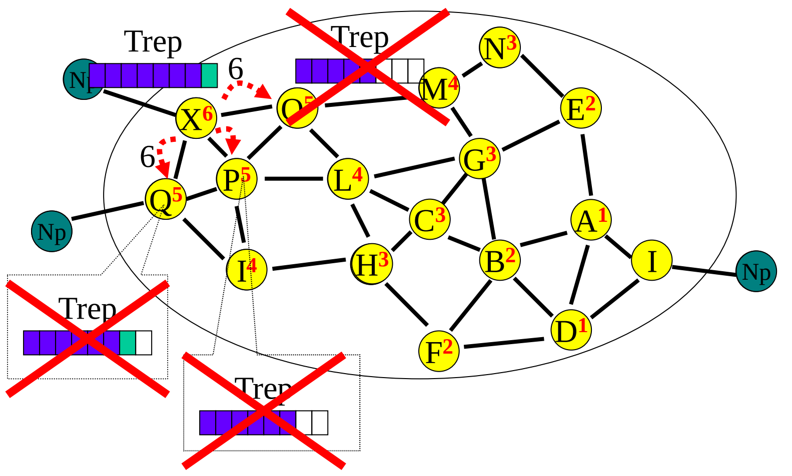


Identifying the Farthest Nodes



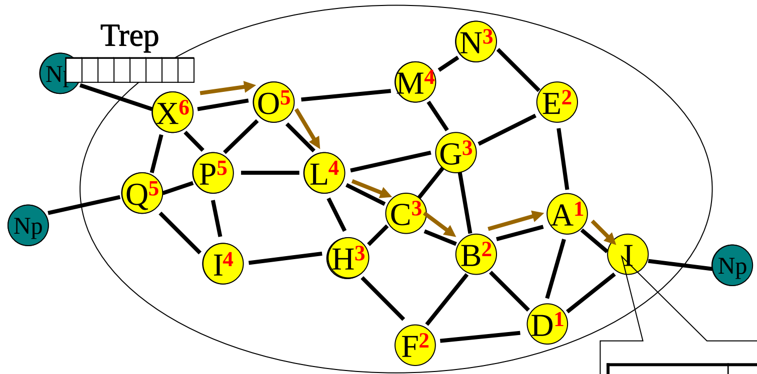
87

Identifying the Farthest Nodes



88

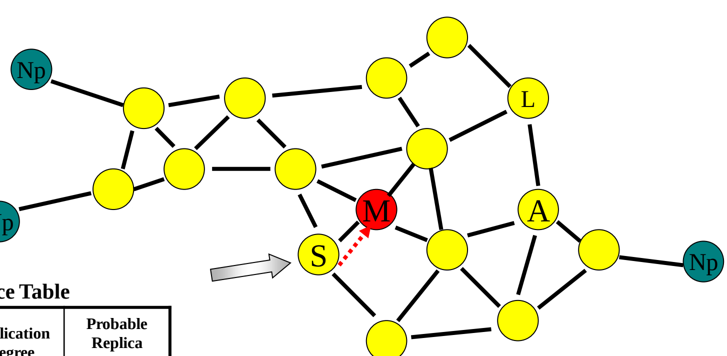
Identifying the Farthest Nodes



Farthest Node	Forwarder
X	A

Degree of Replication: Approximated Consistency

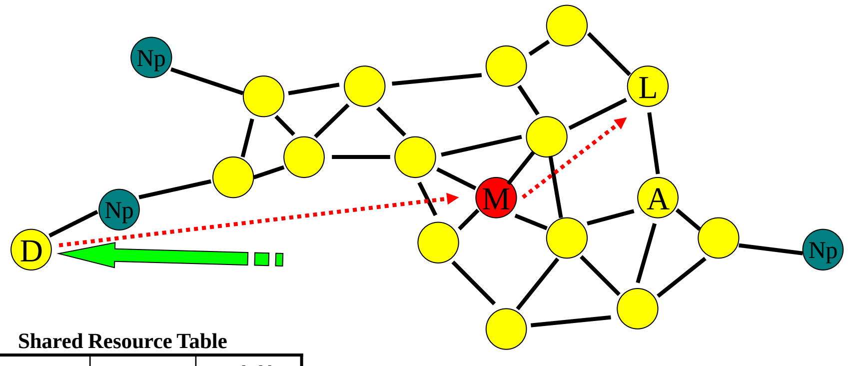
It relaxes the constraint of ***anytime perfect consistency*** for the number of available replicas



Shared Resource Table

Resource Name	Replication Degree	Probable Replica Placement
AlbertoTomba Picture1	3	L, A, S

Degree of Replication: Approximated Consistency

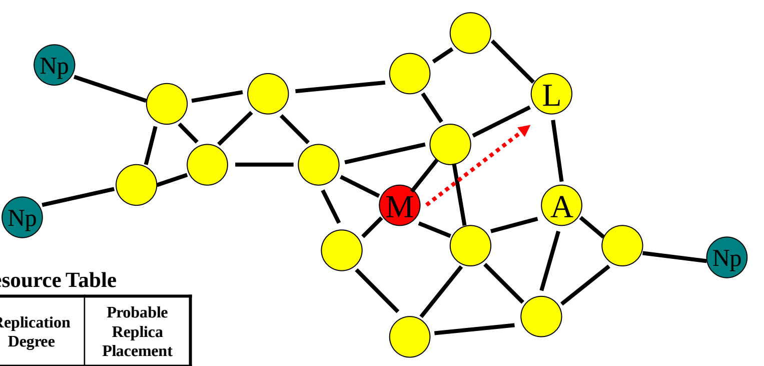


Shared Resource Table

Resource Name	Replication Degree	Probable Replica Placement
AlbertoTomba Picture1	3	~~D~~, L, A

91

Degree of Replication: Approximated Consistency



Shared Resource Table

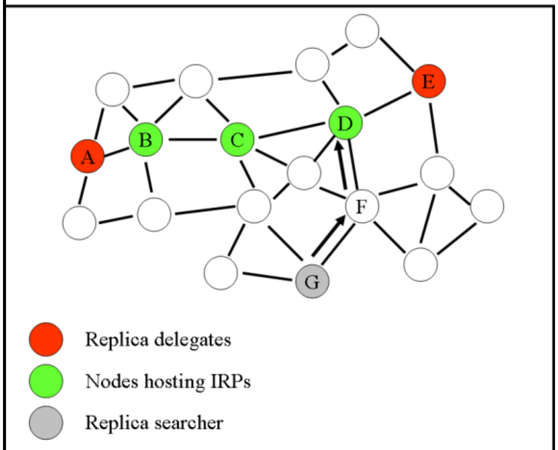
Resource Name	Replication Degree	Probable Replica Placement
AlbertoTomba Pic1	3	~~D~~, L, A

92

Strategies for Replica Dissemination

Different possible strategies:

- ***Random*** distribution
- ***Spatially uniform*** distribution
- …



REDMAN: distribution along ***"straight lines"*** (approxim.)

- No positioning equipment
- Straight lines: neighbors with the ***lowest number of neighbors shared*** with the previous nodes

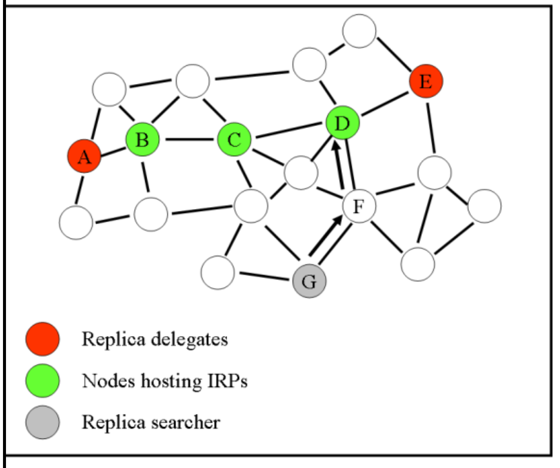
Strategies for Replica Retrieval

Different possible strategies for ***replica retrieval***:

- ***Query*** flooding (QF)
- Flooding of ***Information about Replica Placement*** (IRP)
- ***k-hop Distance IRP Dissemination (k-DID)***

REDMAN exploits ***Straight IRP Dissemination (SID)***

Strategies for Replica Retrieval



IRP are distributed along the ***same approxim. straight lines*** used for replica dissemination

Retrieval along straight lines (***non parallel*** to the lines used for dissemination)

Duality between replica distribution and retrieval