SAPIENZA
UNIVERSITÀ DI ROMA

Reti ad hoc

Localizzazione e clustering

Sistemi Wireless, a.a 2011/2012

Un. of Rome "La Sapienza"

Chiara Petrioli[†]

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

Thanks to Prof. Mani Srivastava
These slides have been derived
From his tutorial on sensor networks

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.

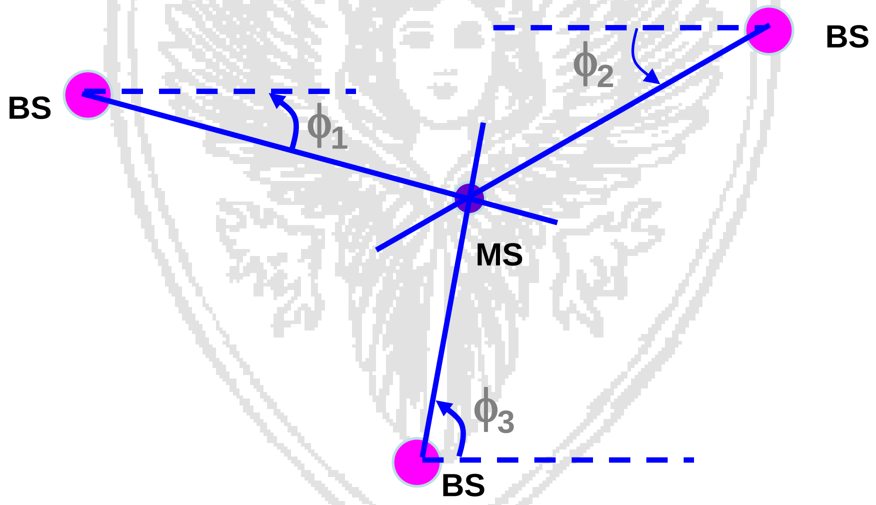
Localization

- 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)

Angle of Arrival

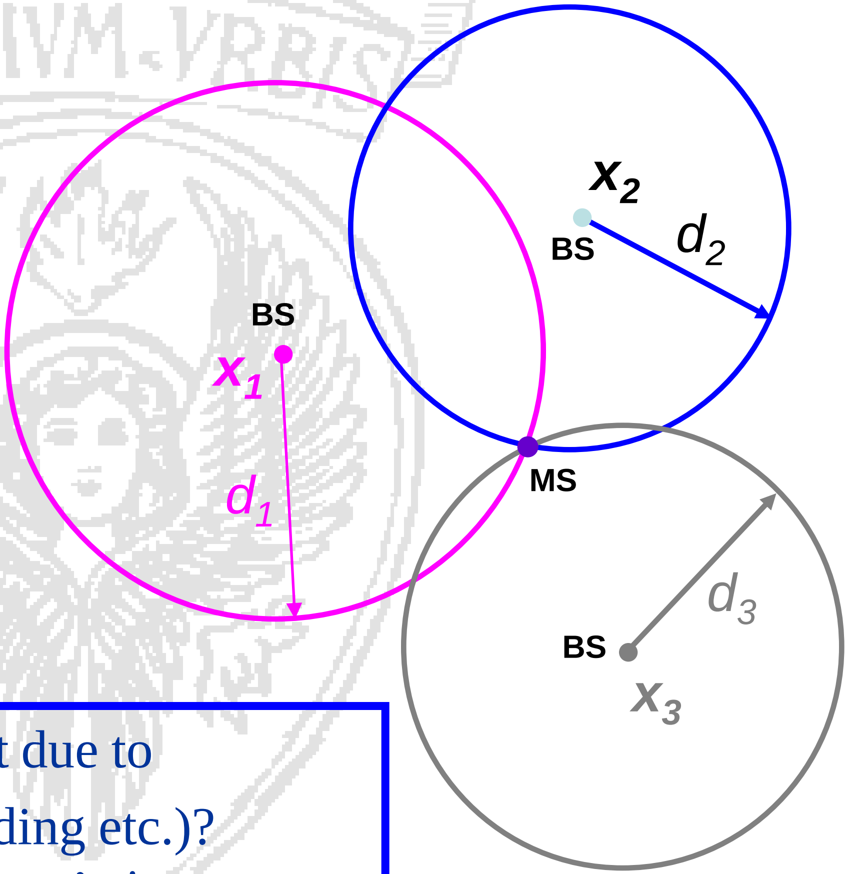
- Measure direction of landmarks
  - Simple geometric relationships can be used to determine the location by finding the intersections of the lines-of-position
  - e.g. Radiolocation based on angle of arrival (AoA) measurements of beacon nodes (e.g. base stations)
    - ✓ can be done using directive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



Ranging techniques (RSS or ToA)

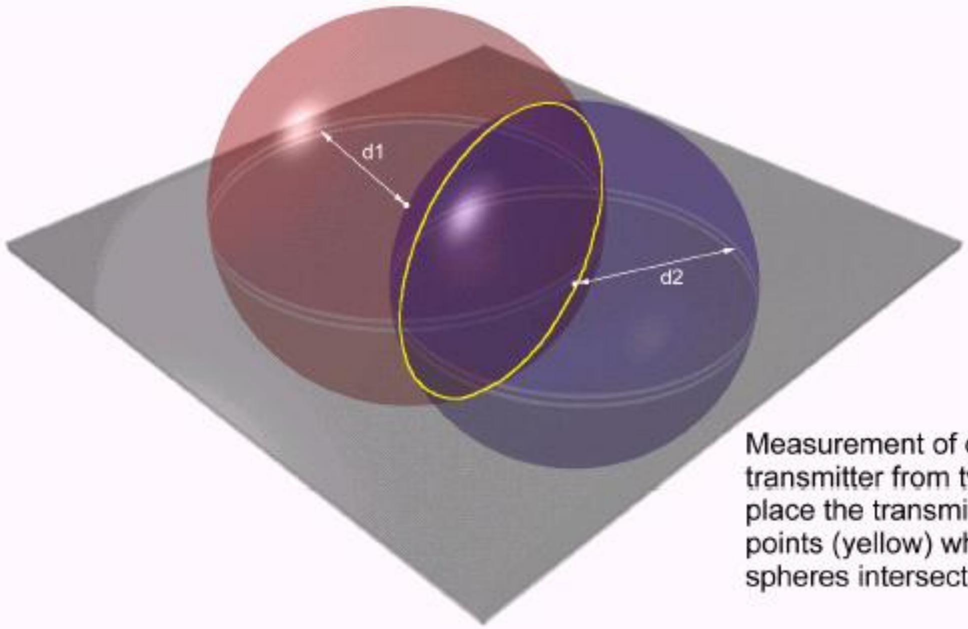
- Measure distance to landmarks, or Ranging
  - e.g. Radiolocation using signal-strength or time-of-flight
    - ✓ also done with optical and acoustic signals
  - Distance via received signal strength
    - ✓ use a mathematical model that describes the path loss attenuation with distance
      - each 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

Radiolocation via ToA and RSSI



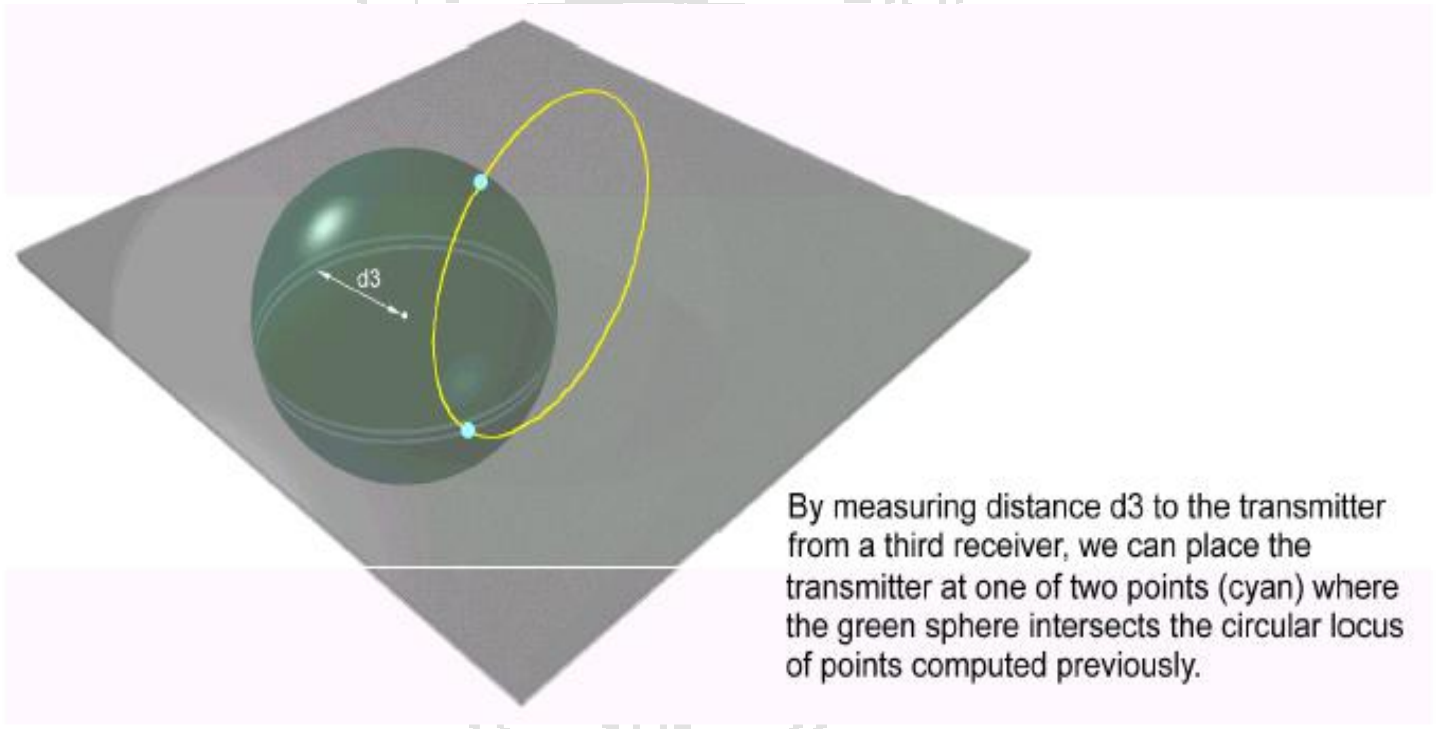
What if the circles do not intersect due to measurement errors (e.g. due to fading etc.)?
→will have to identify the best 'guess' given errors

Location in 3D



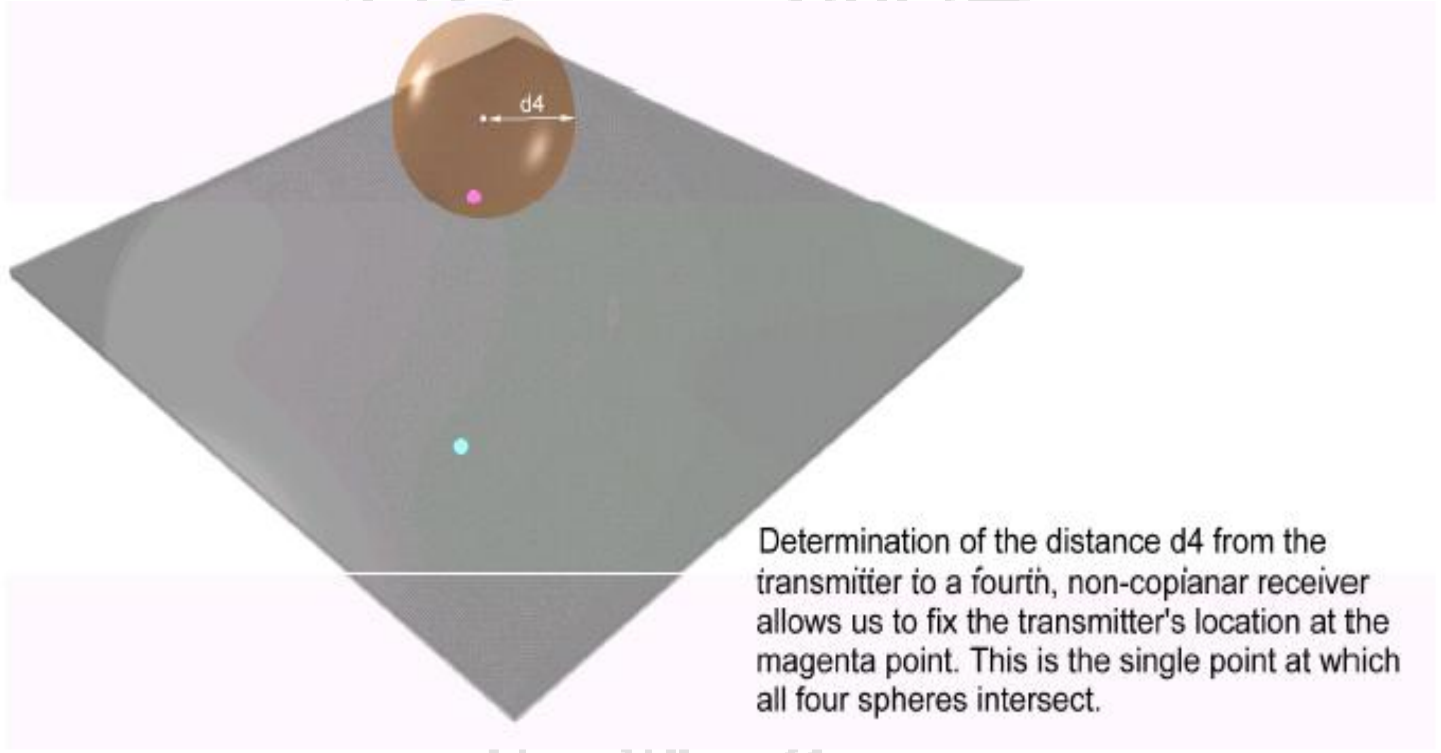
Measurement of distances d1 & d2 to the transmitter from two receivers allows us to place the transmitter on a circular locus of points (yellow) where the red and blue spheres intersect.

Location in 3D



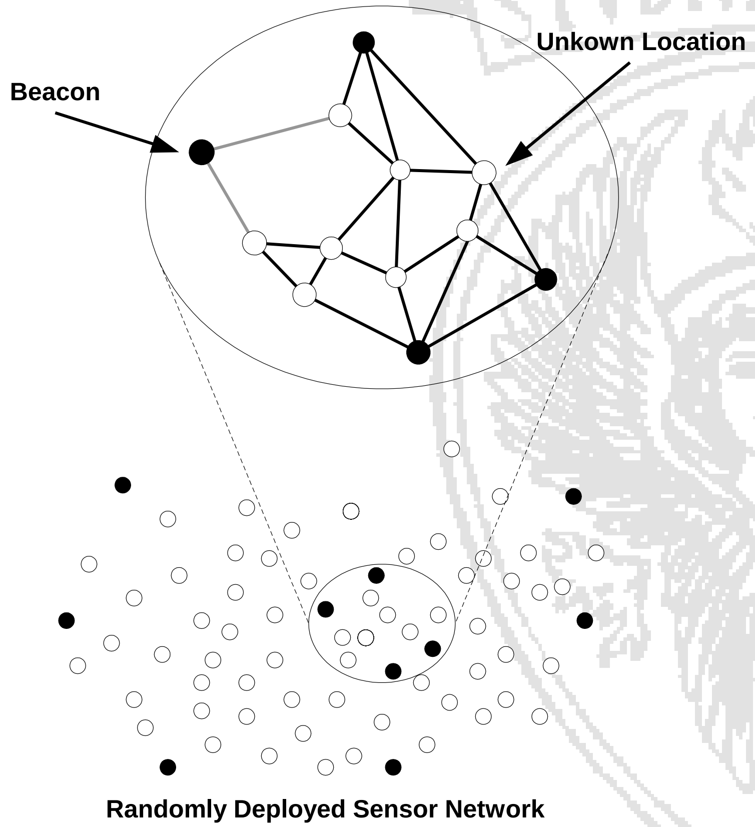
By measuring distance d3 to the transmitter from a third receiver, we can place the transmitter at one of two points (cyan) where the green sphere intersects the circular locus of points computed previously.

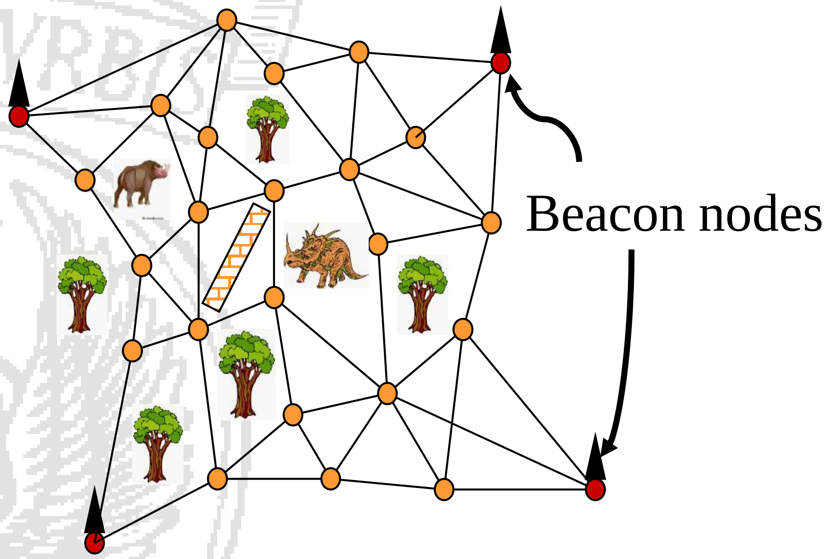
Location in 3D



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

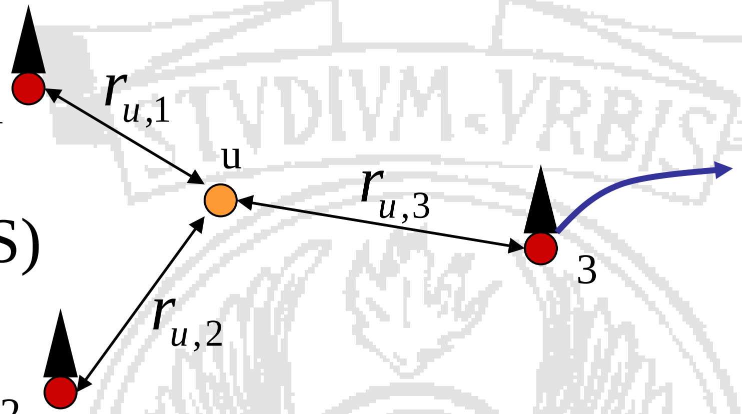




- A small fraction of the nodes is aware of their locations
- Rest need to collaborate to estimate their locations

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

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

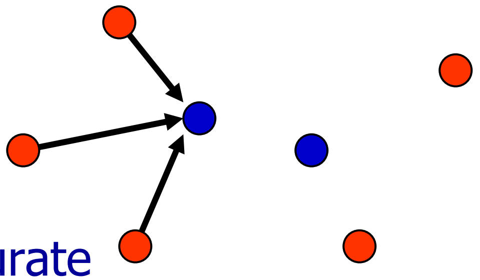
$(\hat{x}_u, \hat{y}_u)$ - initial position estimate for node *u*

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

Ref: based on slides by Andreas Savvides

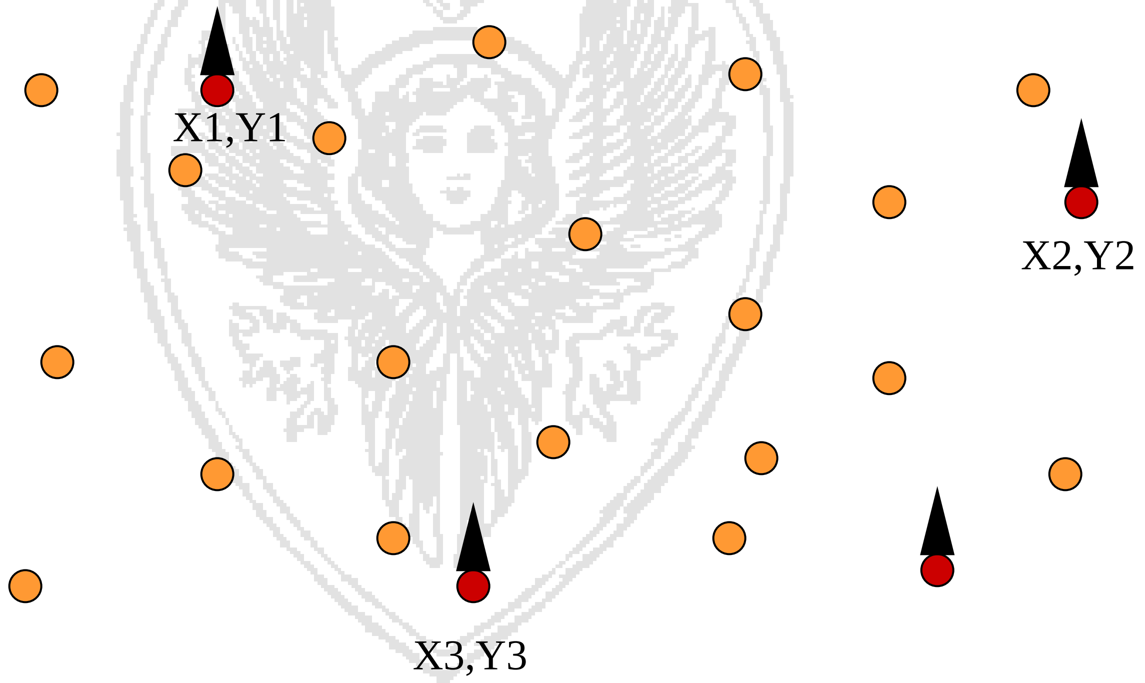
Range free localization

- Non usa ranging, ma solo informazioni che si possono ottenere tramite algo di routing tradizionali

- Idee su come possa funzionare?

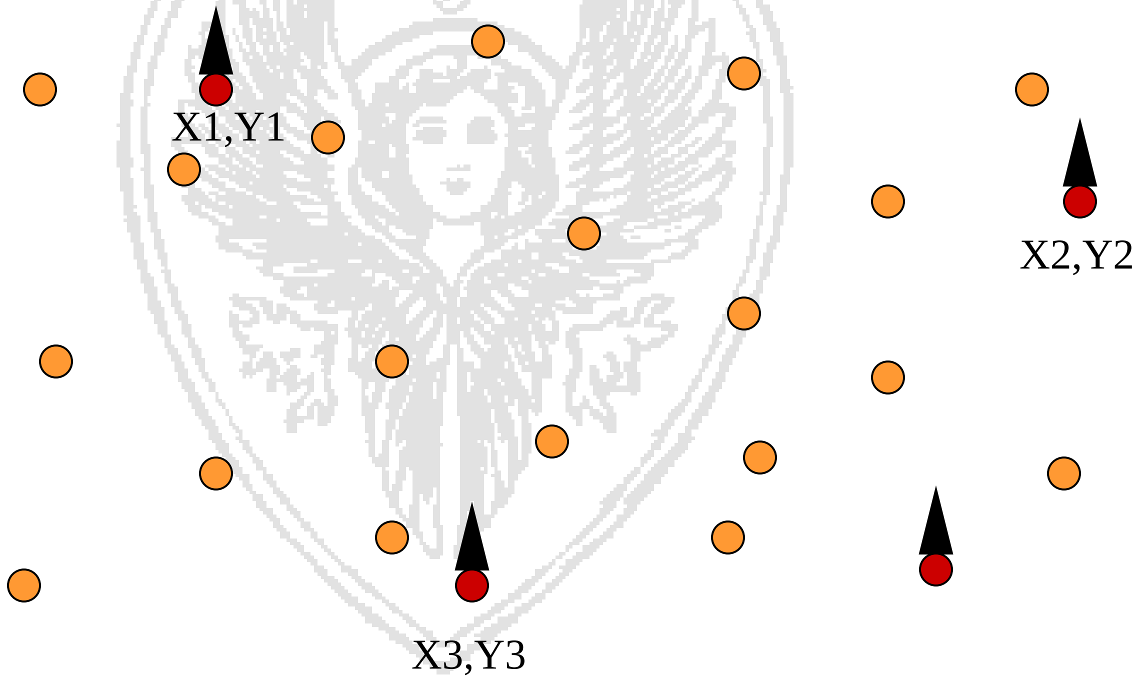
Qualche idea sull' approccio

- 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

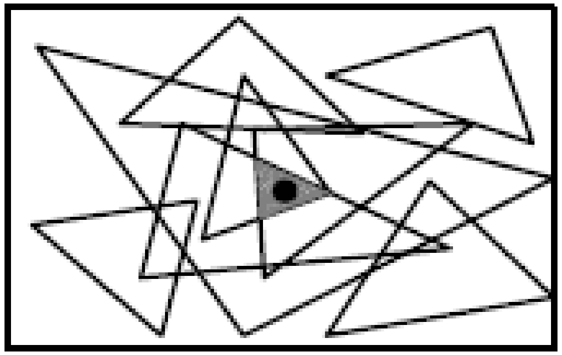
- Pro: Non serve extra HW
- Cons: si perde in precisione

→ DV-hop

SAPIENZA
UNIVERSITÀ DI ROMA

A different range free approach: APIT

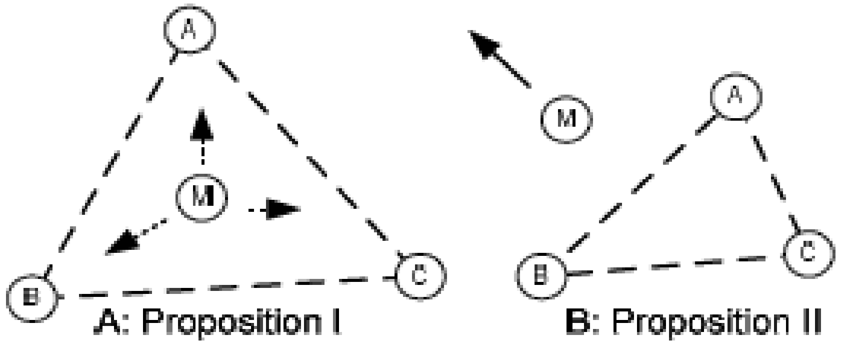
- "Range free" → Signal strength used only to detect whether a node is closer or far away from an anchor NOT to estimate distance from anchors
- Assumptions
  - Anchors have a long transmission range
  - High density scenario
- Point in Trangulation Test (PIT)
- Based on beaconing from anchors a node determins the triangular region within which it falls



```
Receive location beacons $(X_i, Y_i)$ from N anchors.
InsideSet = $\Phi$ // the set of triangles in which I reside
For (each triangle $T_i \in \binom{N}{3}$ triangles) {
    If (Point-In-Triangle-Test ($T_i$) == TRUE)
        InsideSet = InsideSet $\cup$ { $T_i$ }
    If( accuracy(InsideSet) > enough ) break;
}
/* Center of gravity (COG ) calculation */
Estimated Position = COG ( $\cap T_i \in$ InsideSet);
```

PIT test

- Proposition I
  - If M is inside triangle ABC , when M is shifted in any direction, the new position must be nearer to (further from) at least one anchor A,B,C.

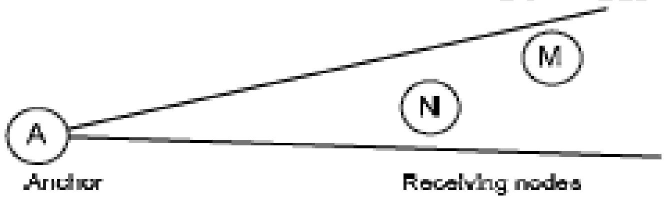


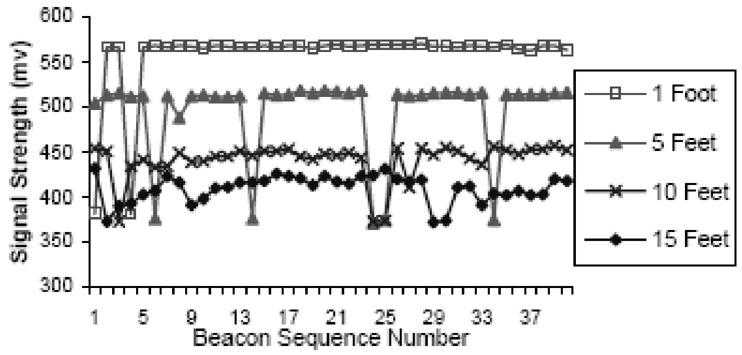
- Proposition II
  - If M is outside triangle ABC, when M is shifted, thre must be a direction in which the position of M is furthest from or closer to all three anchors A,B and C.

SAPIENZA
UNIVERSITÀ DI ROMA

PIT implementation

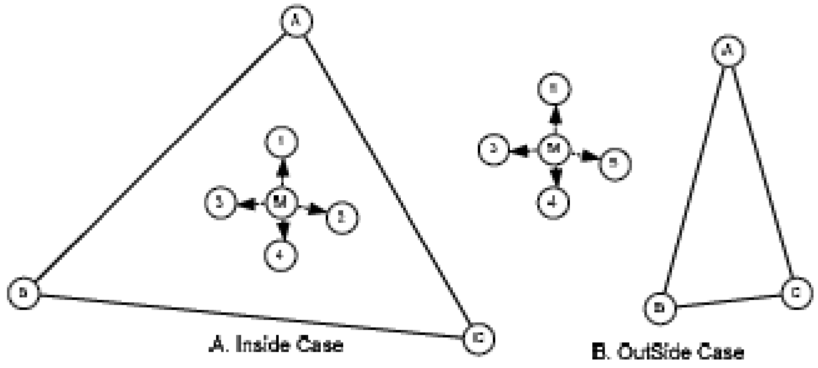
- Perfect P.I.T. Test
  - If there exist a direction such that a point adjacent to M is furthest/closer to points A,B and C simultenaously, then M is outside ABC. Otherwise M is inside.
- How does a node recognize directions or departure from an anchor without moving? How to test all possible directions?
- ANSWER1: Nodes far away tend to experience lower signal strength





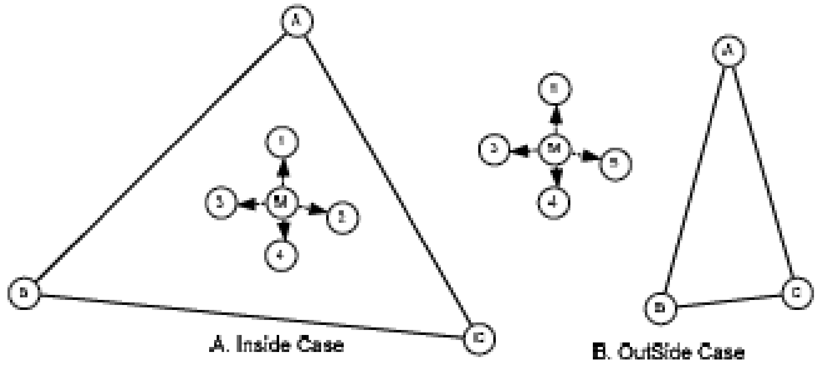
PIT implementation

- Perfect P.I.T. Test
  - If there exist a direction such that a point adjacent to M is furthest/closer to points A,B and C simultenaously, then M is outside ABC. Otherwise M is inside.
- How does a node recognize directions or departure from an anchor without moving? How to test all possible directions?
- RULE: If no neighbor of M is further from/closer to all three anchors A, B and C simultaneously. M assumes that it is inside triangle ABC. Otherwise, M assumes it resides outside this triangle



PIT implementation

- Perfect P.I.T. Test
  - If there exist a direction such that a point adjacent to M is furthest/closer to points A,B and C simultenaously, then M is outside ABC. Otherwise M is inside.
- How does a node recognize directions or departure from an anchor without moving? How to test all possible directions?
- RULE: If no neighbor of M is further from/closer to all three anchors A, B and C simultaneously. M assumes that it is inside triangle ABC. Otherwise, M assumes it resides outside this triangle



Approximate PIT test

Exchange of info on signal strength to anchors with neighbors

Possible errors

- Need for a dense environment. However:
  - inToOut Error: borderline node (with a neighbor outside the triangle)
  - OutToin Error: irregular placement resulting into an erroneous decision

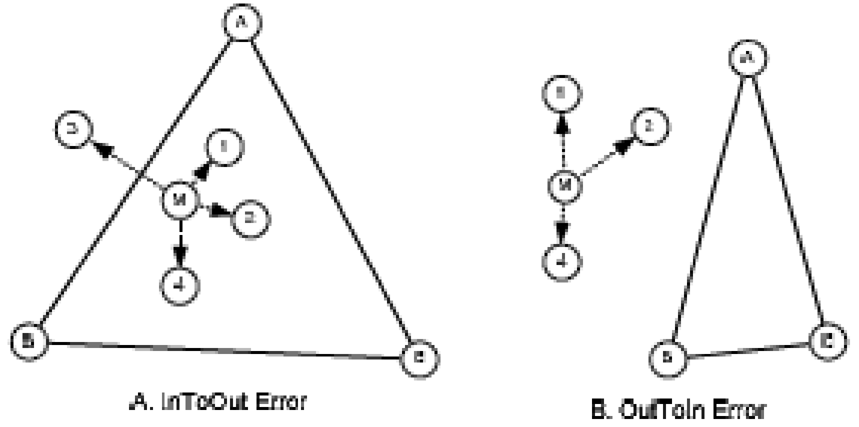


Figure 6: Error Scenarios for the APIT Test.

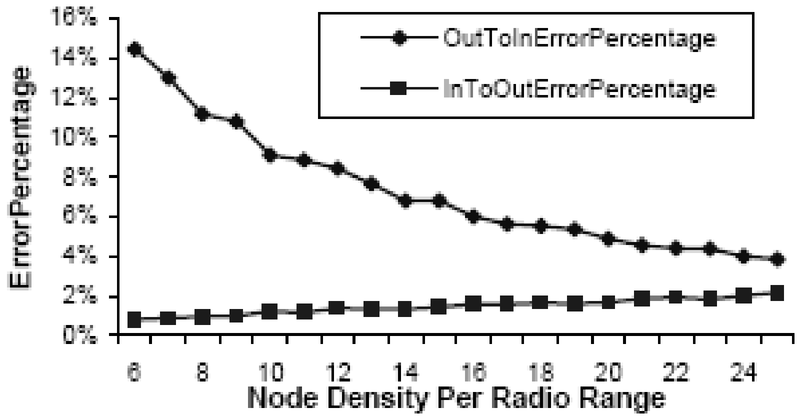
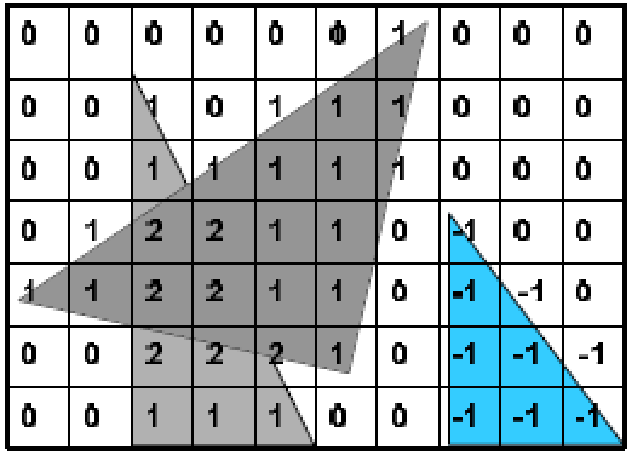


Figure 7: APIT Error under Varying Node Densities

Aggregation of APIT

- Divide deployment area into a grid array
- For each triangle corresponding to a square add 1 if the node is inside that triangle, subtract 1 if the node is outside that triangle



0	0	0	0	0	0	1	0	0	0
0	0	1	0	1	1	1	0	0	0
0	0	1	1	1	1	1	0	0	0
0	1	2	2	1	1	0	-1	0	0
1	1	2	2	1	1	0	-1	-1	0
0	0	2	2	2	1	0	-1	-1	-1
0	0	1	1	1	0	0	-1	-1	-1

The pseudo code for APIT aggregation is as follows:

```
For (each triangle $T_i \in \binom{N}{3}$ triangles) {
    If (APIT($T_i$) == Out ) AddNegtiveTriangle($T_i$);
    If (APIT($T_i$) == In ) AddPositiveTriangle($T_i$);
};
Find the area with Max values;
```

Clustering per reti ad hoc

Sistemi Wireless

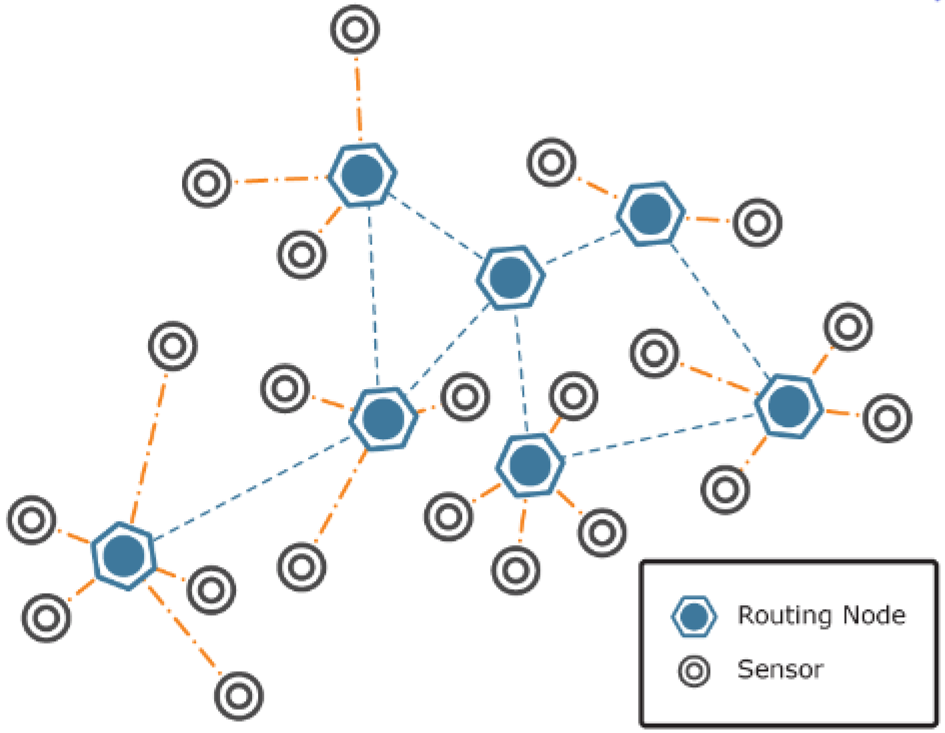
2011/2012

Scalability Problems and Clustering

- What happens to protocols when the number of network nodes grows?
  - Especially crucial in WSNs
- A traditional networking solution: Hierarchical organization of the nodes
- Network nodes are grouped into clusters
- Some nodes, locally the "best," are selected to coordinate the clustering process: Clusterheads
  - Clusterheads arbitrate access to the channel, perform data fusion etc
  - Clusterheads can be interconnected in a "backbone"
    - ✓ Decreasing routing tables and control overhead for routing

Scalability Problems and Clustering

- What happ vork nodes grows?
  - Especial
- A traditiona zation of the nodes
- Network nc
- Some node inate the clustering p
  - Clusterh data fusion etc
  - Clusterheads can be interconnected in a "backbone"
    - ✓ Decreasing routing tables and control overhead for routing



How to Select the Best Nodes

- Independence of the clusterheads
  - no pair of clusterheads has a link connecting them in the topology graph
- Dominance of the clusterheads
  - for each node in the network there is at least one clueterhead covering it
- Possibility to express “preferences”, based on suitability of a node to serve as clusterhead
  - resource consuming role
  - to reduce cost to clustering and backbone maintenance
- Distributed operations
- Fast and simple implementation

Maximum Independent Set (MIS)

- A subset V' of the vertices V of a graph G=(V,E) is independent when for each u,v ∊ V' the edge {u,v} ∉ E
- MIS is an Optimization Problem
- Input: A Graph G=(V,E) with n vertices
- Output: A subset V' of V that is independent and has maximum size

MIS: Hardness

- No known algorithm computer a MIS in polynomial time
- Need for approximate solutions
- And approximation algorithm is an algorithm that produces a solution that is not optimal, but that approximates it
- We sacrifice optimality in favor of a “good” solution that can be computed efficiently

MIS is HARD to Approximate

- Bad news
  - Not only MIS is computationally hard
  - It is also hard to approximate:
    - ✓ Approximate solutions are not so good
    - ✓ They are "unboundedly" far from the optimum

- We consider the simple greedy heuristic for the MIS

Greedy Heuristic for MIS, 1

- Select the vertex with minimum degree and put it in the MIS
  - The degree of a vertex is the number of its neighbors
    - ✓ Cardinality of its adjacency list
  - Keep going till all the vertices are either in the MIS or COVERED by a vertices in the MIS

Greedy Heuristic for MIS, 2

```
MIS(V,E,d) // d is the vector of degrees
  mis = Ø
  while V ≠ Ø do
    v = vertex with min degree
    mis = mis U {v}
    V = V – {{v} U  N(v)}
  return mis
```

On MDS—what if we look at UDG graphs?

- Bad news: Still computationally hard
- Better news: Minimum DS It is approximable “up to a constant”
  - It means that the ratio between the size of a DS computed by MIS greedy on UDGs and the size of a MDS is < c, c a constant
- This constant is 5

Greedy MIS: Maximal Solution

- The greedy solution provides a maximal independent set
  - An independent set is maximal when, if you add a vertex, the set is no longer independent
    - ✓ You cannot make a maximal independent set bigger

- This solution is also a minimal dominating set
  - A dominating set D subset of V is a set such that a vertex v ∊ V is either in D or it has a neighbor in D
    - ✓ MIS is a dominating set
    - ✓ it is a minimal dominating set, i.e. removing any node in the MIS from the dominating set at least that node would be uncovered

- Solutions we will see are variant of this approach

Greedy MIS for MDS on UDG is 5-approximable, 1

- Key fact: In a UDG disk (radius 1) there are at most 5 independent nodes
- Consider an Optimal solution and a Greedy solution
- Since Opt is dominant, it dominates Greedy
- Assign every vertex of Greedy to one dominator in Opt (choose one if more)

Greedy MIS for MDS on UDG is 5-approximable, 2

- For each u in Opt consider its assigned vertices $v_1(u)$, $v_2(u)$, …, $v_k(u)$ of Greedy
- How big is k?
- Well, all $v_i(u)$ must be distant 1 from u and they also have to be independent
- Greedy: at most 5 times bigger than Opt

MIS and Dominating Sets and Wireless Networks

- UDGs provide a first approximation model for ad hoc networks
- IS and DS are useful for clustering ad hoc networks
  - Gives the network a hierarchical organization
  - Decreases the amount of information at each node
  - Enhances scalability
  - Helps in “resource assignment”

Advantages of hierarchical organization

- routing always through the clusterhead
- data aggregation at the clusterhead
- easy to locally synchronize nodes within the cluster, using TDMA MAC protocol for intra-cluster communication and different MAC protocols (e.g. CDMA) for inter-cluster communications

MWIS-Based Clustering

- MWIS = Maximal Weight Independent Set
- Clustering selection based on generic **weights** (real numbers > 0)
  - Mobility/node related parameters
  - Generalizes previous "Independent Set" solutions

DCA: Distributed Clustering Algorithm, 1

- Assumptions
  - Knowledge of IDs and weights of one-hop neighbors
  - Broadcast transmission of a packet in finite time (a “step”)
  - Nodes do not move during clustering

DCA, 2

- (Only) Two messages:
  - CH(v): Sent by a clusterhead v
  - JOIN(u,t): Sent by ordinary node u when it joins the cluster of clusterhead t
- Three (simple) procedures:
  - Init (start up)
  - OnReceivingCH(v), OnReceivingJOIN(u,v) (message triggered)

DCA

- Ogni nodo conosce i suoi vicini ed il loro peso
- Un nodo è init se ha il peso più grande dei pesi dei suoi nodi vicini
- Gli init node diventano clusterhead e invitano i loro vicini a far parte del loro cluster
- Un nodo x aspetta di ricevere messaggi dai vicini di peso maggiore prima di prendere una decisione
  - Se un vicino di peso maggiore lo invita a far parte del suo cluster allora x entra a far parte del cluster del vicino di peso maggiore che lo contatta (inviando un messaggio di Join) → nodo ordinario
  - Altrimenti diventa clusterhead lui stesso e invia un messaggio di CH

DCA

- Due tipi di messaggi
  - CH(v) è usato da un nodo v per rendere consapevoli i suoi vicini del fatto che ha assunto il ruolo di clusterhead
  - JOIN(v,u) è usato dal nodo v per comunicare ai suoi vicini che sarà parte di un cluster il cui clusterhead è il nodo u

DCA

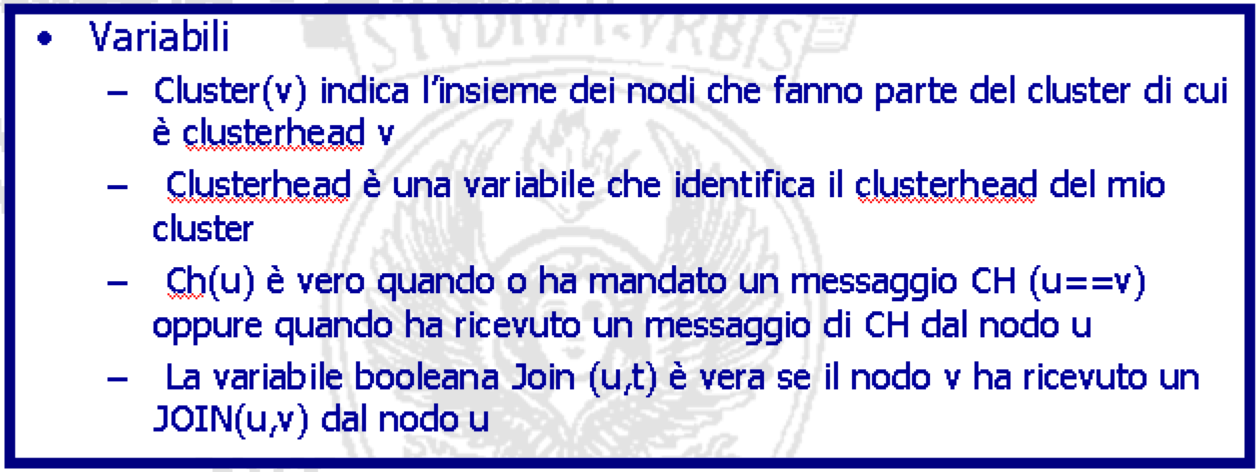
- Variabili
  - Cluster(v) indica l’insieme dei nodi che fanno parte del cluster di cui è clusterhead v
  - Clusterhead è una variabile che identifica il clusterhead del mio cluster
  - Ch(u) è vero quando o ha mandato un messaggio CH (u==v) oppure quando ha ricevuto un messaggio di CH dal nodo u
  - La variabile booleana Join (u,t) è vera se il nodo v ha ricevuto un JOIN(u,v) dal nodo u

DCA-Procedure (eseguite dal nodo v)

http://twiki.di.uniroma1.it/twiki/viewfile/Reti_Avanzate/AA0910/WebHome?rev=1;filename=basagni99distributed.pdf

- Init

```
Se tutti i nodi vicini hanno un peso minore di v
    invia CH(v);
    Cluster(v)=Cluster(v)U{v};
    Ch(v)=true;
    Clusterhead=v;
```



- Variabili
  - Cluster(v) indica l'insieme dei nodi che fanno parte del cluster di cui è clusterhead v
  - Clusterhead è una variabile che identifica il clusterhead del mio cluster
  - Ch(u) è vero quando o ha mandato un messaggio CH (u==v) oppure quando ha ricevuto un messaggio di CH dal nodo u
  - La variabile booleana Join (u,t) è vera se il nodo v ha ricevuto un JOIN(u,v) dal nodo u

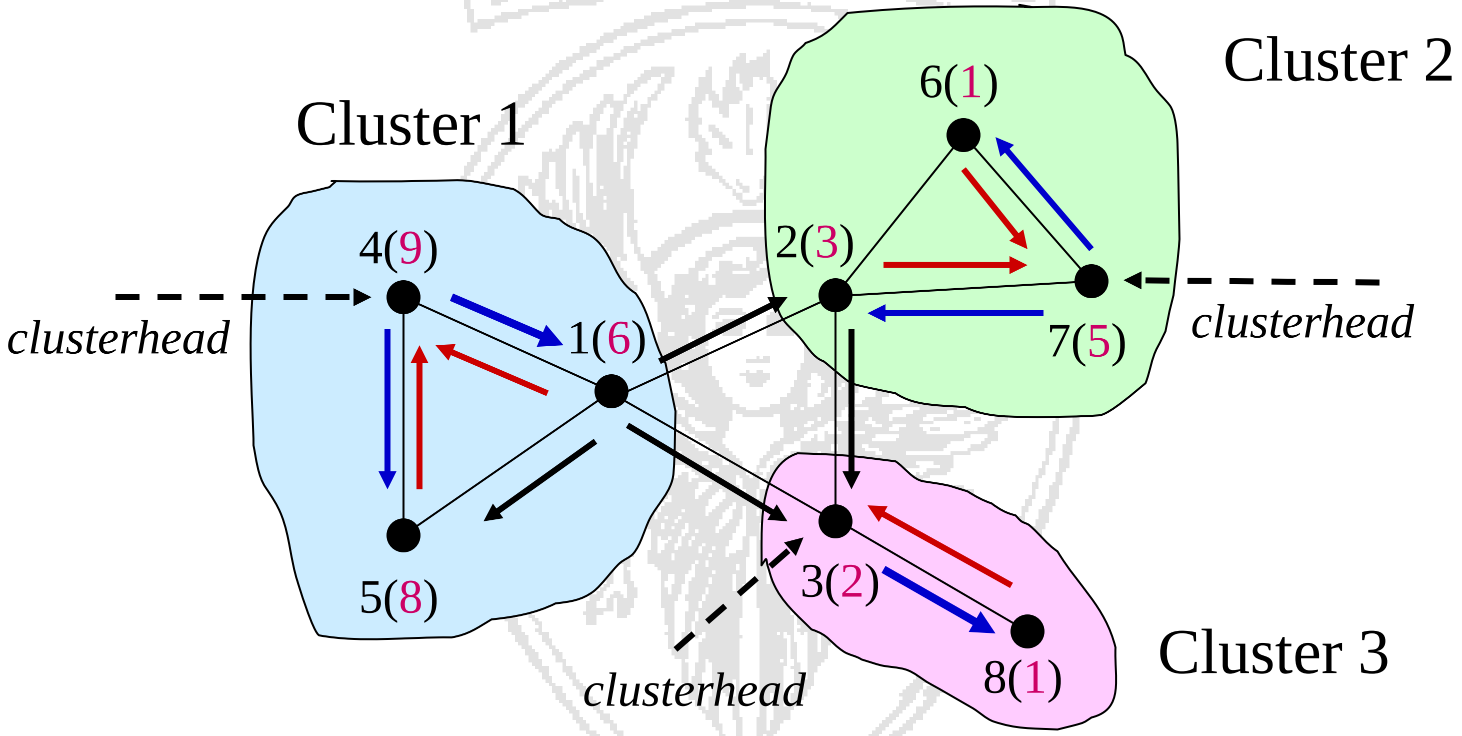
DCA-Procedure (eseguite dal nodo v)

```
• On receiving CH(u)
  Ch(u)=true;
  Se u ha un peso maggiore di me e i vicini di peso
  maggiore di v con peso maggiore di u hanno tutti
  mandato un Join, allora
        Clusterhead=u;
        invia JOIN(v,Clusterhead);
```

DCA-Procedure (eseguite dal nodo v)

```
On receiving JOIN(u,t)
    Join(u,t)=true;
    Se v è un clusterhead allora se t==v
            {Cluster(v)=Cluster(v)U{u};
            Se ho ricevuto Join da tutti i vicini più          piccoli EXIT}
    Altrimenti se tutti i vicini di peso maggiore hanno preso una decisione
    sul ruolo.
            {e tutti i vicini di peso maggiore hanno mandato JOIN
                    {mandiamo un CH(v);
                    Cluster(v)=Cluster(v)U{v};
                    Clusterhead =v;
                    Se si è ricevuto JOIN da tutti i vicini minori EXIT.}
            Altrimenti se uno o più vicini di peso maggiore hanno mandato
    un CH
                    {Clusterhead=il vicino con peso più grande tra quelli
    che sono                diventati clusterhead e mi hanno invitato.
                    manda JOIN(v,Clusterhead);
                    EXIT;}
            }
   }
```

Example



I Step II Step III Step IV Step V Step

DCA: Provable Properties

- Consider

$$\tau: V \rightarrow \{1,2,3, \ldots , 2k\}$$

V = set of network nodes, k = number of clusters

- **Proposition**: Each node v in V sends exactly one message by $\tau(v)$ steps
- **Corollary 1**: DCA message complexity is n =|V|

Joining Clusterheads: Dynamic Backbone

- A theorem from Chlamtac and Farago:

  If a network is connected, and DCA is used, then if and only if each clusterhead is linked to all the clusterheads at most three hops away, the resulting backbone network is guaranteed to be connected

PROVATE A DIMOSTRARLO

4 Backbone Formation Protocols

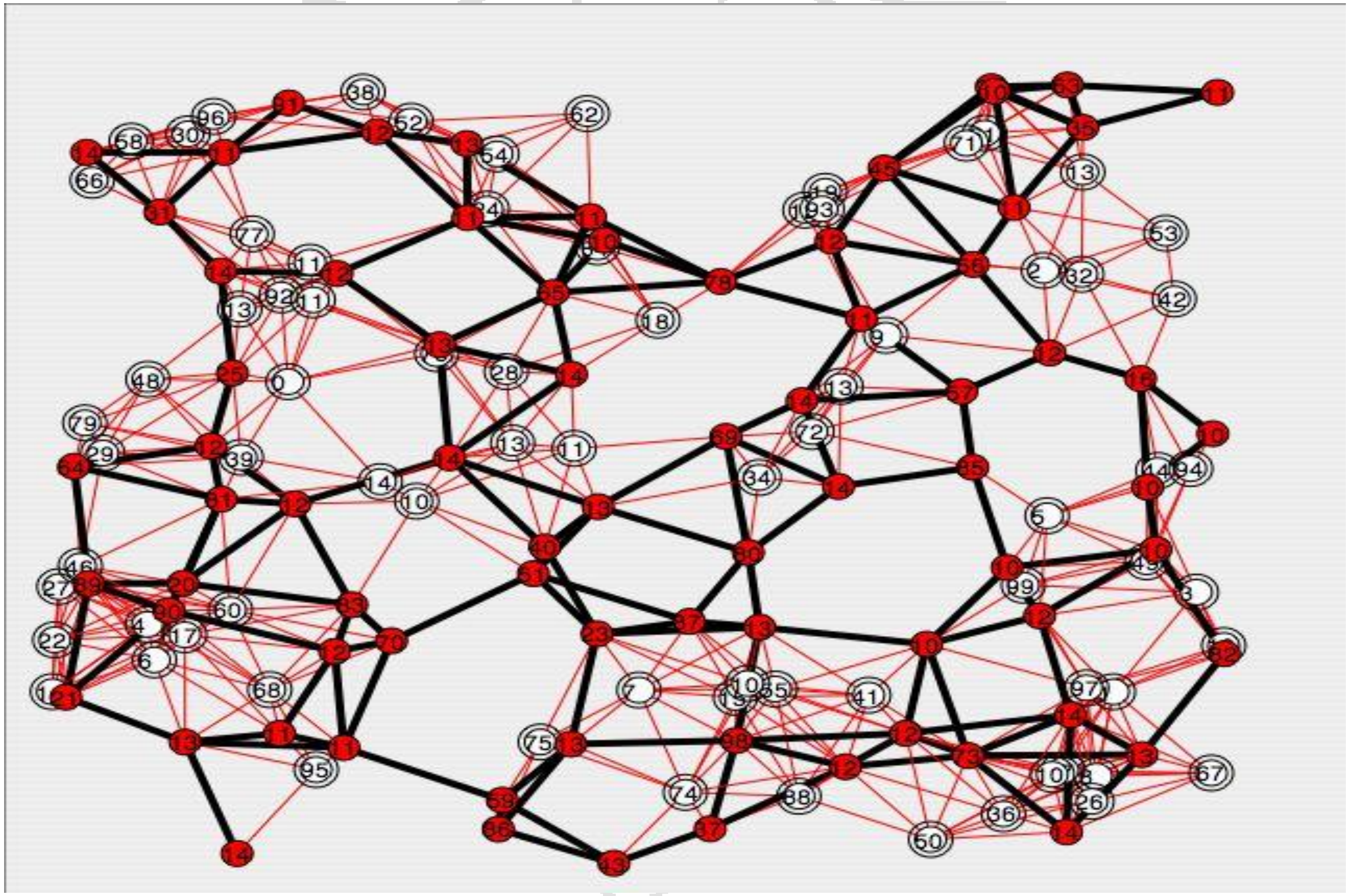
- 3 representatives of major approaches
  - Selection of independent set of nodes and backbone construction (DCA)
  - Rich dominating set formation and pruning (WuLi)
  - Two-phase algorithm with theoretical guarantees (WAF)
- 1 proposal after the performance comparison (DCA-S)

Distributed Clustering Algorithm (DCA)

- Distributed and localized implementation of the greedy for independent set
- Takes node status into account for node selection
- Independent nodes are joined into a connected backbone (connectivity is guaranteed) via gateways
- Low degree of parallelism (“dependency chains”)

SAPIENZA
UNIVERSITÀ DI ROMA

A DCA Backbone



WuLi: Wu and Li protocol

- Distributed and localized protocols for forming a connected dominating set
- Build a rich connected dominating set
- Applies localized rules for pruning unnecessary nodes/links
- High degree of parallelism ("all localized")

WuLi: Wu and Li protocol

- Distributed and localized protocols for forming a connected dominating set
- Build a rich
- Applies localized rules for pruning unnecessary nodes/links
- High degree of parallelism ("all localized")

If a vertex v has two neighbors which are not in visibility range it enters the set C

WuLi: Wu and Li protocol

- Distributed and localized protocols for forming a connected dominating set
- Build a rich connected dominating set
- Applies localized rules for pruning unnecessary nodes/links
- High degree of parallelism ("all localized")

What is needed is, from the neighbors, whether they are in C and their list of neighbors.

WuLi-dominance property

da 'On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks', Wu and Li

- **THEOREM 1: Given a G = (V, E) that is connected, but not completely connected, the vertex subset V', derived from the marking process of WuLi (i,e, the rule according to which if a node has two neighbors which are not in visibility it enters V'), forms a dominating set of G.**
- **DEFINITION: if the node is in V' it is marked T ; otherwise it is market F**

- **PROOF: Randomly select a vertex v in G. We show that v is either in V' (a set of vertices in V that are marked T) or adjacent to a vertex in V' . Assume v is marked F, if there is at least one neighbor marked T, the theorem is proved. When all its neighbors are marked F, we consider the following two cases: (1) All the other vertices in G are neighbors of v. Based on the marking process and the fact that m(v)=F, all these neighbors must be pairwise connected, i.e., G is completely connected. This contradicts to the assumption that G is not completely connected. (2) There is at least one vertex u in G that is not adjacent to vertex v. Construct a shortest path, {v,vi,v2, ... u}, between vertices v and u. Such a path always exists since G is a connected graph. Note that v2 is u when v and u are 2-distance apart in G. Also, v and v2 are not directly connected; otherwise, {v, v2,. .. u} is a shorter path between v and u. Based on the marking process, vertex vi, with both v and v2 as its neighbors, must be marked T. Again this contradicts the assumption that v's neighbors are all marked F. CVD**

WuLi (first step produces a CDS)

- THEOREM 2: The resulting DS C=G′ is a connected graph.
- DEF: G′ is the subgraph G′=(V′,E′). E′ includes links (x,y)

- PROOF: We prove this theorem by contradiction. Assume G′ is disconnected and v and u are two disconnected vertices in G′. Assume $dis_G(v,u) = k+1 > 1$ and{v,v1,v2,...,vk,u} is a shortest path between vertices v and u in G. Clearly, all v1,v2, ...vk are distinct and among them there is at least one vi such that m(vi) = F (otherwise, v and u are connected in G′). On the other hand, the two adjacent vertices of vi, vi-1 and vi+i, are not connected in G (otherwise, {v, ..vi, vi+1, ... vk,u} would be a shorter path). Therefore, m(vi) =T based on the marking process.

Rule 1: for each pair of nodes u and v in C the one with the smallest ID, say v, can be removed from C if v and all its neighbors are covered by u

Rule 2: Assume nodes u,v, and w are in C and assume that v's ID is the smallest. If u and w are neighors of v and are in each other transmission range and if each neighbor of v is covered by u and w, then v can be removed from C.

- Distr
  conn
- Buil
- Appl
  node
- High degree of parallelism ("all localized")

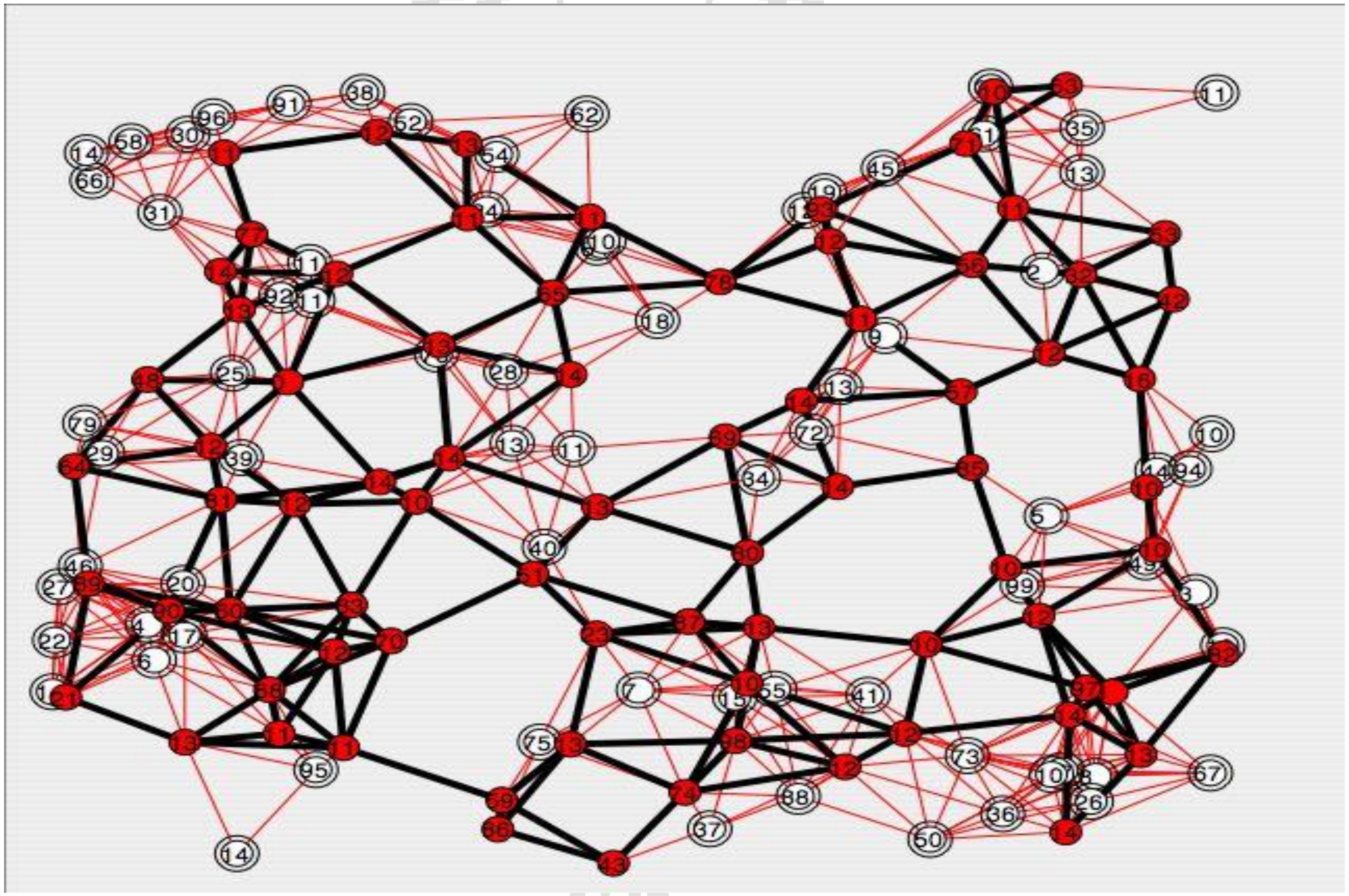
Mantengono la proprietà di connessione e di dominanza

WuLi: Wu and Li protocol

- Distributed and localized protocols for forming a connected dominating set
- Build a rich connected dominating set
- Applies localized rules for pruning unnecessary nodes/links
- High degree of parallelism (“all localized”)

What is needed is, from the neighbors, whether they are in C and their list of neighbors.

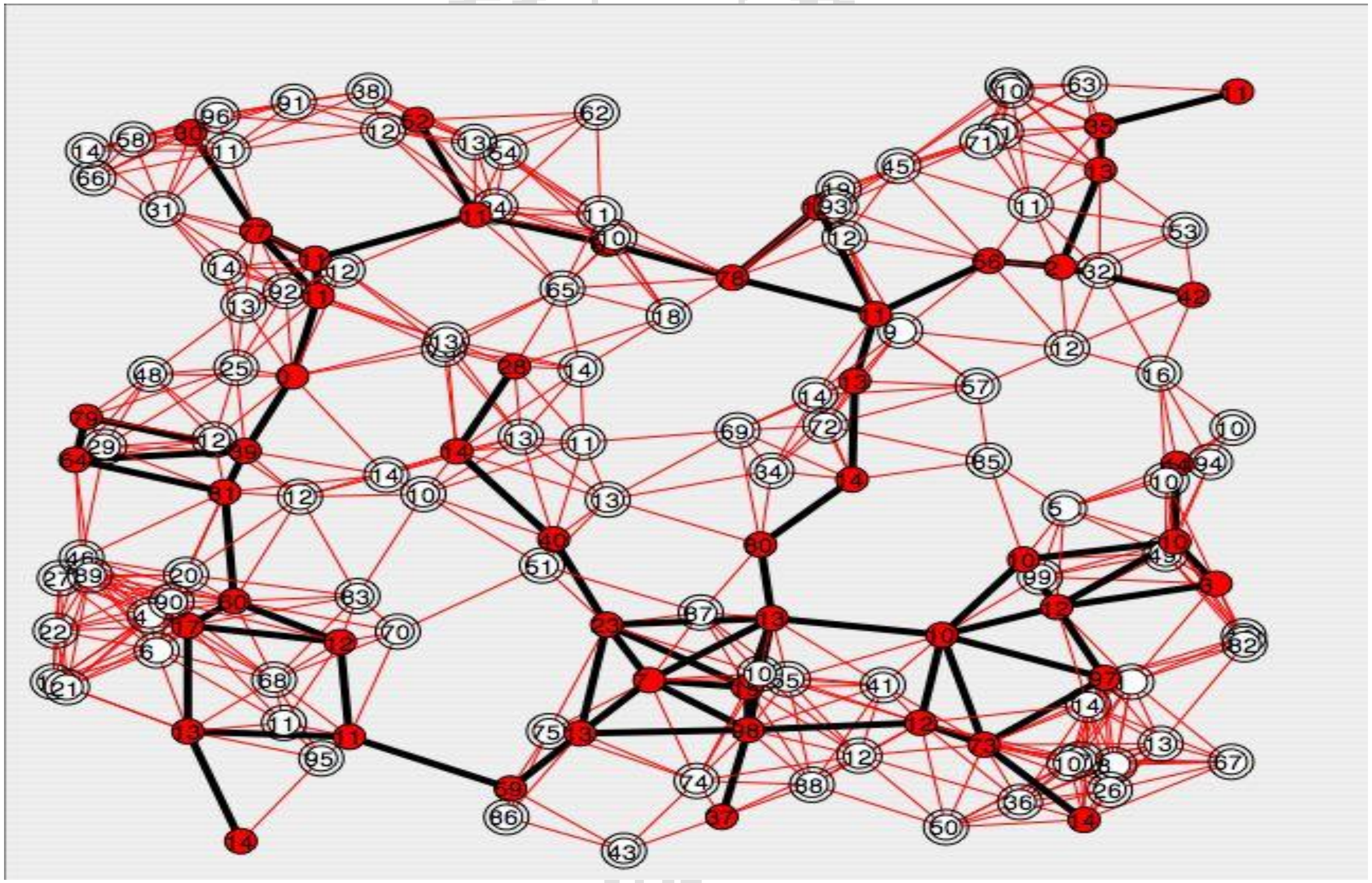
A WuLi Backbone



WAF: Wan, Alzoubi and Frieder

- Two phases
  - Leader election: One node is chosen among all network nodes to be the root of a tree
  - Nodes at different levels of the trees can be chosen to form a connected dominating set
- The “leader election tree” is quite expensive
- Very low degree of parallelism

A WAF Backbone



DCA-S: DCA Sparsified

- Build a connected dominating set (say, with DCA) and consider its spanned sub-graph H (include gateways)
- Erdös: If a graph does not have small cycles then it is sparse
- Find and break small cycles (small=log n)
  - In practice we search and break cycles with 3 and 4 links
- Breaking cycles does not compromise connectivity

Simulation Results

- Metrics (all averages)

1. Protocol duration
2. Operation overhead (in bytes)
3. Energy consumption (per node)
4. Backbone size
5. Route length
6. Backbone robustness (node deaths for disconnections)

Simulation Results, 2

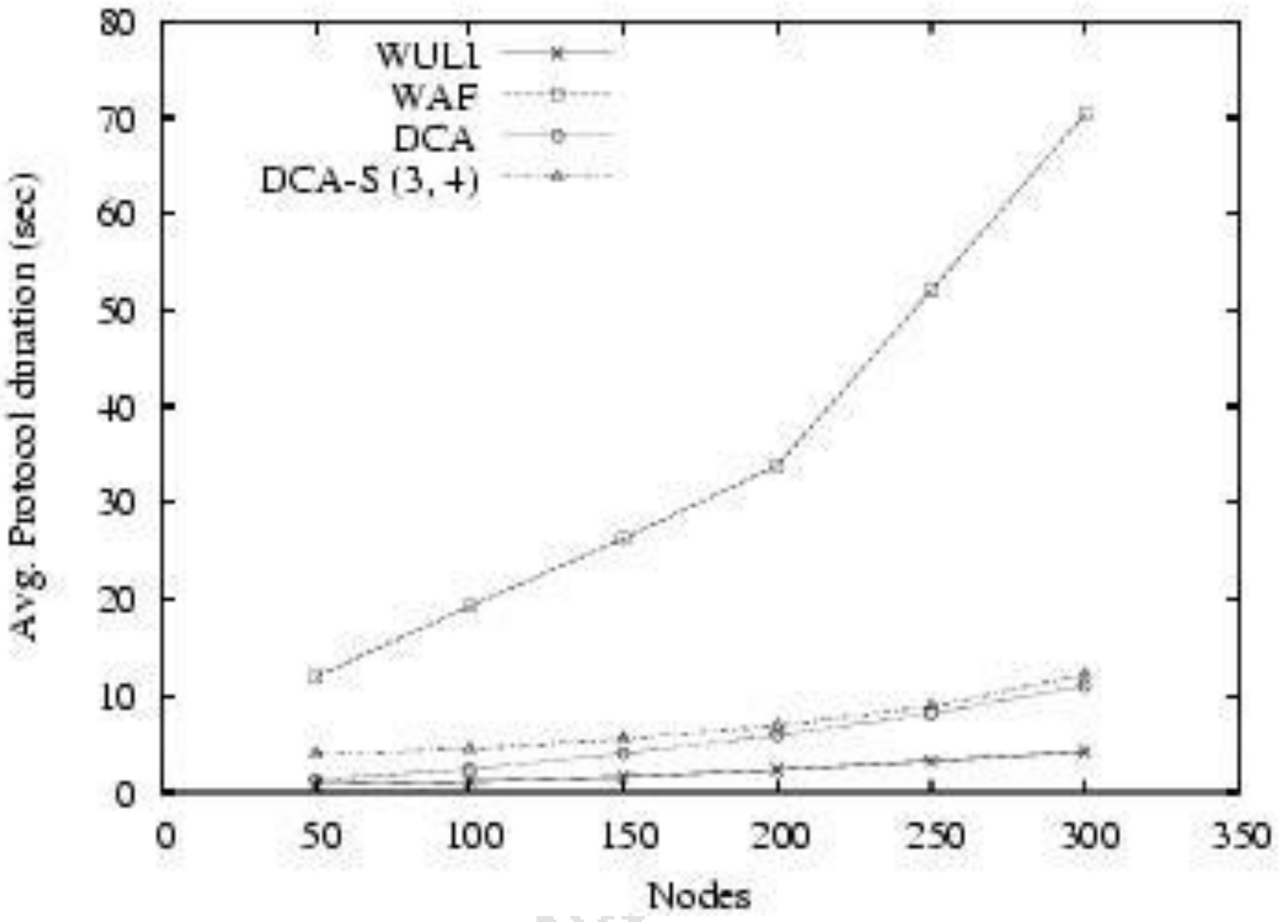
- Parameters of ns2-based simulations
  - Nodes: ≤ 300, IST EYES prototype
    - ✓ Tx range: 30m
    - ✓ Initial (residual) energy: 1J
    - ✓ Tx, Rx, idle power: 24, 14.4, 0.015 (mW)
  - Area: 200 x 200m
  - Six scenarios with increasing densities (avg. degrees: 3.5 to 20)

Protocol Duration

- WuLi is fastest
  - Simple operation; parallelism
- DCA: Reasonably fast
  - Possible dependencies and gateway selection
- DCA-S: As DCA
  - The sparsification phase is executed by fewer nodes and requires little info exchange
- WAF: Slower
  - Non-trivial leader election

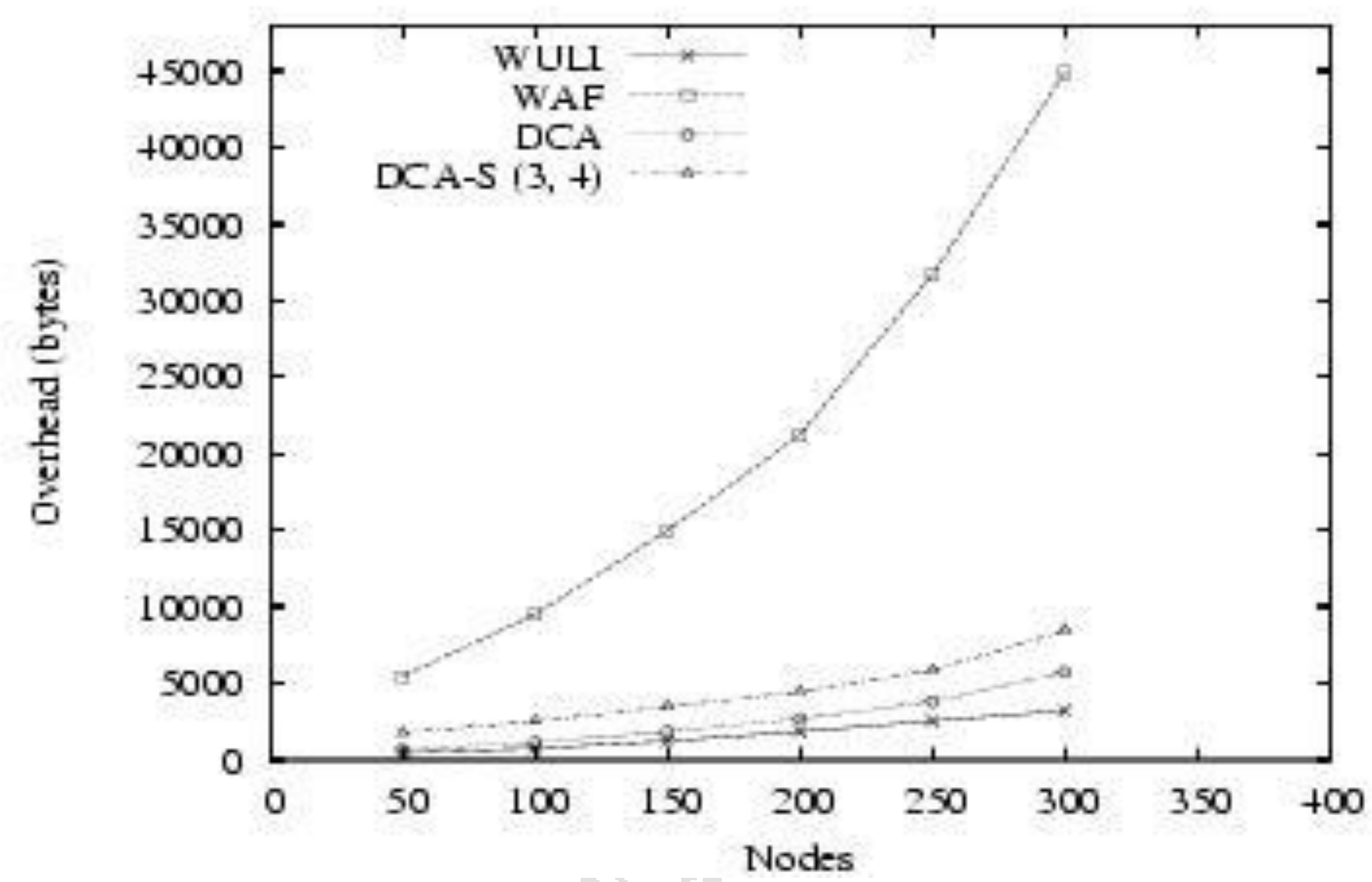
SAPIENZA
UNIVERSITÀ DI ROMA

Protocol Duration, 2



SAPIENZA
UNIVERSITÀ DI ROMA

Protocol Overhead



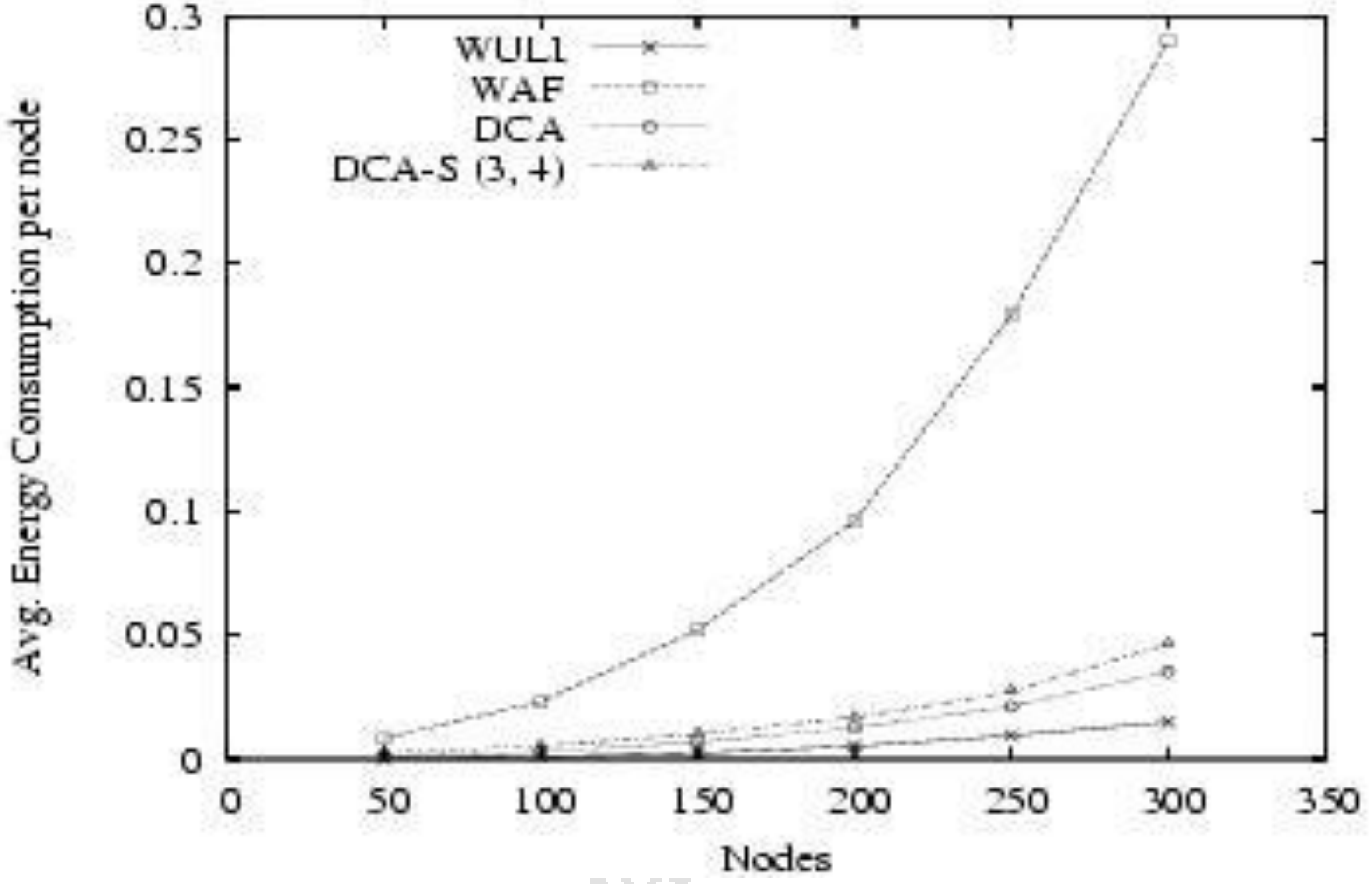
Protocol Overhead, 2

- Average number of protocol bytes per node
- WuLi: Best performing
  - Simple list exchange
- DCA(-S): Almost twice as much
  - Bit more info needed (weight, IDs, …)
- WAF
  - Leader election complexity

Energy Consumption

- Important metric per backbone set up and maintenance
- Similar to overhead results
- WuLi and DCA perform quite well
- DCA-S performs similarly: No difference in breaking cycles with 3 or 4 links
- WAF: High consumption due to first phase

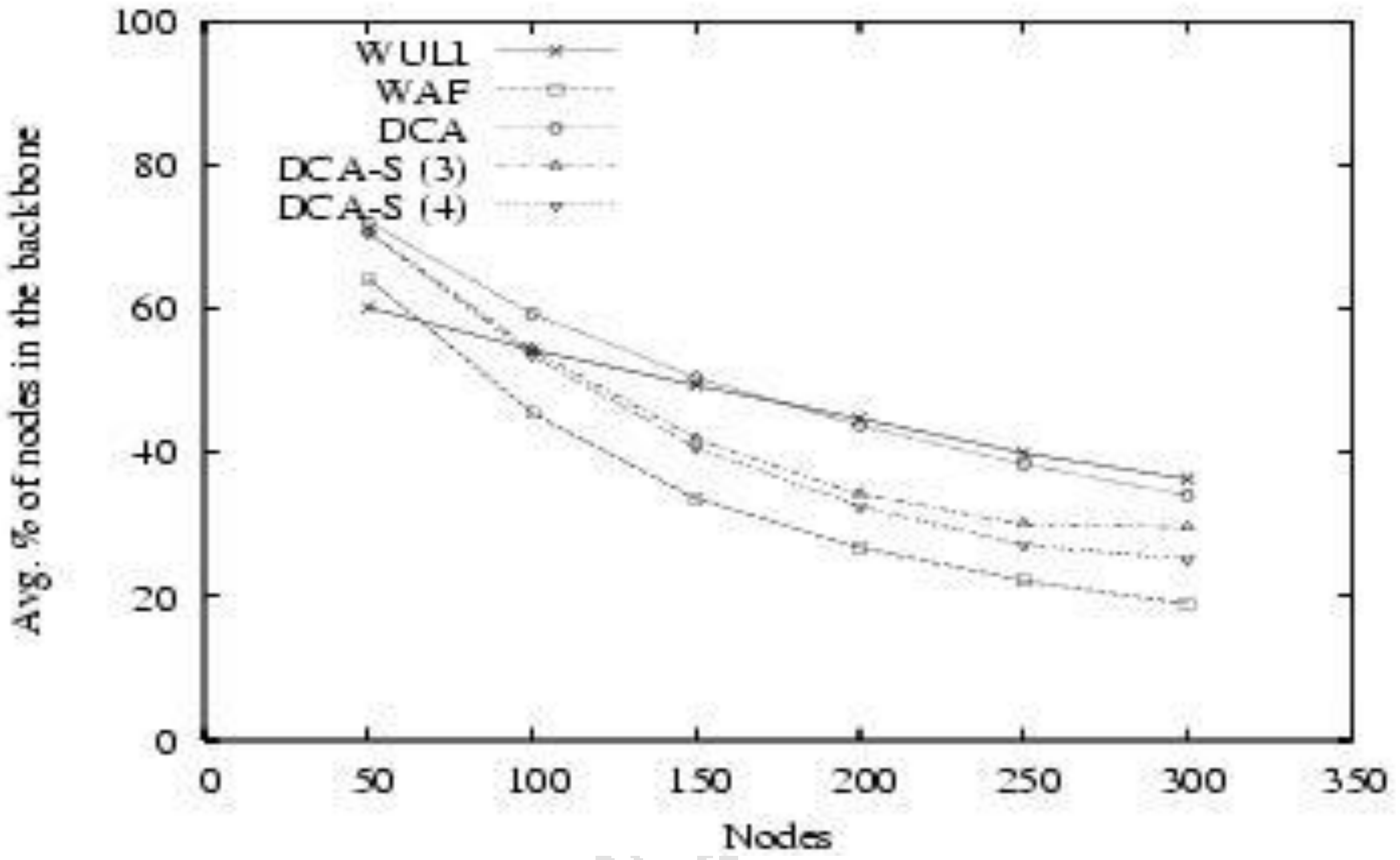
Energy Consumption, 2



Backbone Size

- Important metric: Routing info and awake/asleep cycles
  - Small backbone + role rotation: key for WSNs
- Decrease with n increasing (bigger clusters)
- WAF: "Slimmer" backbone (tree like)
- DCA-S, 4 < DCA-S, 3 < DCA < WuLi

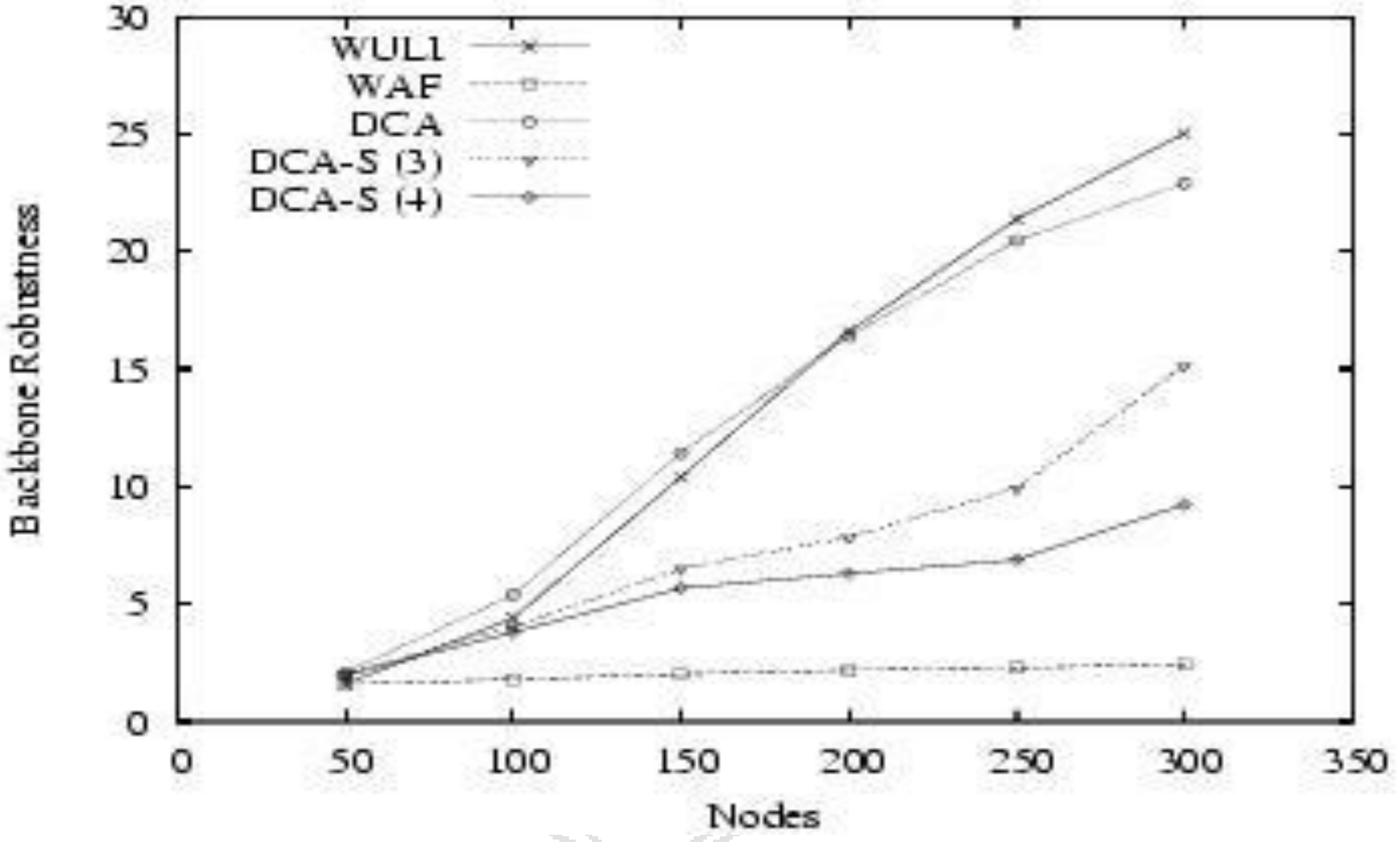
Backbone Size, 2



Backbone Robustness

- Number of nodes needed to disconnect the backbone
- Useful for planning backbone re-orgs
- Increases with network density
- WuLi and DCA: More robust
  - Resilient to up to 25 "death" when n = 300
- WAF: Quite a disaster (tree-like topologies)
- DCA-S: In the middle

Backbone Robustness, 2

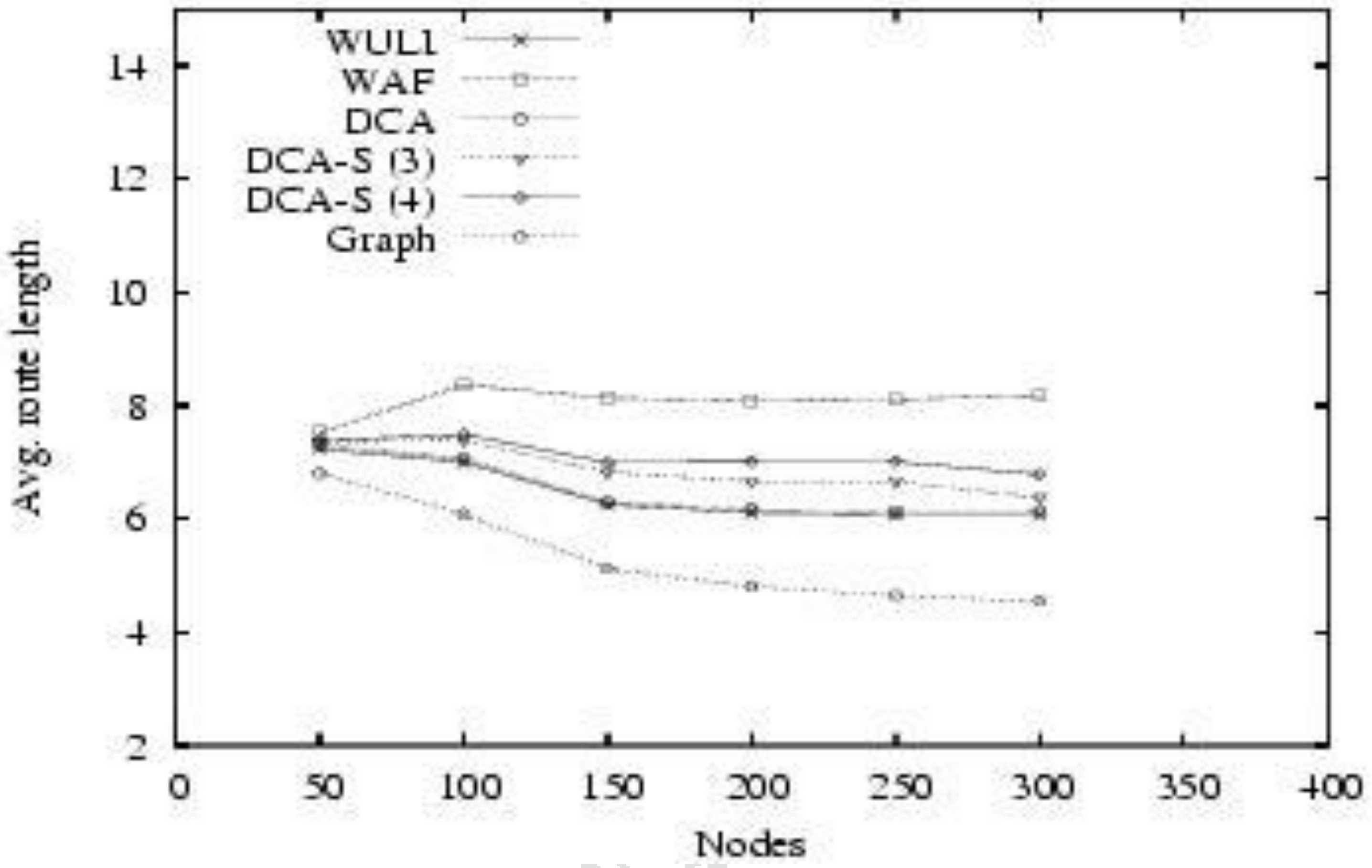


Route Length

- Flat topology ("visibility graph") as a base
- Expected increase: Hierarchy routes are longer
- DCA & WuLi: 7 to 34.7% longer routes
- DCA-S: Up to 9% more than DCA
- WAF: Up to 33.4% longer than DCA

SAPIENZA UNIVERSITÀ DI ROMA

Route Length, 2



"To Go"

- Hierarchical organization is effective for prolonging network lifetime
- Four protocols for backbone formation:
  - DCA, WuLi, WAF and DCA-S
- Nice theoretical features → hard to implement
- Simple solutions (WuLi, DCA): Good starting point for efficient implementations
- DCA-S: "Slimmer" backbone at a reasonable cost