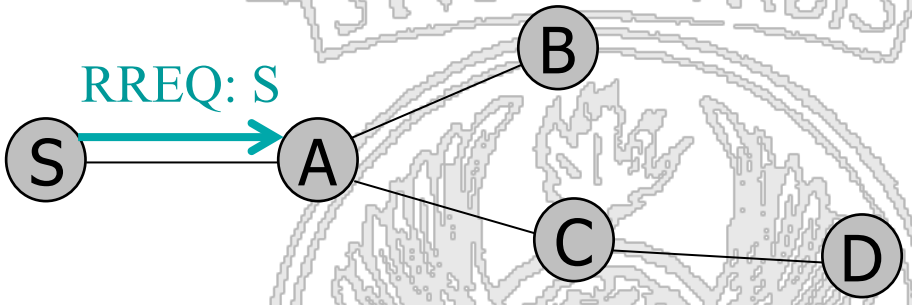
SAPIENZA
UNIVERSITÀ DI ROMA

Dynamic Source Routing (DSR)

- Reactive
- *Route discovery cycle* used for route finding
- Maintenance of *active* routes
- Utilizes *source routing*

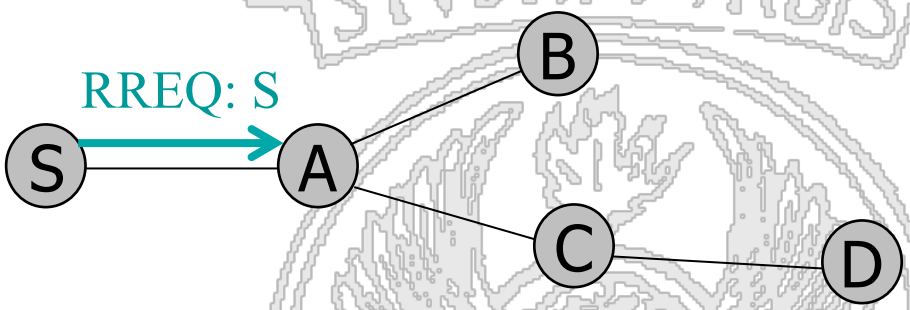
DSR: Route Discovery



1. Node *S* needs a route to *D*
2. Broadcasts RREQ packet
    1. RREQ identifies the target of the route discovery, contains a route record in which the traversed route is accumulated, contains a pair <initiator, request id> uniquely identifying the request

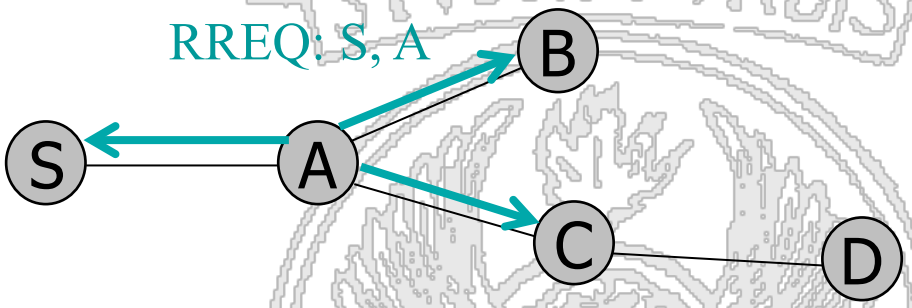
SAPIENZA
UNIVERSITÀ DI ROMA

DSR: Route Discovery



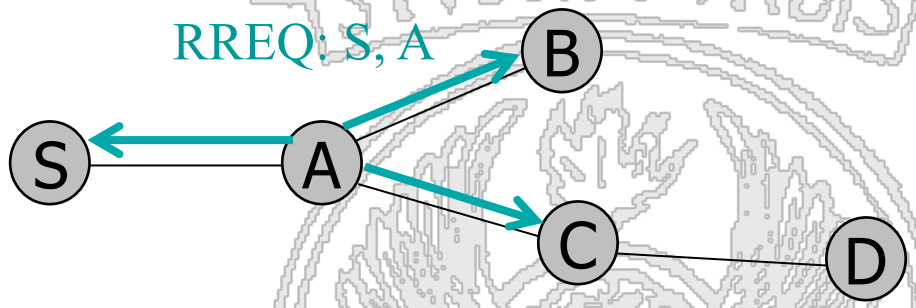
1. Node *S* needs a route to *D*
2. Broadcasts RREQ packet
3. Node *A* receives packet, has no route to *D AND is NOT D*
   - Rebroadcasts packet after adding its address to source route

DSR: Route Discovery



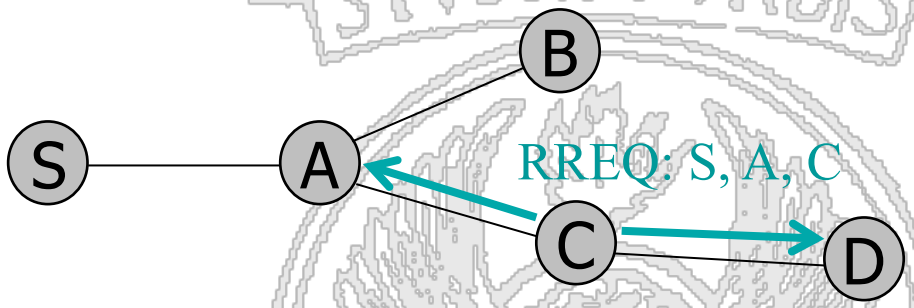
1. Node *S* needs a route to *D*
2. Broadcasts RREQ packet
3. Node *A* receives packet, has no route to *D*
   - Rebroadcasts packet after adding its address to source route

DSR: Route Discovery



4. Node *C* receives RREQ, has no route to *D*
   - Rebroadcasts packet after adding its address to source route

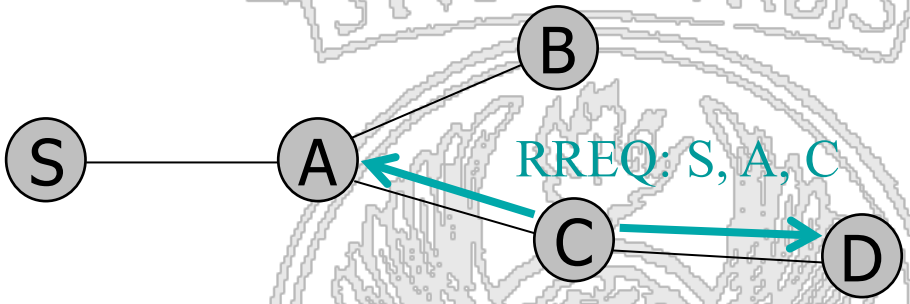
DSR: Route Discovery



4. Node *C* receives RREQ, has no route to *D*
   - Rebroadcasts packet after adding its address to source route

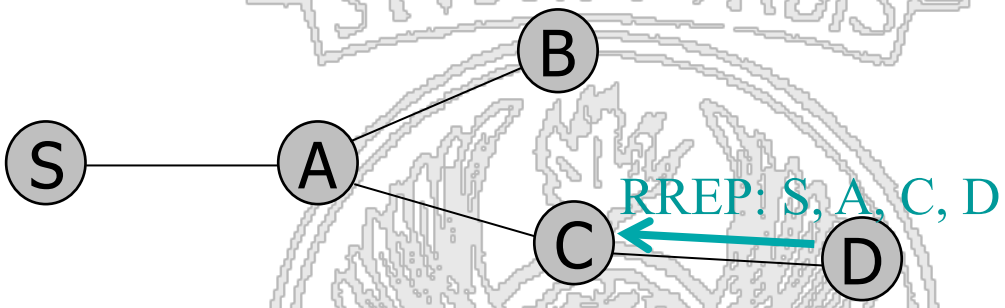
SAPIENZA
UNIVERSITÀ DI ROMA

DSR: Route Discovery



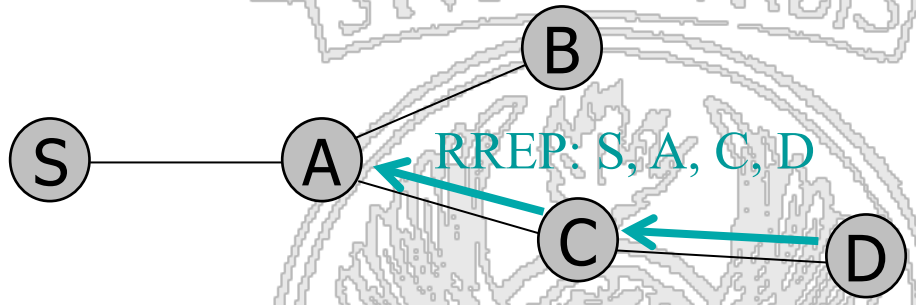
4. Node *C* receives RREQ, has no route to *D*
   - Rebroadcasts packet after adding its address to source route
5. Node *D* receives RREQ, unicasts RREP to *C*
   - Puts source route accumulated in RREQ into RREP

DSR: Route Discovery



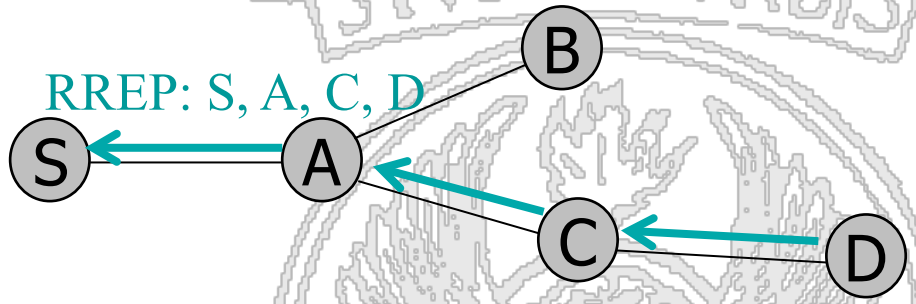
4. Node *C* receives RREQ, has no route to *D*
   - Rebroadcasts packet after adding its address to source route
5. Node *D* receives RREQ, unicasts RREP to *C*
   - Puts source route accumulated in RREQ into RREP

DSR: Route Discovery



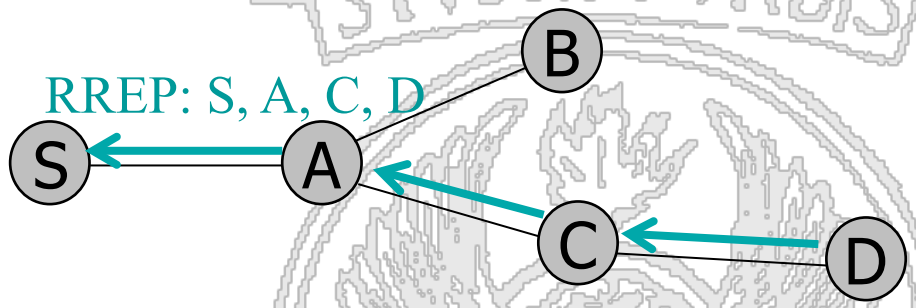
6. Node *C* receives RREP
   - Unicasts to *A*

DSR: Route Discovery



6. Node *A* receives RREP
   - Unicasts to *S*

DSR: Route Discovery



8. Node *S* receives RREP
   - Uses route for data packet transmissions

SAPIENZA
UNIVERSITÀ DI ROMA

General node operation upon receiving RREQ

- If the pair <initiator address, request ID> has recently been seen, DISCARD
- If the node ID is already listed in the source route DISCARD→ avoids loops
- If I'm the destination, send a RREP
- Otrherwise, append my ID in the source route and rebroadcast (orange cases already seen in the previous slides)

SAPIENZA
UNIVERSITÀ DI ROMA

Route maintenance

- The two endpoints of a failed link are transmitted to the source in a route error packet
- Upon a receiving a RERR packet a node invalidates all the routes going through that link
- If the route is invalidated and it is needed, a new route must be discovered

SAPIENZA
UNIVERSITÀ DI ROMA

Optimizations (1)

- Extensive use of caching (caching source routes means that I already know all the route to intermediate destinations, discovery a better route to an intermediate destination also brings me to improve the route to the final destination). Transmitting packets or sending back replies make me learn routes.
- A node that knows a route to a given destination (has a source route in cache) can immediately answer a RREQ
  - Broadcast storm? Each nodes waits for a time which is C*(h-1+r), r random in (0,1), h length of the route I'm advertising. Only if I haven't received other routes –listen to other routes tx in the meanwhile-I transmit mine.

SAPIENZA
UNIVERSITÀ DI ROMA

Optimization (2)

- Operation in promiscuous mode (I keep discovering new routes by transmission of routes by my neighbours)
- RREQ overhead minimization: first set a TTL=1, if I do not get answer I set it to infinity
- Path shortening: if Y receives a packet by node X but in the source route we have X, B,…,C,Y, Y signals the path can be shortened (unsolicited RREP)
- What if the network is disconnected? Exponential back-off to decrease the quantity of RREQ sent

SAPIENZA
UNIVERSITÀ DI ROMA

AODV and DSR Differences

- DSR uses source routing; AODV uses next hop entry
- DSR uses route cache; AODV uses route table
- DSR route cache entries do not have lifetimes;
  AODV route table entries do have lifetimes

Proactive Solutions: Drawbacks

- Updates overhead, especially in presence of high mobility
- Overhead for enforcing loop freedom
- Large routing tables
- Low *scalability*

➢ Is it really necessary to maintain a consistent view of the network topology?

SAPIENZA
UNIVERSITÀ DI ROMA

Reactive Protocols: Drawbacks

- The discovery phase introduces long delays
- Route discovery and maintenance is very sensitive to node mobility
- Route caching is memory greedy
- The size of the header of a data packet can become cumbersome in approaches such as DSR (no scalability)
- Operating in promiscuous mode is energy-consuming.
- Relying on flooding based route discovery is resource consuming.

➢ Is the dependency on the network topology avoidable?

SAPIENZA
UNIVERSITÀ DI ROMA

Geographically-Enabled Routing

- Outline
  - Problems with proactive approaches
  - Problems with reactive approaches
  - A new way of naming\locating the destination node: geographic routing
  - Two seminal protocols
    - ✓ DREAM & LAR
  - Geo-enable routing costs: I need to know where I am, where the destination is.

SAPIENZA
UNIVERSITÀ DI ROMA

Location-Enabled Ad Hoc Routing

- Nodes are equipped with positioning system devices (e.g., Global Positioning System receivers) that make them aware of their position
- This enables "directional" routing
- Possible solutions differ on how the location information of the destination nodes is achieved

SAPIENZA
UNIVERSITÀ DI ROMA

Strengths

- No need to update big routing tables, no need to piggyback routes to the packet
  - Destination position must be known at the source.
- No need to know the nodes on the way to the destination: they can be moving while the packet travels

SAPIENZA
UNIVERSITÀ DI ROMA

Drawbacks

- Needs extra hardware
- Depends on the extra hardware limitation (and resource requirements)
- Scalability is an issue (indeed the problem here translates to how to maintain correct estimates of the nodes positions)

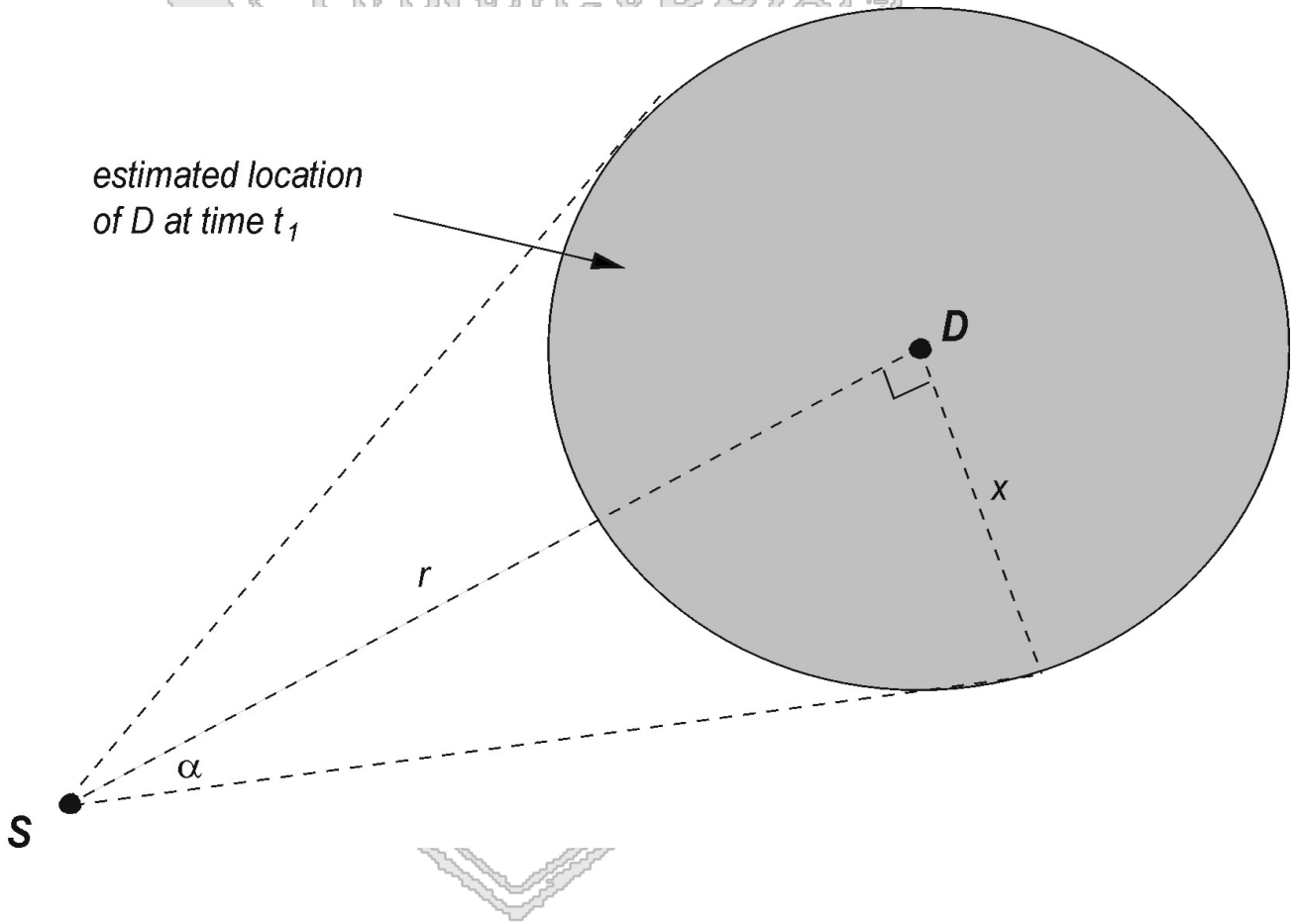
DREAM

- Distance routing effect algorithm for mobility [Basagni+, 1998]
- A proactive, effective way to spread location information
- Directional routing

DREAM: Directional Routing

- Source S determines the location of destination D at time t_0 based on its location table
- Based on the current time t_1 and t_0 S determines the area in which D can be found (hence, D's direction)
- S transmits the data packet to all its neighbors in D's direction
- Each neighbor does the same till D is reached

DREAM: Routing a Data Packet



SAPIENZA
UNIVERSITÀ DI ROMA

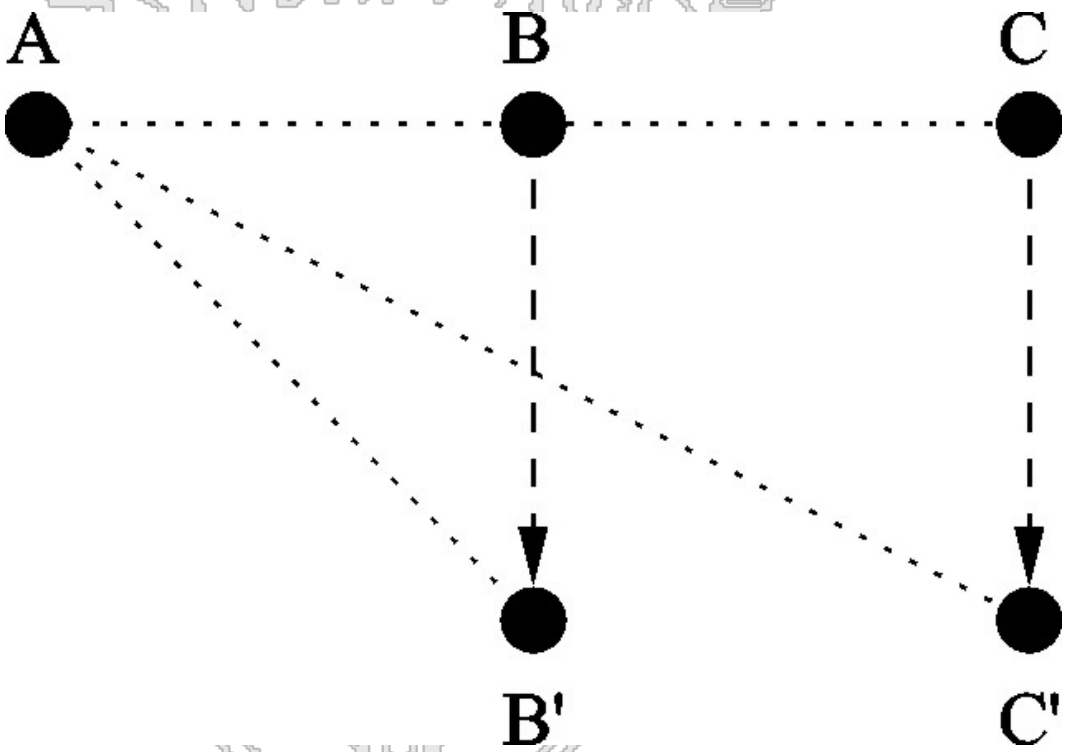
Disseminating Location Information: Problems

- Need to periodically update the location of a moving node.
  - Efficient broadcast of location information
  - Determining how far each location packet should travel
  - Determining how often a location packet should be sent

Disseminating Location Information: Solutions

- Distance effect
- Rate of updates is bound to the mobility of the node

The Distance Effect



SAPIENZA
UNIVERSITÀ DI ROMA

The Distance Effect

- "Closer nodes look like they are moving faster"
- Need to receive more location updates from closer node
- Each location update packet is associated with an age that determines how far that packet must travel

SAPIENZA
UNIVERSITÀ DI ROMA

DREAM: Rate of updates

- Triggered by the mobility of the nodes
- The faster the node the more updates it sends
- A plus: slow moving nodes impose little overhead

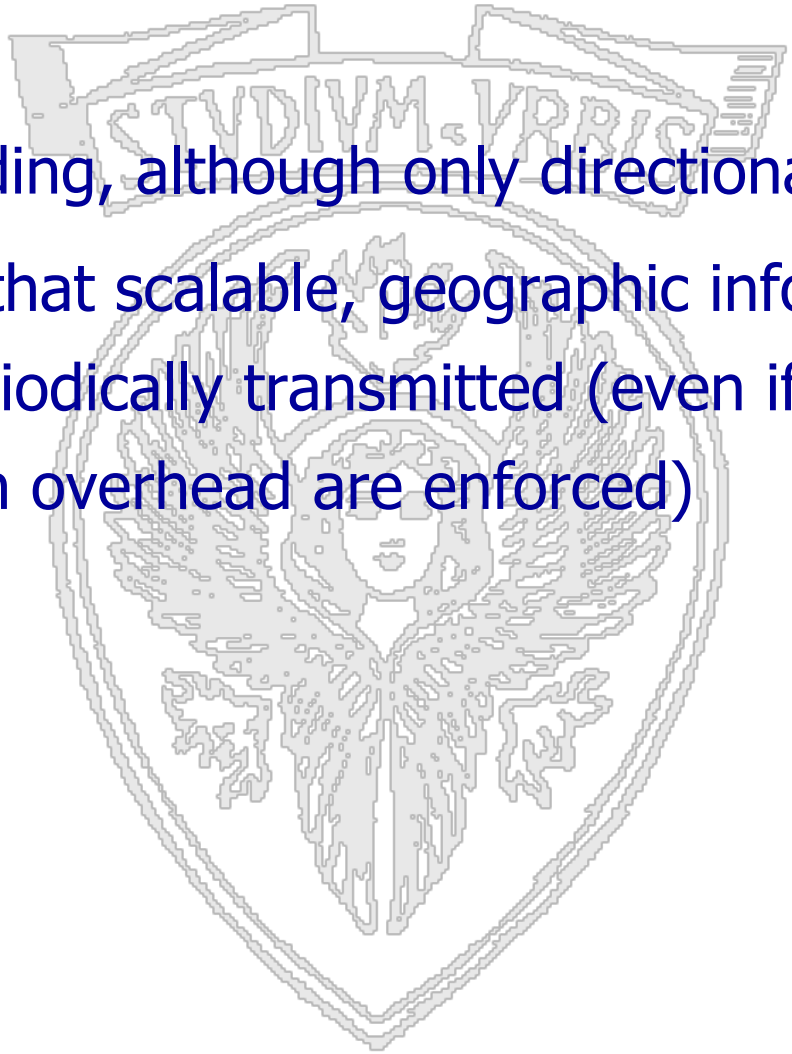
SAPIENZA
UNIVERSITÀ DI ROMA

DREAM, Strengths

- First of its kind: after that, the deluge!
- Robustness: multiple routes to the destination
  - directional flooding

SAPIENZA
UNIVERSITÀ DI ROMA

DREAM, Weaknesses



- It is flooding, although only directional
- It is not that scalable, geographic info updates have to be periodically transmitted (even if mechanisms to limit such overhead are enforced)

SAPIENZA
UNIVERSITÀ DI ROMA

Location-Aided Routing (LAR)

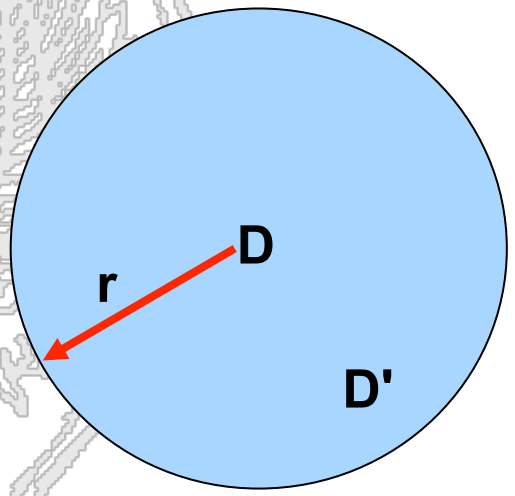
- Exploits location information to limit scope of RREQ flood
- *Expected Zone*: region that is expected to hold the current location of the destination
  - Expected region determined based on potentially old location information, and knowledge of the destination's speed
- RREQs limited to a *Request Zone* that contains the Expected Zone and location of the sender node

LAR: Expected Zone

D = last known location of node D, at time t_0

D' = location of node D at current time t_1, unknown to node S

r = (t_1 – t_0) * estimate of D's speed



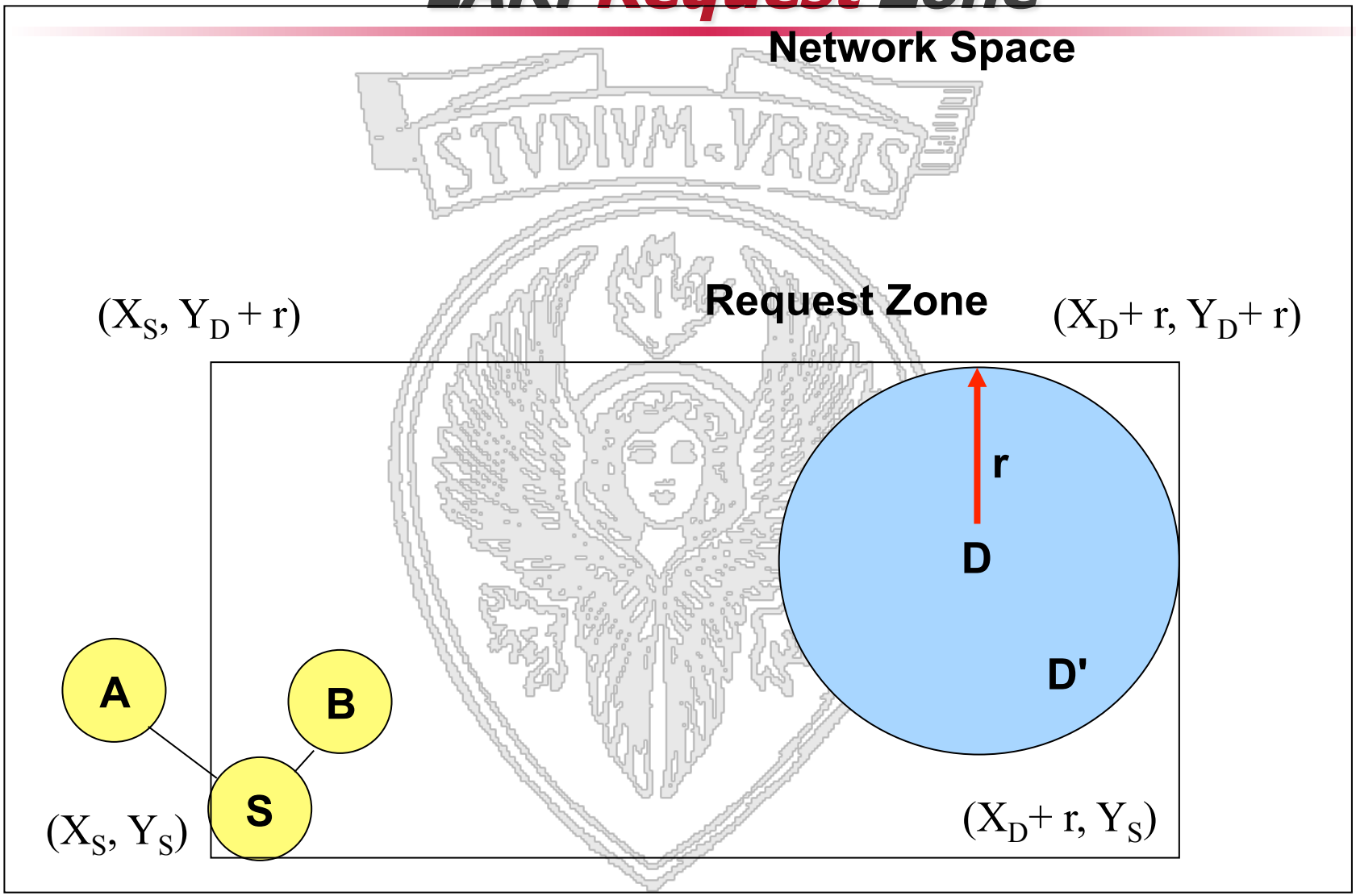
Expected Zone

SAPIENZA
UNIVERSITÀ DI ROMA

LAR

- The request zone is the smallest rectangle that includes the current location of the source and the expected zone
- Only nodes within the request zone forward route requests
  - Node A does not forward RREQ, but node B does
- Request zone explicitly specified in the RREQ
- Each node must know its physical location to determine whether it is within the request zone

LAR: Request Zone



SAPIENZA
UNIVERSITÀ DI ROMA

LAR, Possible Failures

- If route discovery using the smaller request zone fails to find a route, the sender initiates another route discovery (after a timeout) using a larger request zone
  - The larger request zone may be the entire network
- Rest of route discovery protocol similar to DSR

SAPIENZA
UNIVERSITÀ DI ROMA

LAR, the Routing

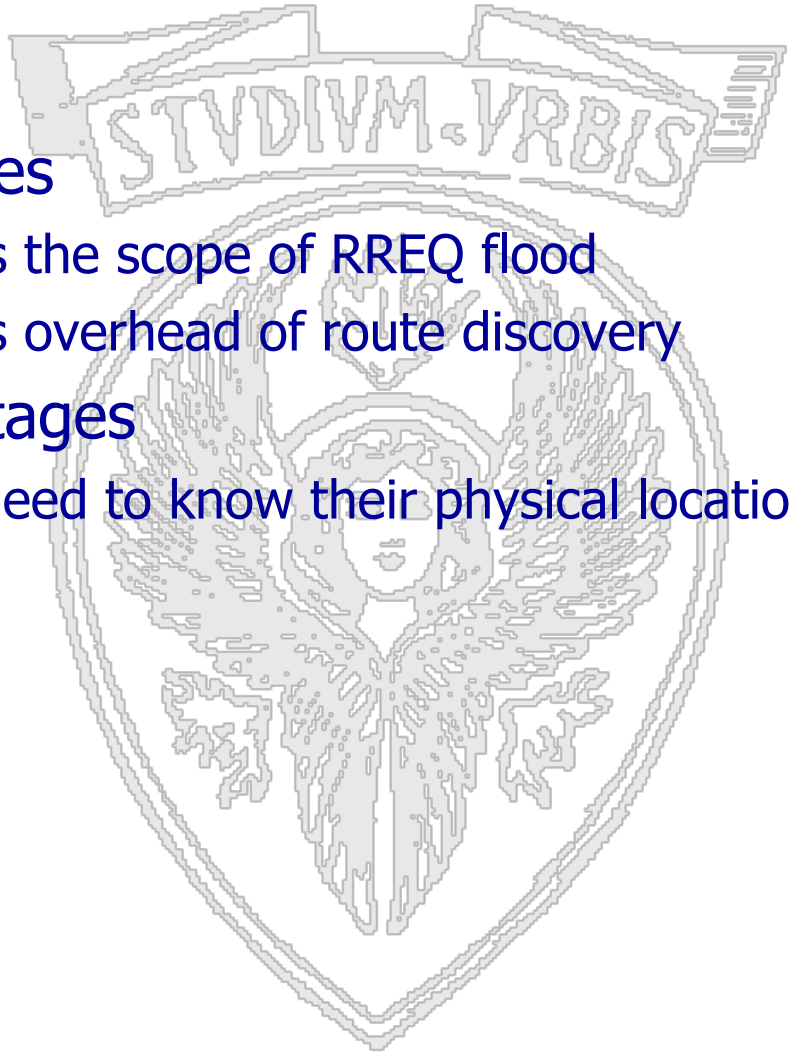
- The basic proposal assumes that, *initially*, location information for node X becomes known to Y only during a route discovery
- This location information is used for a future route discovery

Variations

- Location information can also be piggybacked on any message from Y to X
- Y may also proactively distribute its location information

SAPIENZA
UNIVERSITÀ DI ROMA

LAR, Pros and Cons



- Advantages
  - Reduces the scope of RREQ flood
  - Reduces overhead of route discovery
- Disadvantages
  - Nodes need to know their physical locations


STVDIVM-VRBIS

- For solutions which scale
- Which are energy saving
- Which are well integrated with awake/asleep schedules
- Which do not require to maintain routing tables
- Which are simple
- Solutions such as AODV and DSR have been proven to work well iff

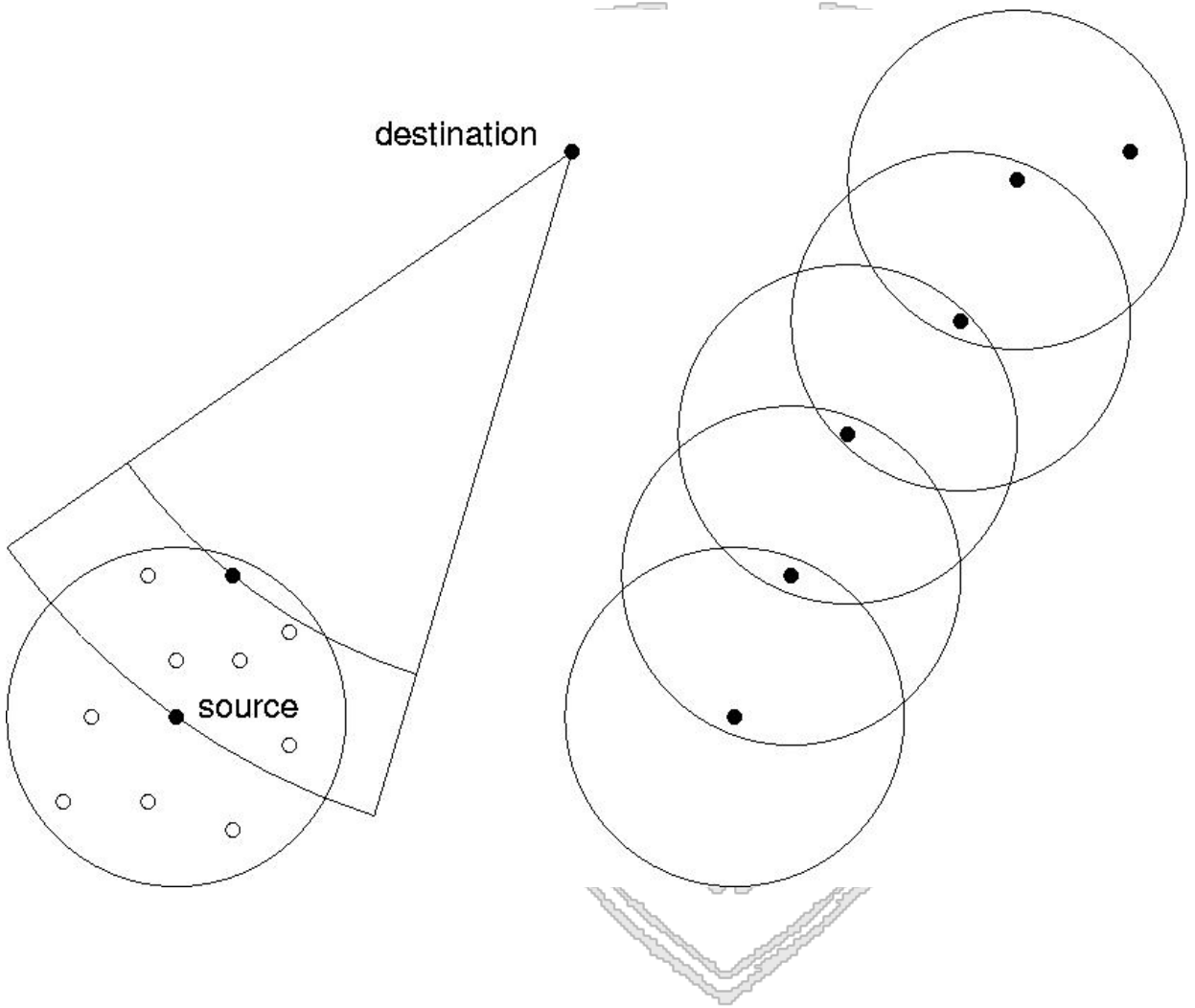
they exploit intensively caching and promiscuous mode operation (energy inefficient← work by L. Feeney et al, 2001) and have been shown not to scale to the volumes of nodes expected in sensor networks (work by E. Belding Royer and S.J. Lee)

- What can we use?
  - communication sensors – sink
  - Info such as localization and some level of synchronization often needed by the application (if I sense an event I have to say WHERE and WHEN it occurred, otherwise the information is not very interesting)

An example: GeRaF

- Integrates
  - geographic routing
  - awake/asleep schedule
  - MAC
- How do nodes alternate between awake and asleep states? According to a duty cycle (time ON/time ON+ OFF)

GeRaF basic idea



Geographic routing: each node needs to know its location, the destination (sink) location, and the location of whom is transmitting (communicated in the packet)

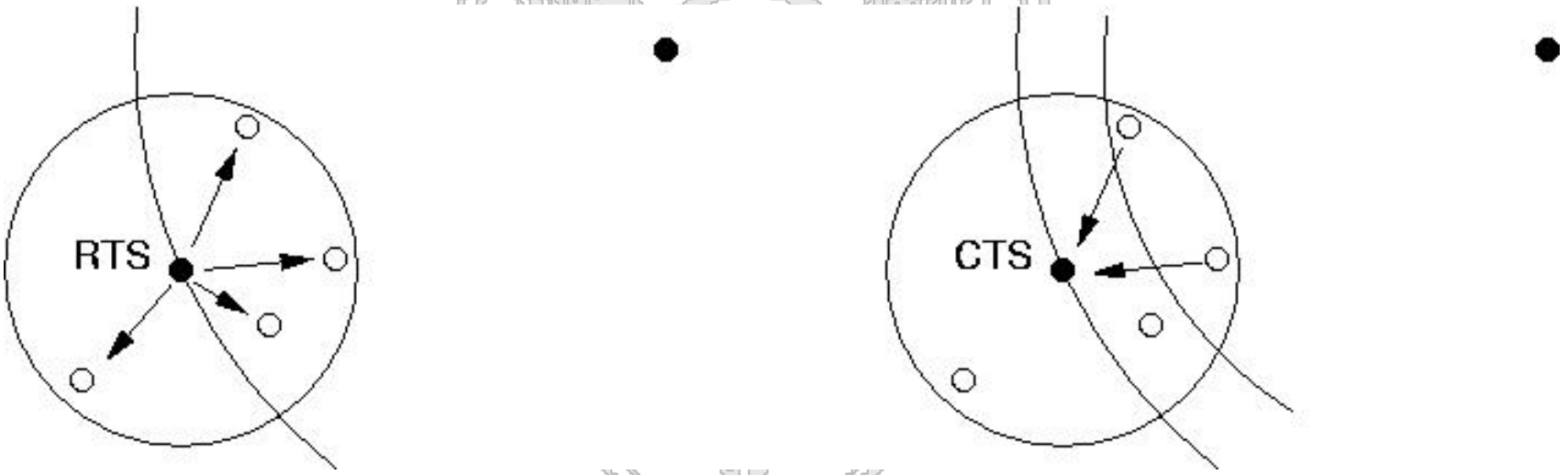
Greedy approach: tries to select relays so to advance as much as possible toward the destination at each hop

GeRaF: operations

- Main problem to be solved: how to make a contention-based scheme/routing work in the presence of sleep modes
  - Flat solution
  - Integrated MAC/routing/awake-asleep but awake-asleep schedule and routing decoupled → each node does not know its neighbor and their schedules → low overhead
- Tightly integrated with the routing layer (no clear separation really)
  - Without requiring routing tables/routing table updates
  - Based on the knowledge of the nodes location and on the knowledge of the sink location

Example of GeRaF operation

- RTS invites all awake neighbors to become a relay
- Nodes in best position should win
  - Nodes within tx range are divided in areas depending on how close they are to the final destination (the closest the better as relay)



•Need of location awareness

GeraF operation

- Node i sends RTS with the identity of the area it is polling now (starting from the closer to the sink, among the slices in which its tx range has been divided)
- Each node, upon receiving the RTS, decides whether it belongs to the polled area or not (based on location info)
- Only nodes in the polled area answer with a CTS
  - No node answers → node i polls next area (no node available for forwarding in the area-there are no nodes or they are sleeping)
  - One answer, CTS correctly received, send DATA
  - Multiple answer COLLISION, sender sends a collision packet, MAC needed to solve collision (next slide)

SAPIENZA
UNIVERSITÀ DI ROMA

GeraF operation (how to handle collision)

- 1)A node receiving a collision packet tosses a coin and with probability p transmits a CTS iff it was participating to the previous phase (it had previously sent a CTS resulting in collision)
  - if only one node answers node i sends data
  - If no node answers node i asks these nodes to toss a coin again..
  - if more nodes answer COLLISION. Collision packet is sent. GO TO 1) (only the nodes which have lead to collision survive to the next phase)

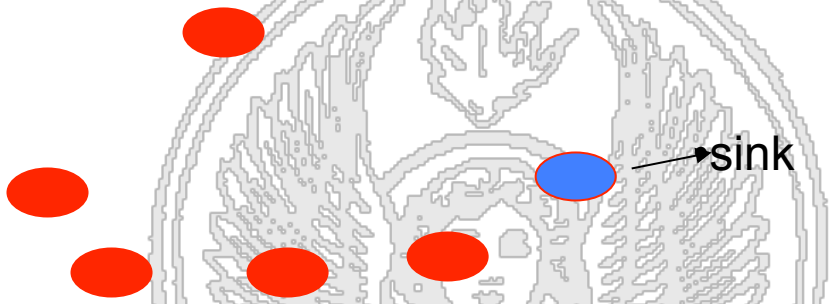
What If

- All areas are polled unsuccessfully?
  - Try again after some time (exponential backoff)
- Can I always reach the destination in this way?

What if (answer)

- No. Here is

An example

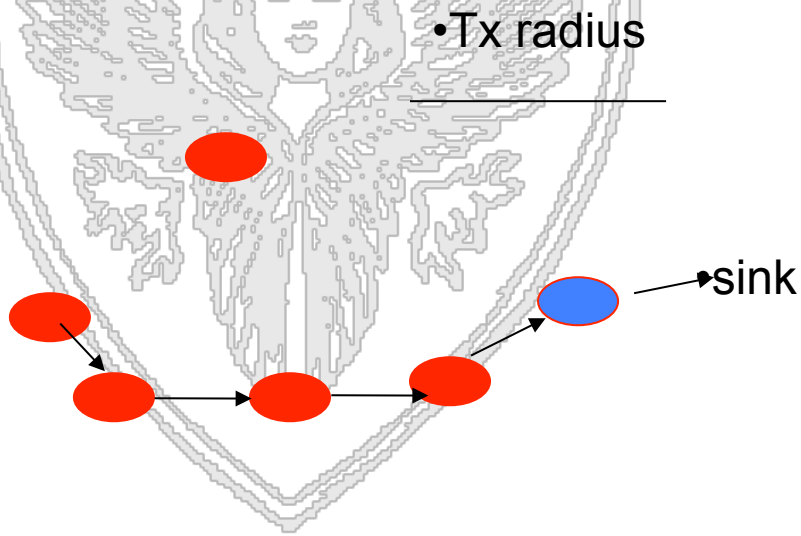


- Solutions? (mechanisms have to be added to recognize the problem, do backtracking and try another route)

SAPIENZA
UNIVERSITÀ DI ROMA

How to solve dead-ends

- A problem only at low density
- We can set a maximum number of attempts to find a relay. When a node fails to find a relay it starts decreasing its duty cycle/or the probability to propose itself as relay …over time nodes along paths to dead-ends are less and less selected as next hop relays and other paths able to bring to the destination are instead found
- Still..we may have problems…



SAPIENZA
UNIVERSITÀ DI ROMA

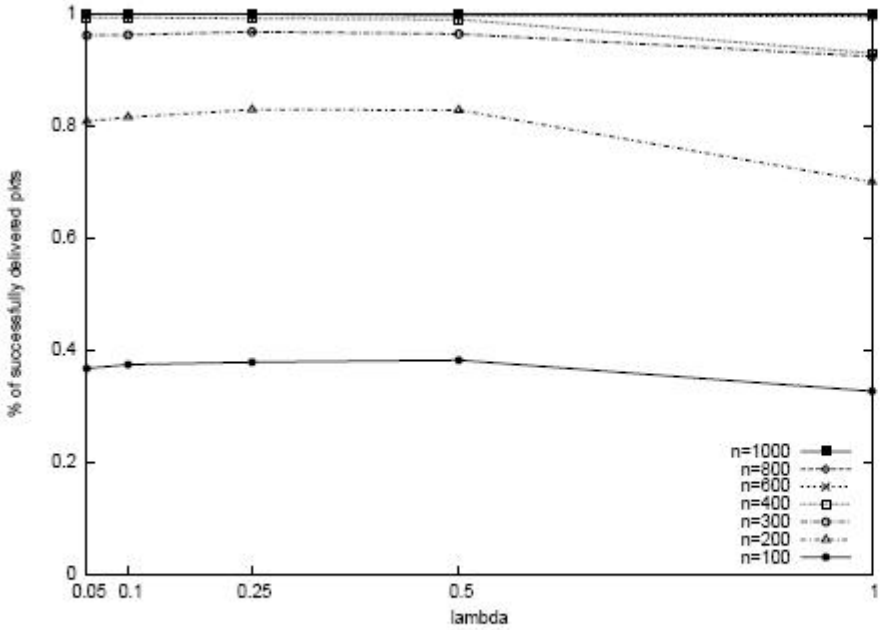
GeRaf performance

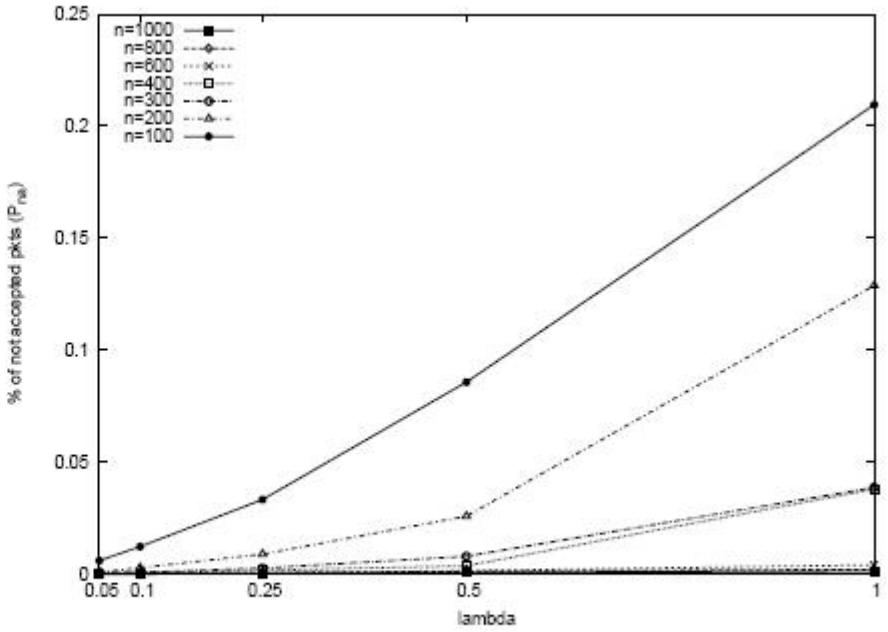
Casari, Marcucci,Nati, Petrioli, Zorzi IEEE MILCOM 2005

- square area 320m x 320m
- Transmission range=40m
- 100-1000 randomly deployed nodes (avg degree 5-50)
- Duty cycle =0.01,0.1,0.5
- Comparable costs for tx/rx/idle
- Poisson packet arrival
- Channel data rate 38Kbps

GeRaf performance, d=0.1

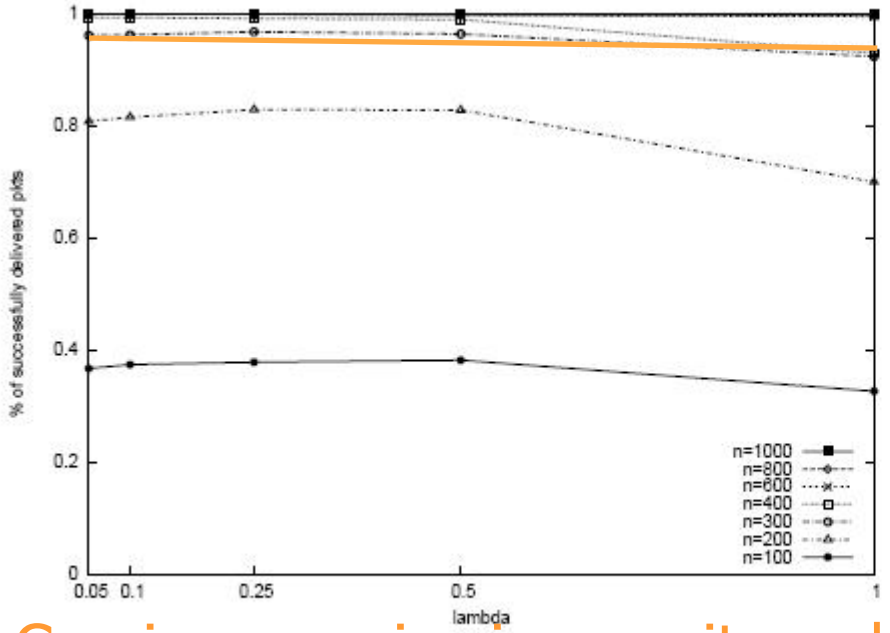
Casari, Marcucci, Nati, Petrioli, Zorzi IEEE MILCOM 2005

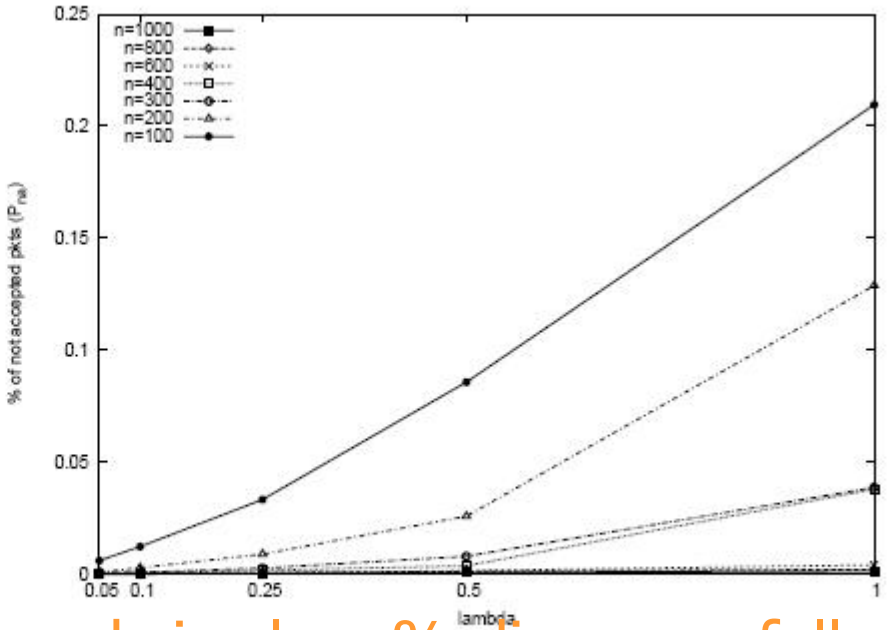




GeRaf performance, d=0.1

Casari, Marcucci,Nati, Petrioli, Zorzi IEEE MILCOM 2005





Con i meccanismi per evitare dead end si sale a % di successfully Delivered packets nel caso di 200 nodi pari a 93-97% (evitando Nel tempo cammini che portino a dead ends). Non si risolve il Problema nel caso n=100. Soluzione completa in uno schema che abbiamo proposto: ALBA.


STVDIVM-VRBIS

SAPIENZA
UNIVERSITÀ DI ROMA

Localization in sensor networks

Thanks to Prof. Mani Srivastava
These slides have been derived
From his tutorial on sensor networks
Given at Rome Un. On July 2003

SAPIENZA
UNIVERSITÀ DI ROMA

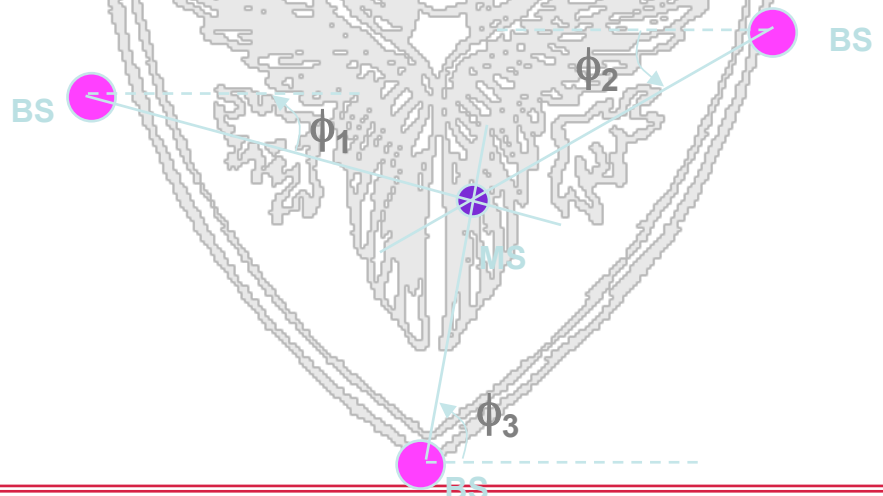
Localization

- Useful info
  - Helps with some protocols (e.g. GeraF)
  - Needed for being able to identify where events occur
- Why not just GPS (Global Positioning System) at every node?
  - Large size
  - High power consumption
  - Works only when LOS to satellites (not in indoor, heavy foliage…)
  - Over kill – often only relative position is needed (e.g. enough to know that relative to a coordinate system centered in the sink the event occurred in a position (x,y). Starting from relative info if some nodes have global coordinates global coordinates of events can be inferred.

Localization

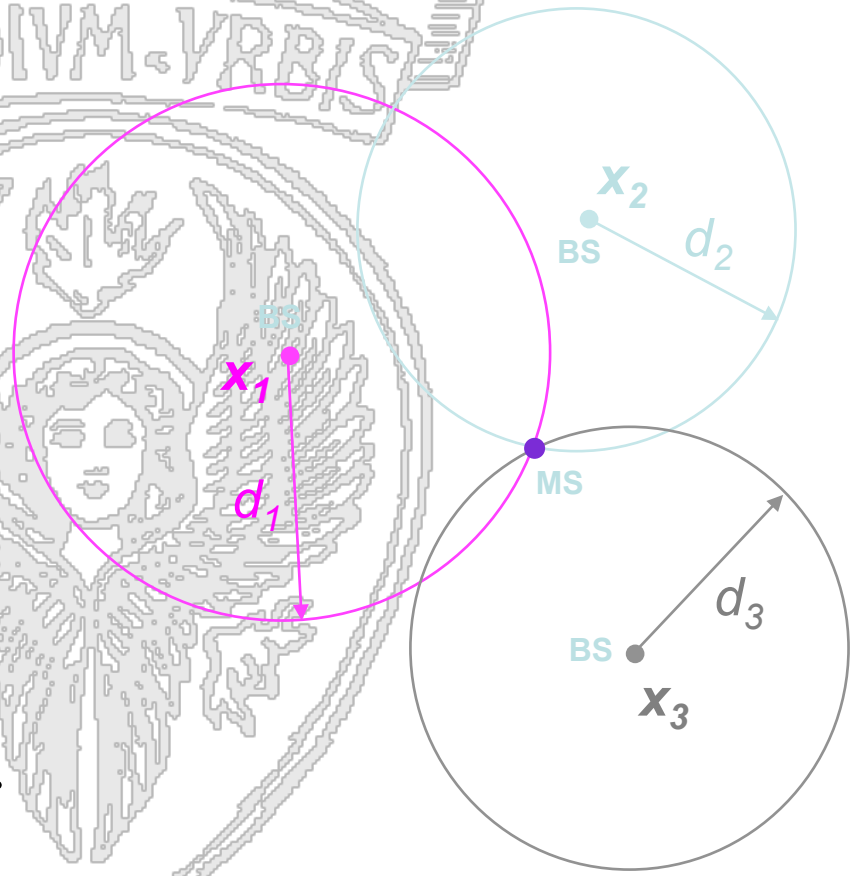
- Basic step is to evaluate distance between two nodes (ranging). Different techniques depending on the available HW:
  - AoA (e.g. directional antennas)
  - RSS
  - ToA
- Range free approaches (number of hops between nodes used to estimate the distance between them without using any extra HW)

- Measure direction of landmarks
  - Simple geometric relationships can be used to determine the location by finding the intersections of the lines-of-position
  - e.g. Radiolocation based on angle of arrival (AoA) measurements of beacon nodes (e.g. base stations)
    - ✓ can be done using directive antennas + a compass
    - ✓ **need at least two measurements**



Techniques for Location Sensing (RSS or ToA)

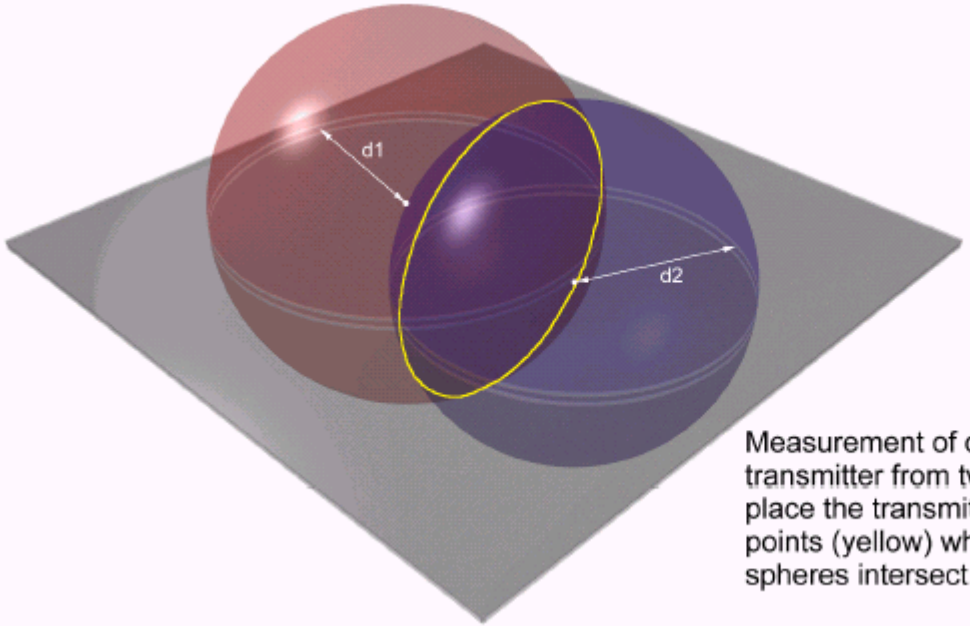
- Measure distance to landmarks, or Ranging
  - e.g. Radiolocation using signal-strength or time-of-flight
    - ✓ also done with optical and acoustic signals
  - Distance via received signal strength
    - ✓ use a mathematical model that describes the path loss attenuation with distance
      - each measurement gives a circle on which the MS must lie
    - ✓ use pre-measured signal strength contours around fixed basestation (beacon) nodes
      - can combat shadowing
      - location obtained by overlaying contours for each BS
  - Distance via Time-of-arrival (ToA)
    - ✓ distance measured by the propagation time
      - distance = time * c
    - ✓ each measurement gives a circle on which the MS must lie
    - ✓ active vs. passive
      - active: receiver sends a signal that is bounced back so that the receiver knows the round-trip time
      - passive: receiver and transmitter are separate
        - » time of signal transmission needs to be known
  - N+1 BSs give N+1 distance measurements to locate in N dimensions



•But what if the circles do not intersect due to measurement errors (e.g. due to fading etc.)?

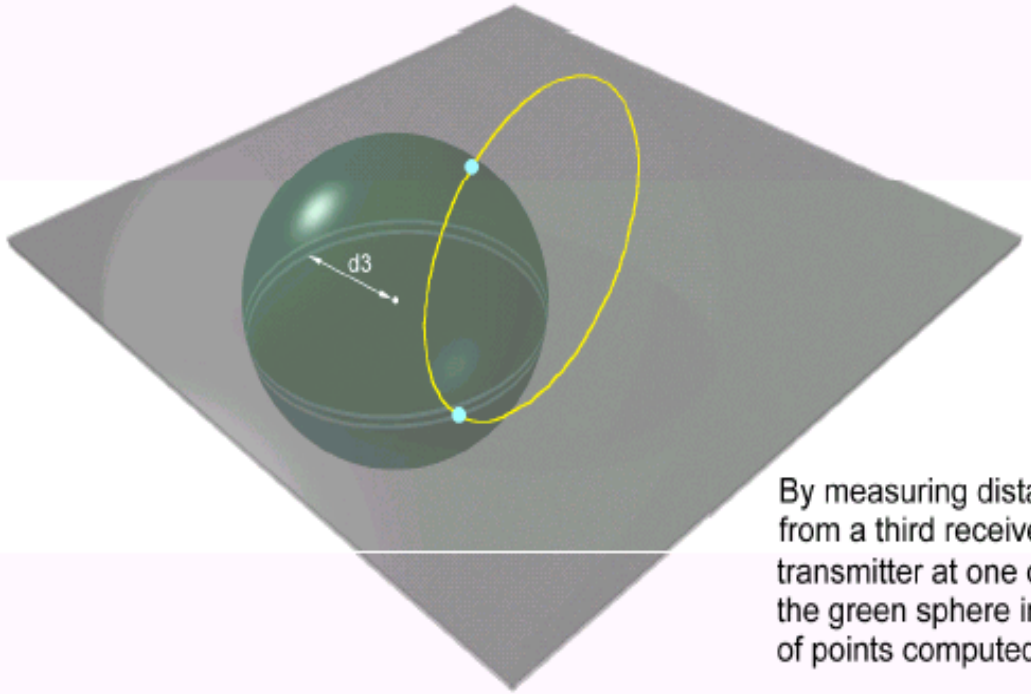
→will have to identify the best 'guess' given errors

Location in 3D



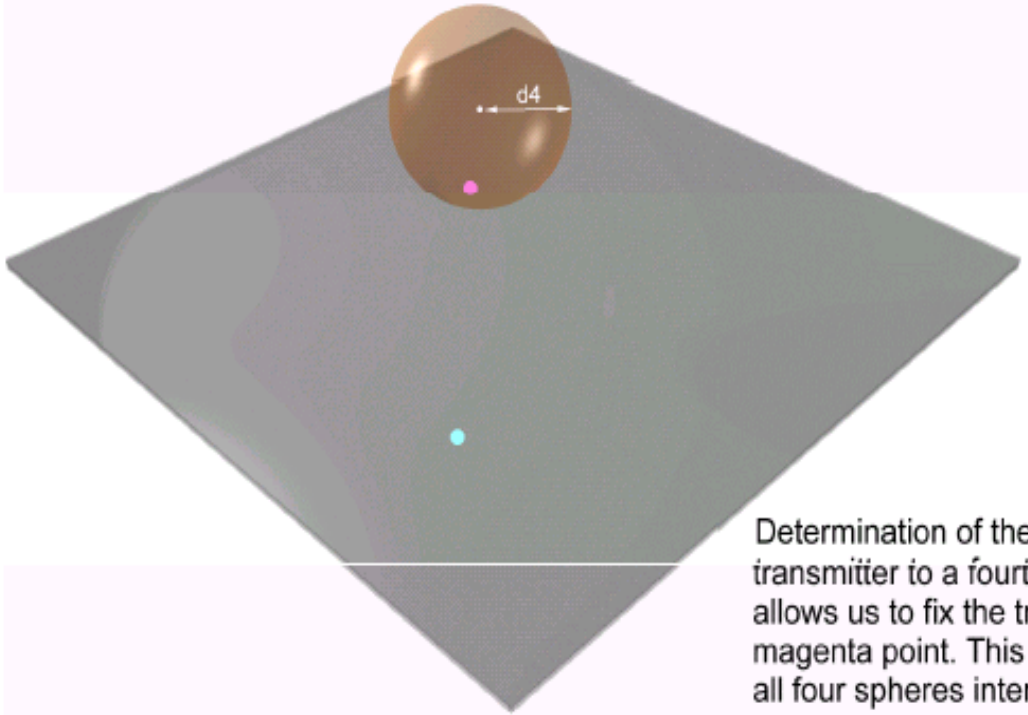
Measurement of distances d1 & d2 to the transmitter from two receivers allows us to place the transmitter on a circular locus of points (yellow) where the red and blue spheres intersect.

Location in 3D



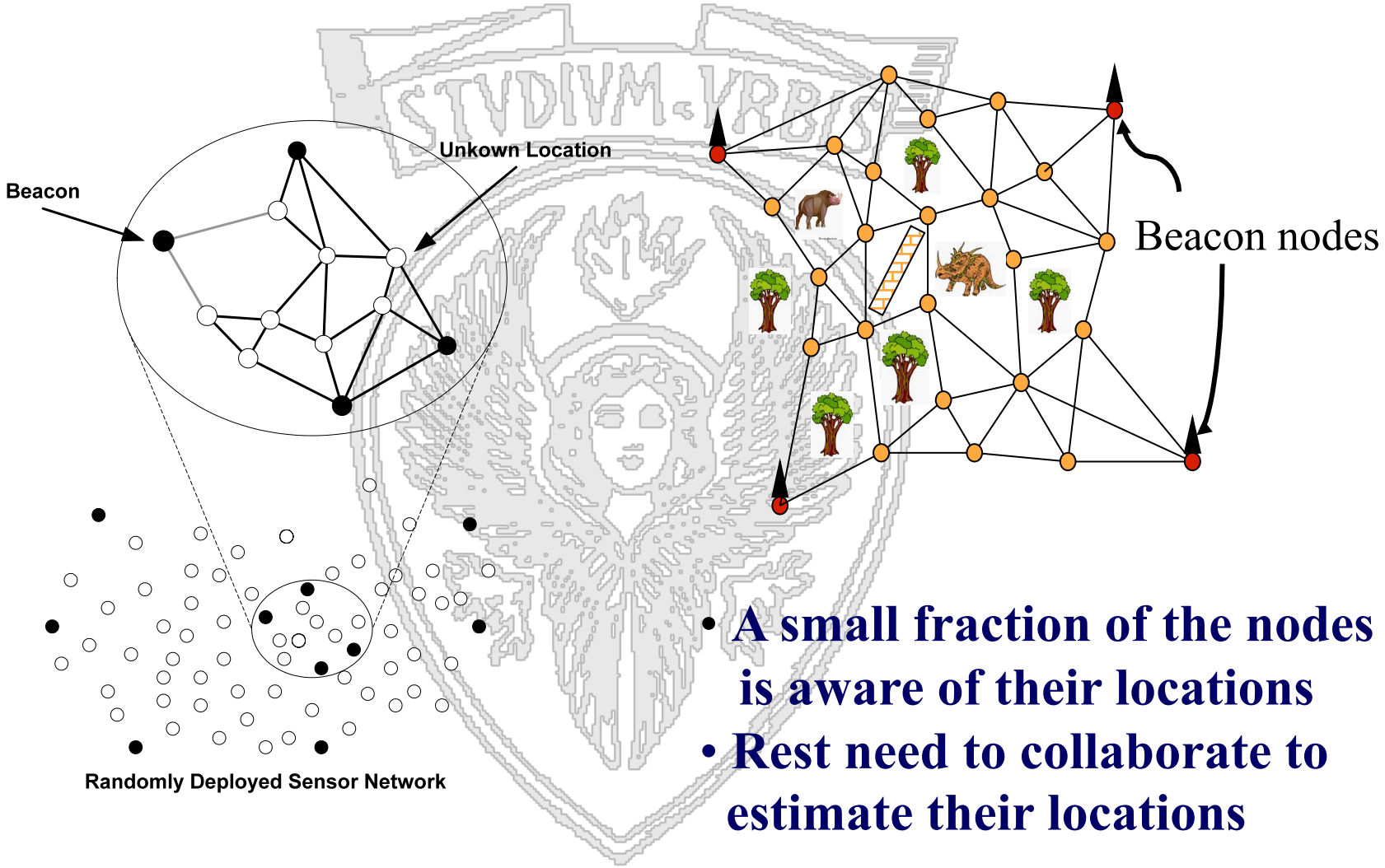
By measuring distance d3 to the transmitter from a third receiver, we can place the transmitter at one of two points (cyan) where the green sphere intersects the circular locus of points computed previously.

Location in 3D



Determination of the distance d4 from the transmitter to a fourth, non-copianar receiver allows us to fix the transmitter's location at the magenta point. This is the single point at which all four spheres intersect.

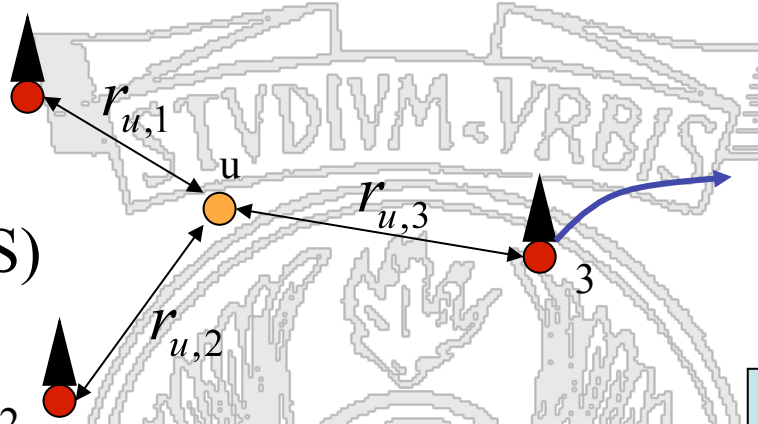
A possible solution: Absolute Localization



- A small fraction of the nodes is aware of their locations
- Rest need to collaborate to estimate their locations

Atomic Multilateration

Nodi che hanno almeno 3 vicini$_1$ (in 2D, se si usa Ad esempio RSS) beacon possono stimare$_2$ la loro posizione (triangolarization)



Beacon node with known location

In presenza di errori
Metrica di interesse
Errore quadratico medio

$$f_{u,i} = r_{u,i} - \sqrt{(x_i - \hat{x}_u)^2 + (y_i - \hat{y}_u)^2}$$

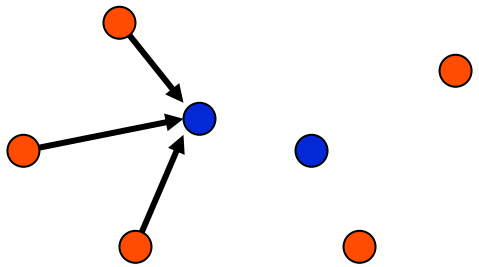
$(\hat{x}_u, \hat{y}_u)$ - initial position estimate for node u

Our objective function is:

$$F(x_u, y_u) = \min \sum f_{u,i}^2$$

Iterative Multilateration

- Nodes that estimate their locations can become beacons and help other nodes discover their locations.



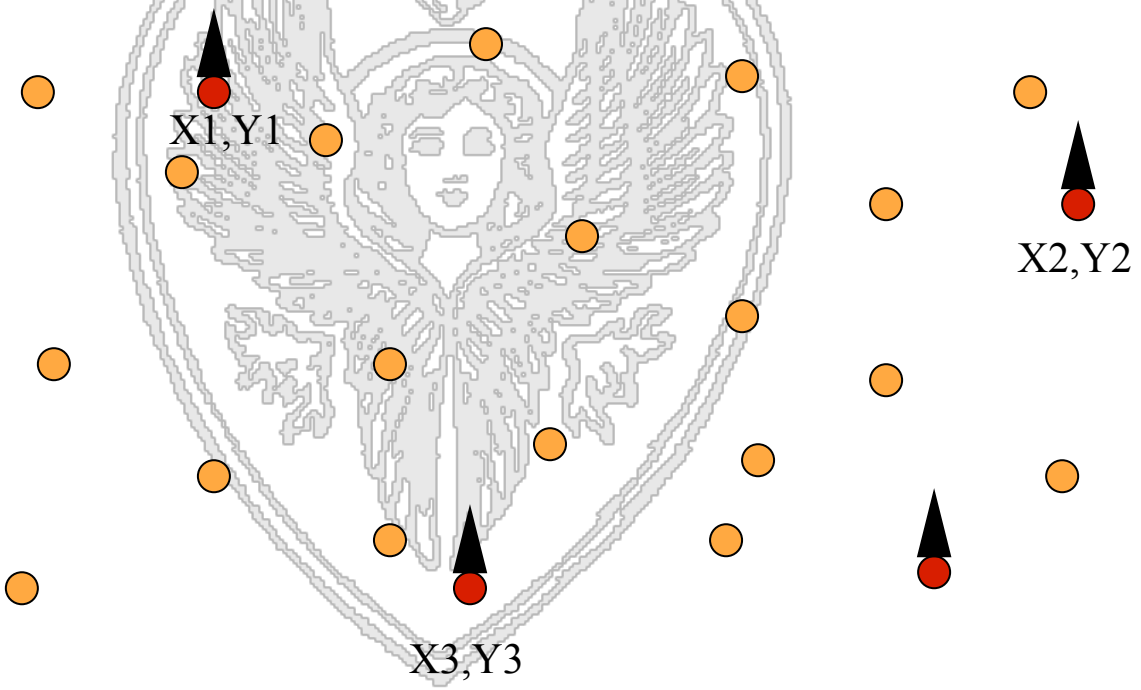
- Some observations:
  - Can work for small networks, if ranging is accurate
  - Energy efficient
  - Still requires quite a lot of initial beacons
  - Suffers from error accumulation
  - **Bad geometry yields bad results => unpredictable performance**
  - Still a useful primitive for Distributed Collaborative Multilateration **Primo approccio semplice, vasta letteratura**

Range free localization

- Non usa ranging, ma solo informazioni che si possono ottenere tramite algo di routing tradizionali
- Idee su come possa funzionare?

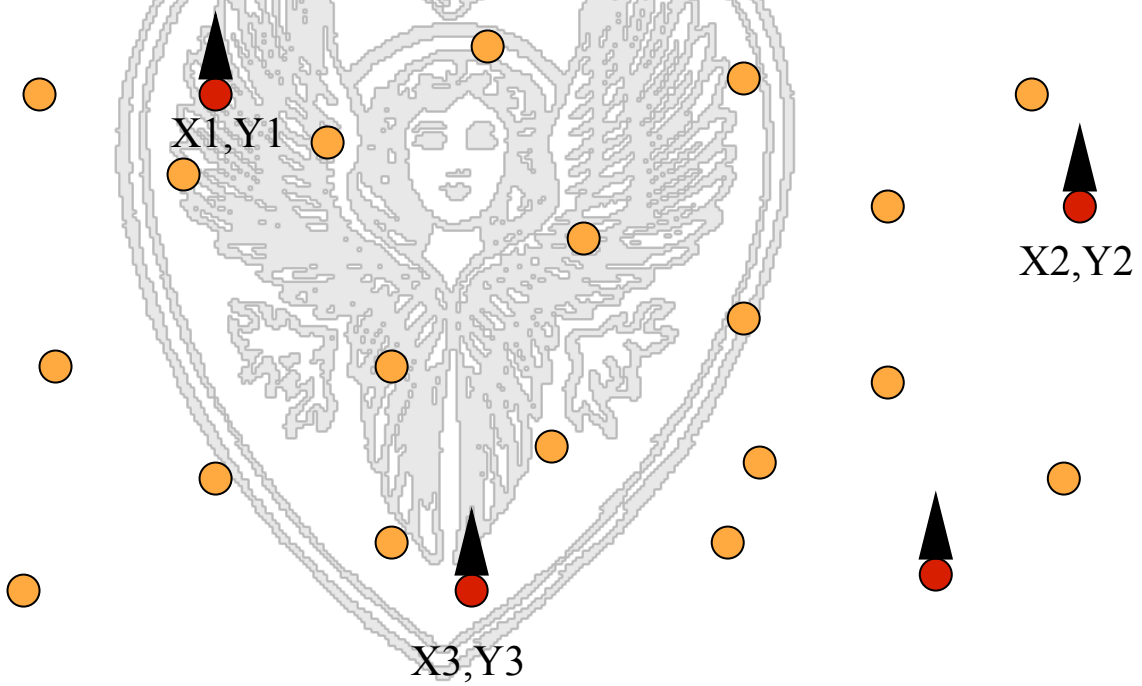
Qualche idea sull'approccio ...

- Servono degli anchor, nodi che conoscono la propria posizione in un sistema di coordinate comune



Qualche idea sull'approccio …

- Tutti i nodi calcolano il numero min. di hop tra loro e gli anchor
- Anche gli anchor lo fanno tra loro



Qualche idea sull'approccio …

- Anchor A: conosco la posizione esatta mia e degli altri anchor, il num. di hop, posso stimare la 'lunghezza media di un hop'
- Questa informazione e' usata per stimare le distanze da tutti i nodi agli anchor. Sulla base di tali distanze, le corrette coordinate degli anchor, per triangolarizzazione ciascun nodo stima le proprie coordinate

- Pro: Non serve extra HW
- Cons: si perde in precisione