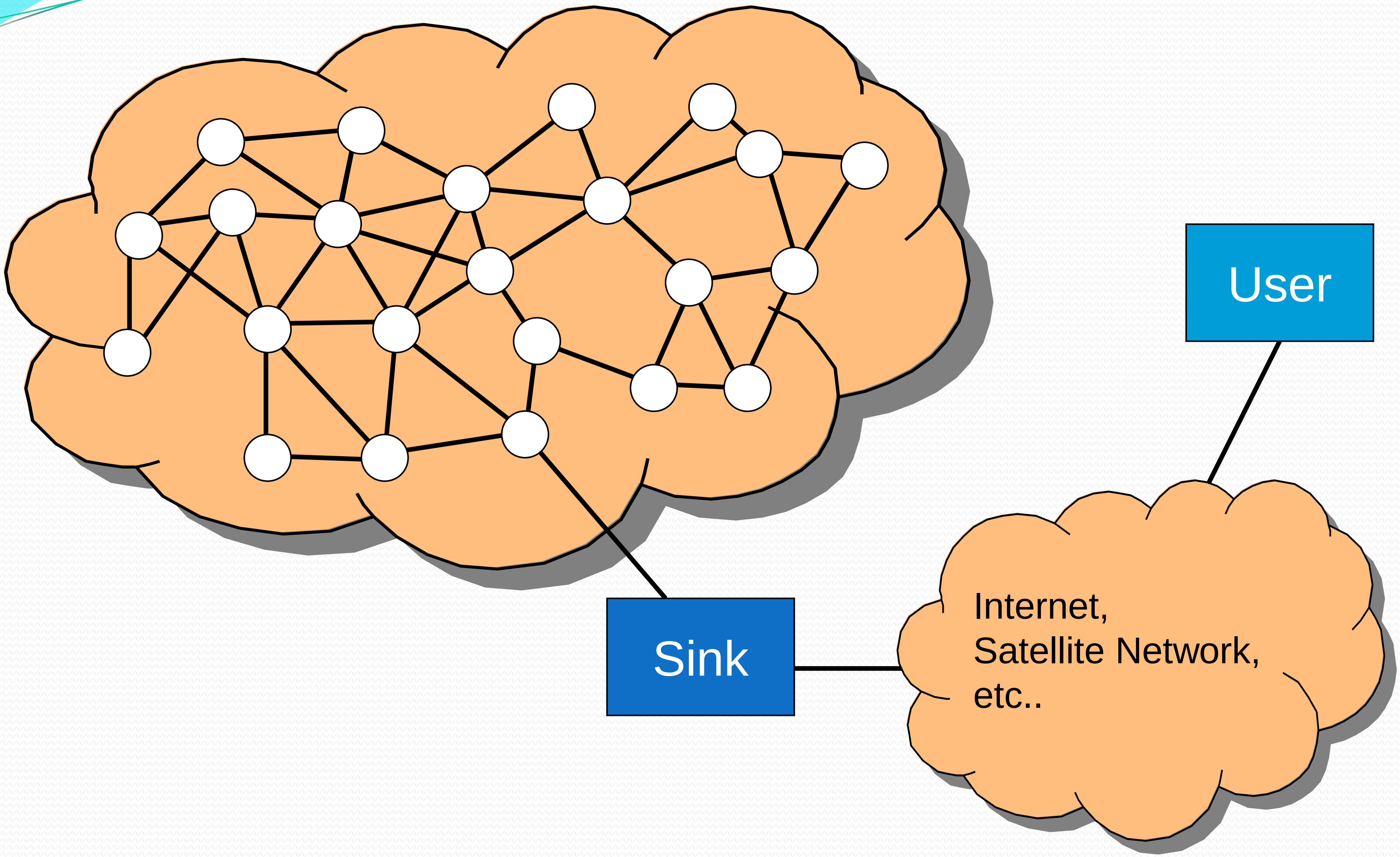
Corso di Reti mobili

Reti ad hoc & Reti di Sensori

Stefano Chessa

Wireless Sensor Networks Issues

WSN: a typical configuration…



… where

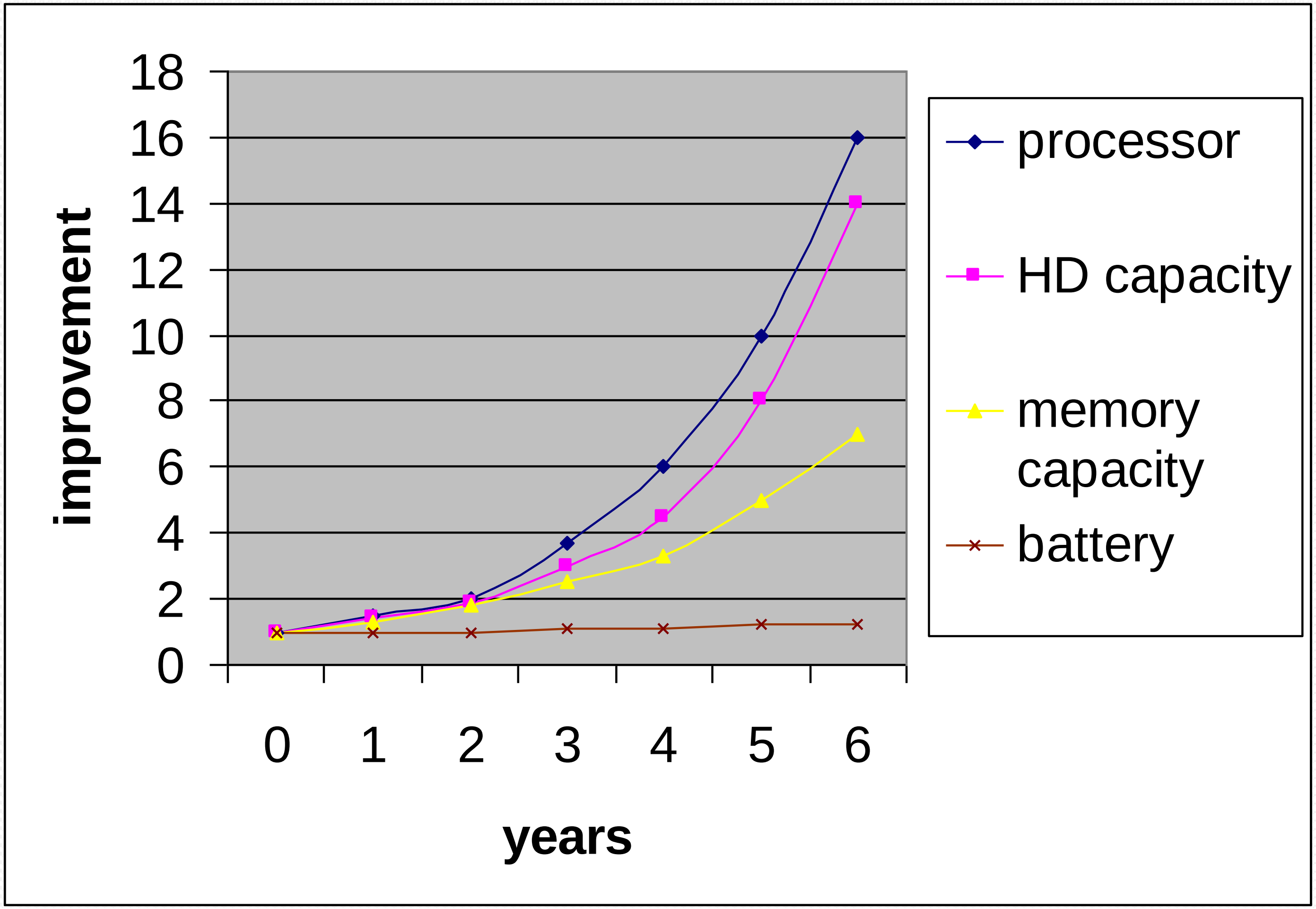
- Each sensor :
  - Low power, low cost system
  - Small
  - Autonomous
- Sensors equipped with:
  - Processor
  - Memory
  - Radio Transceiver
  - Sensing devices
    - Acceleration, pressure, humidity, light, acoustic, temperature, GPS, magnetic, …
  - Battery, solar cells, …

Issues in WSN architecture design

- Sensors are battery-powered
  - Need for HW/SW energy efficient solutions
- Multihop communications
  - Need for protocol stacks
  - Mobility should be taken into account only in some scenarios
  - Constraints on energy, memory, and processor capacity limit the protocols complexity
  - Need for dynamic network management &programming
- Demand for security
  - Constraints on energy, memory, and processor capacity limit the complexity of security protocols

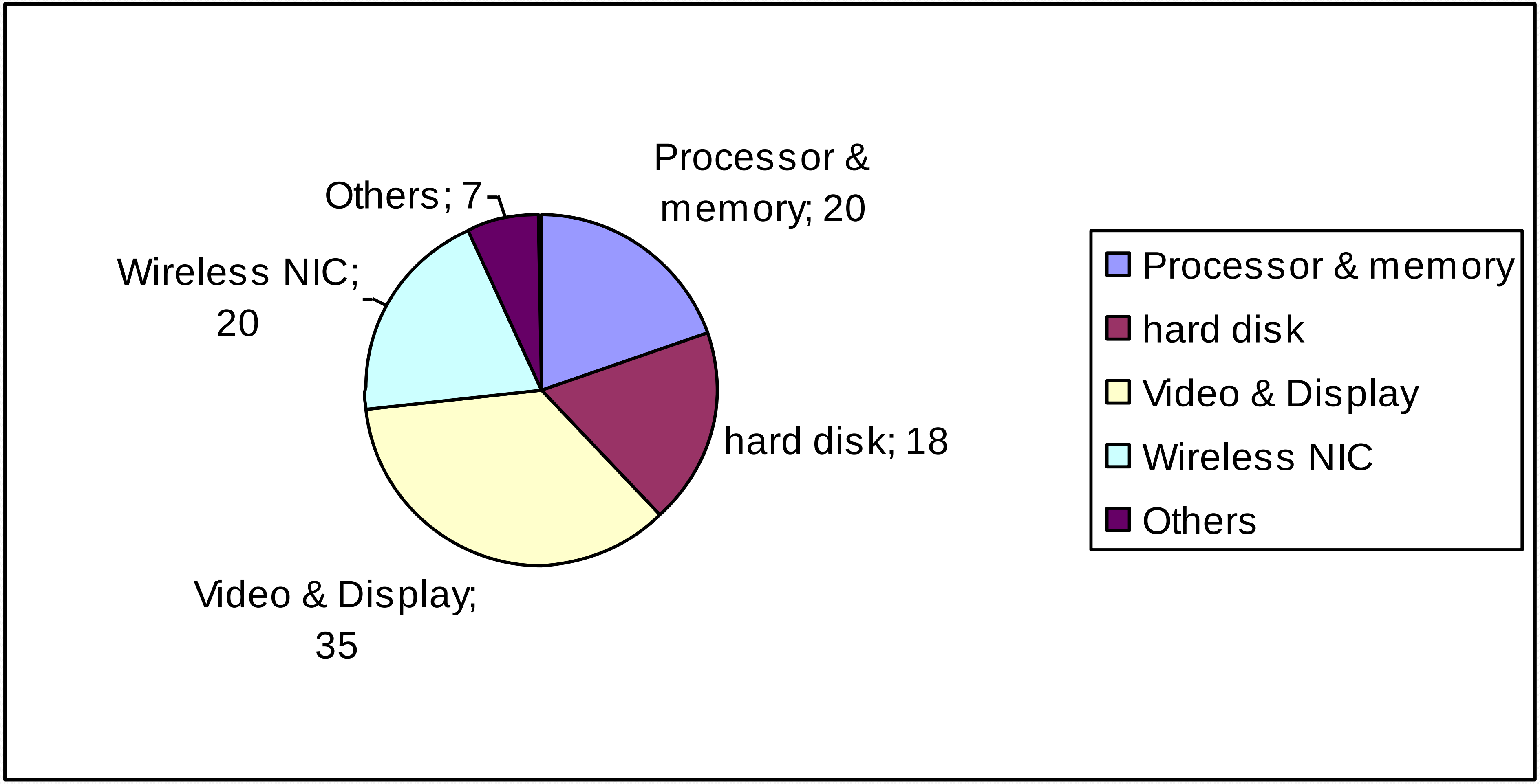
Energy efficiency issues

Intel VS Duracell



Energy efficiency issues

Sources of power consumption



Energy efficiency issues

Example of energy consumption of a Wireless NIC

- Sleep mode: 10mA
- Listen mode: 180mA
- Receive mode: 200 mA
- Transmit mode: 280 mA

Energy efficiency issues

Energy consumption of a sensor (Mote-clone)

- Sleep mode: 0.016 mW
- Listen mode: 12.36 mW
- Receive mode: 12.50 mW
- Transmit mode
  - 0.1 power level, 19.2kbps: 12.36 mW
  - 0.4 power level, 19.2kbps: 15.54 mW
  - 0.7 power level, 19.2kbps: 17.76 mW

Energy efficiency issues

Energy consumption of a sensor (Mote-clone)

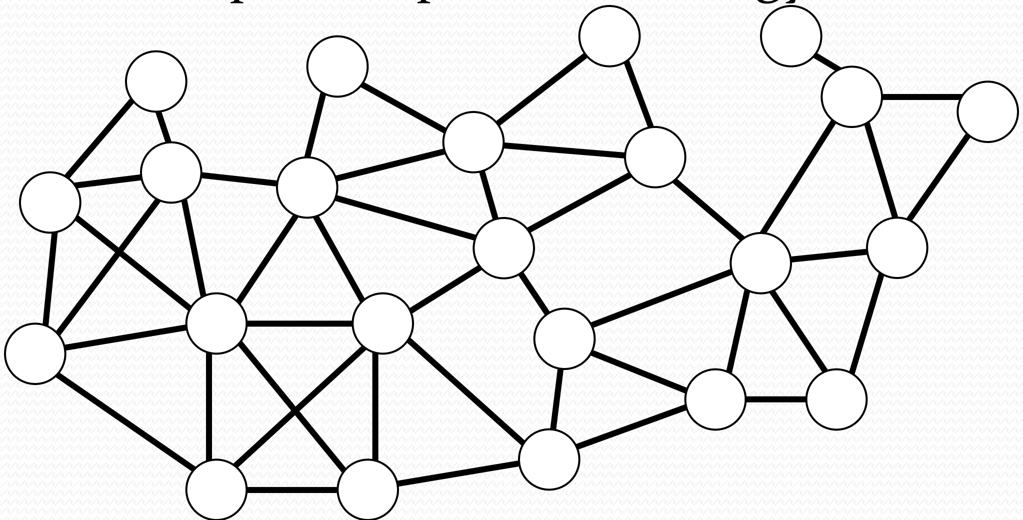
- In some cases transmit power < receive power!
- listen power ≈ receive power
- Radio should be turned off as much as possible
  - Processor power around 30-50% of total power
  - Processor as well should be turned off!
- Turning on and off processor and radio consumes power as well

MAC Protocols

- Low-level communication protocols
  - Basically send/receive packets to/from in-range sensors
- In conventional networks MAC protocols umpire the shared communication channel
- In WSN they also implement strategies for energy efficiency
  - Sinchronize the sensors
  - Turn off the radio when it is not needed
    - Turning off the radio means excluding a sensor from the network

Network protocols

- The network topology is a graph
  - The communications between a pair of sensors should be supported by intermediate sensors
  - The network topology may change due to mobility or failures
- The network protocols construct for paths connecting arbitrary pair of sensors
  - Energy efficiency is important
  - Synchronization of the sensors in a path may save energy
  - Need for cross-layer solutions to optimize paths and energy



“Application” Issues

- Management of the sensor network:
  - The network is static, no new nodes
  - The network is dynamic, nodes may join and leave
  - Nodes may offer services
- Sensor network programming
  - Static vs dynamic

Security

- The main requirements are :
  - Confidentiality
    - Radio communication can be easily eavesdropped
  - Authenticity
    - External entities may inject forged packets
  - Integrity
    - The packets should not be damaged/altered by errors or external entities
  - Data freshness
    - Ensures that the received packet is recent (fresh), and it is not, for example, a replica of an old packet
- The main difficulty is in the limited processing capacity of the sensors
  - Symmetric encryption is often the preferred solution

WSN Design

Many limitations in the WSN design are related to processing, memory, and communication constraints

The evolution of HW technologies may overcome these constraints (??)

The Moore's law and WSN

"The number of transistors that can be (inexpensively) embedded in a chip grows exponentially"

(it doubles every two years)

The Moore's law and WSN

The Moore's law offers three different interpretations:

- The performances double every two years at the same cost
  - Up to now this is true for processors of servers/desktops
- The chip's size halves every two years at the same cost
  - Consequently also the energy consumption halves
- The size and the processing power remain the same but the cost halves every two years

The Moore’s law and WSN

In the case of WSN all the three interpretation are valid

There are applications that:

- Require small-sized sensors and/or that have low power consumption
- Require higher processing capabilities to the single sensor

The cost is important in (almost) all applications

The Moore's law and WSN

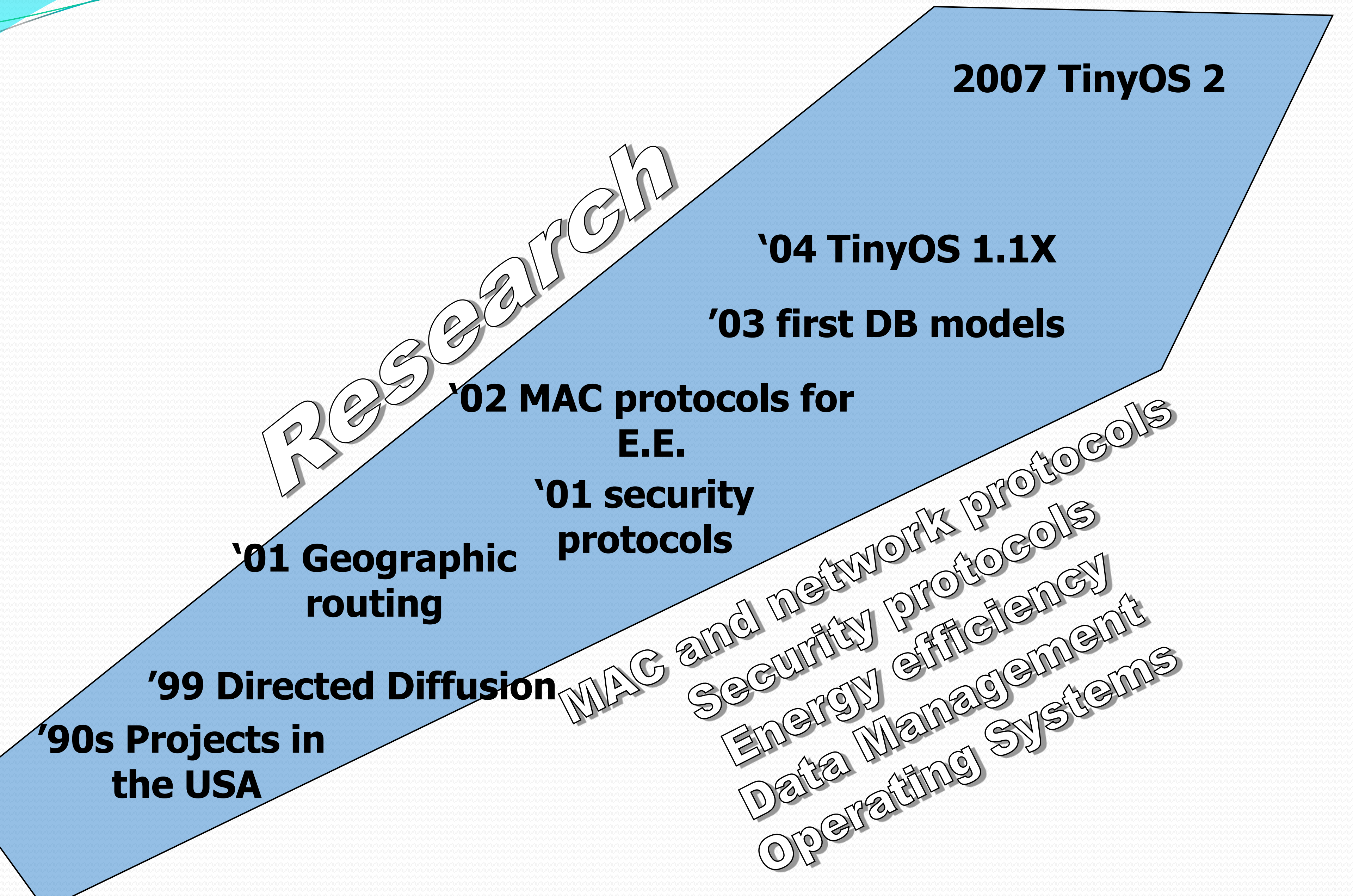
- Nowadays there exist several sensors with different capabilities in terms of processing and energy consumption
- Differently than server/desktop applications the sensors use low-power, cheap processors that are still on the market
- It is normally important to use the cheapest HW that can sustain the WSN application
  - considering the scale factor due to the large number of sensors this have considerable effects on the final costs

A brief history of research and development on WSN

“Milestones”

- The concept of “wireless sensor network” was introduced by some USA projects at the end of the 90’s
- In ’99 appeared the first scientific papers on WSN
- In 2001 appeared the first industrial prototypes
- In late 2003 appeared the standards IEEE 802.15.14 and ZigBee

WSN and research



WSN and research

- 2000-2003 Definition of the main models
  - Energy efficiency:
    - MAC-level synchronization
    - Topology control
  - Routing protocols
    - Critic to the protocols for ad hoc networks
    - Routing on trees
    - Geographic routing
  - Paradigms for the query of the network and for data gathering
    - Idea of network query
    - Database models
    - Data centric storage and geographic hash tables

WSN and research

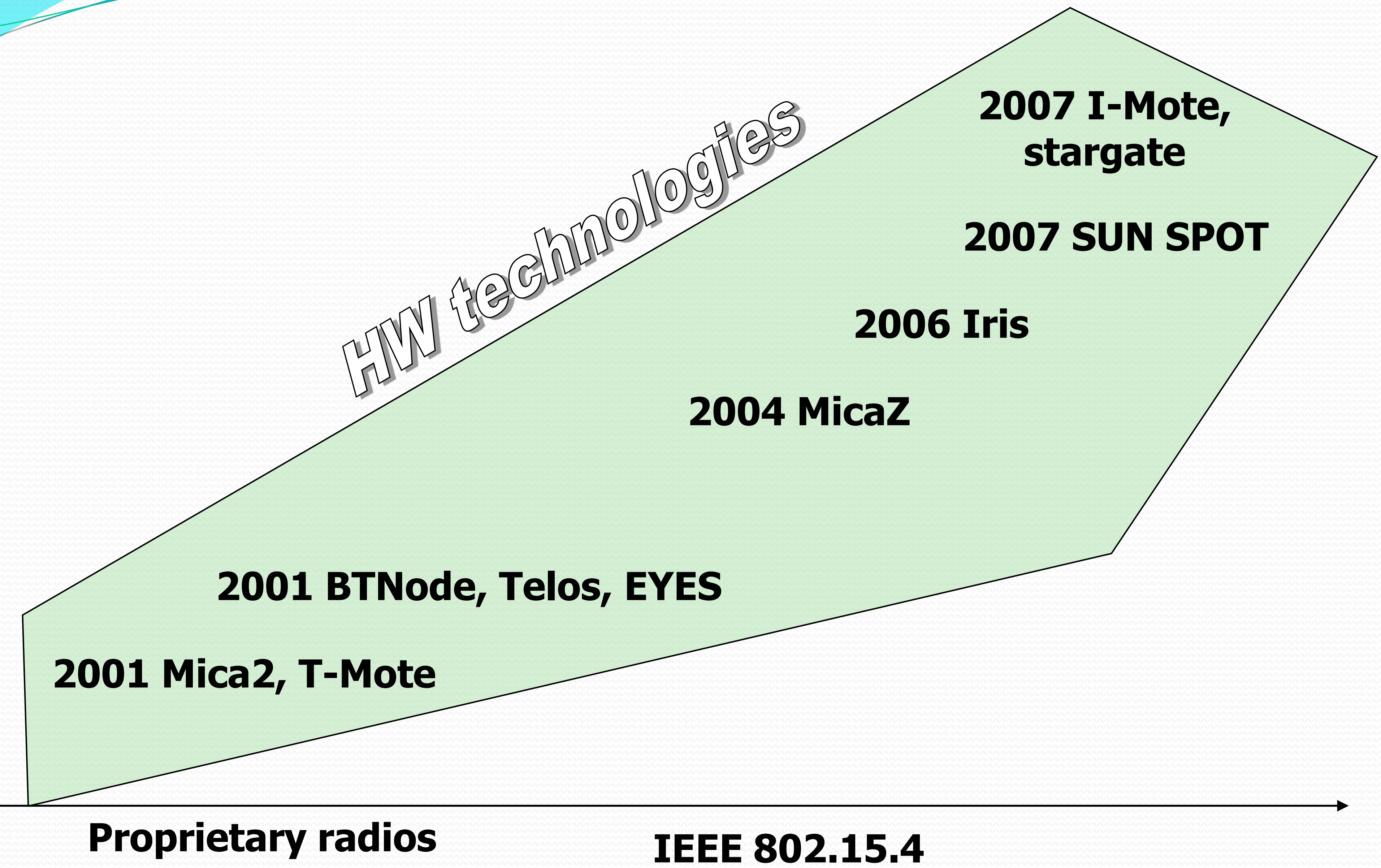
- 2000-2003 Definition of the main models
  - Operating systems
    - TinyOS is the first
    - And then Contiki, SOS, …
  - Middleware for network management
  - Security protocols
    - Use of symmetric keys
    - Issues related to key management
- 2003-today
  - Effort to improve the models and theories introduced in the previous years
  - Necessity for a middleware for the interaction with access networks

WSN and research

Currently the research programs of the EU invest in the use of WSN as an enabling technology for context-aware systems

- Systems that can interpret the context information obtained from heterogeneous sources
- The main applications are about:
  - Advanced multimedia systems
    - Relationship with domotics
  - Support to elders and disabled
  - Remote monitoring of patients and telemedicine
    - Monitoring of physiological parameters
    - Support to the correct use of medicines
    - …
  - Pervasive systems
    - Users guidance in public buildings (airports, hospitals, museums..)
    - …
  - Automotive
    - Management of the sensors on board
    - Integration with environmental sensor networks
    - …

WSN and HW developement



WSN and HW developement

- in 2000
  - 8 bit processors, proprietary radios
  - Gateways on serial lines (RS232)
- 2003
  - Standard radios
- 2007
  - 32 bit processors
  - Gateway with WiFi etc…

WSN and standards

Standards

2008: Texas Instrument's Z-Stack

2007 6LowPan

2007: Texas Instrument's SimpliciTi

2006: revision of IEEE 802.15.4 e ZigBee

End of 2003 IEEE 802.15.4 e ZigBee

WSN and standards

- 2003
  - Physical and MAC layer standardization (IEEE 802.15.4)
  - Network, transport, and application layers standardization (ZigBee)
- 2006
  - Second release of standards IEEE 802.15.4 and ZigBee
- 2007
  - Alternative middleware
    - Lighter middleware or
    - IPV6 compatible (6LowPan)

Sensor Networks Hardware Platforms

HW platforms

- Different trends:
  - Commercial platforms to be assembled in microsystems tailored to specific applications
    - ATMEL 128 / 256 /… + CC 2420 (IEEE 802.15.4)
    - TI MPS 430 … + CC 2420
    - XScale + CC 2420
    - ARM + CC 2420
  - “general purpose”, application-ready microsystems
    - Already embed transducers
    - The transducer set can be tailored to a specific application

Mica Motes

- HW platform widely used in the academy
- Produced in the USA
- A family of products based on IEE 802.15.4
- Microsistems ready to use
- Customizable set of transducers

MicaZ-class WSN hardware

	Btnode 3	mica2	mica2dot	micaz	telos A	tmote	EYES
Manufacturer	Art of Technology	Crossbow			Imote iv		Univ. of Twente
Microcontroller	Atmel Atmega 128L				Texas Instruments MSP430		
Clock freq.	7.37 Mhz		4 MHz	7.37 MHz	8 MHz		5 MHz
RAM (KB)	64 + 180	4	4	4	2	10	2
ROM (KB)	128	128	128	128	60	48	60
Storage (KB)	4	512	512	512	256	1024	4
Radio	Chipcon CC1000 315/433/868/916 MHz 38.4 Kbauds			Chipcon CC2420 2.4 GHz 250Kbps IEEE 802.15.4			RFM TR1001868 MHz 57.6 Kbps
Max Range (m)	150-300			75-100			
Power	2 AA batteries		Coin cell	2 AA Batteries			
PC connector	Through PC-connected programming board				USB		Serial Port
OS	Nut/OS	TinyOS					PEEROS
Transducers	On acquisition board				On board		On acquisition board
Extras	Bluetooth radio						

Mica Motes

a)

Mica Z

Iris

b)

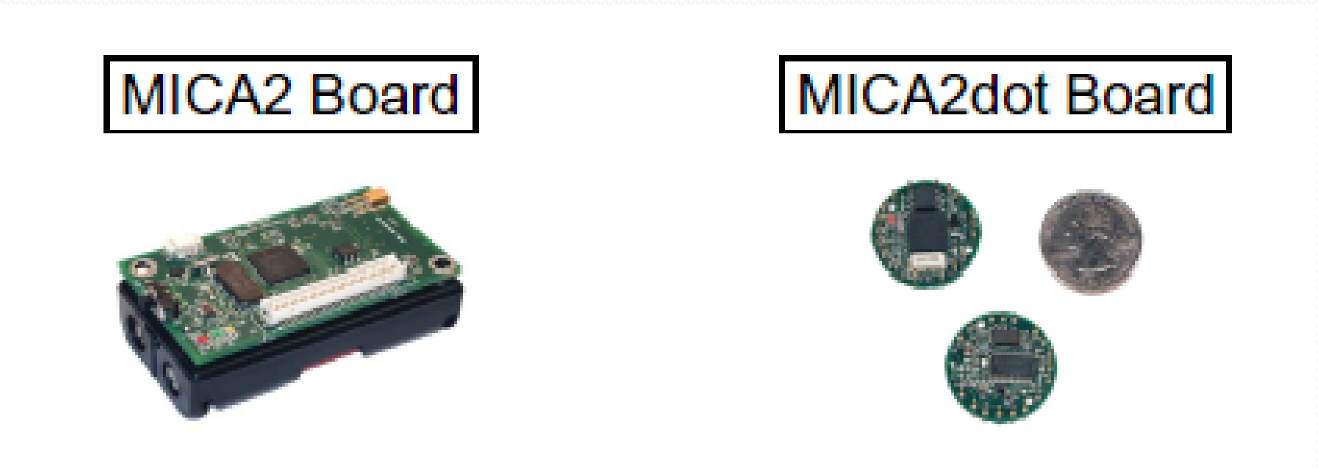
Cricket

c)

Sensor network hardware

The Mica2/MicaZ platform:

- Low power CPU
  - ATMEL 128L (8 bit, 8Mhz)
- Program memory: 128 KB Flash memory
- Data memory: 4 KB RAM – 512 KB Flash memory



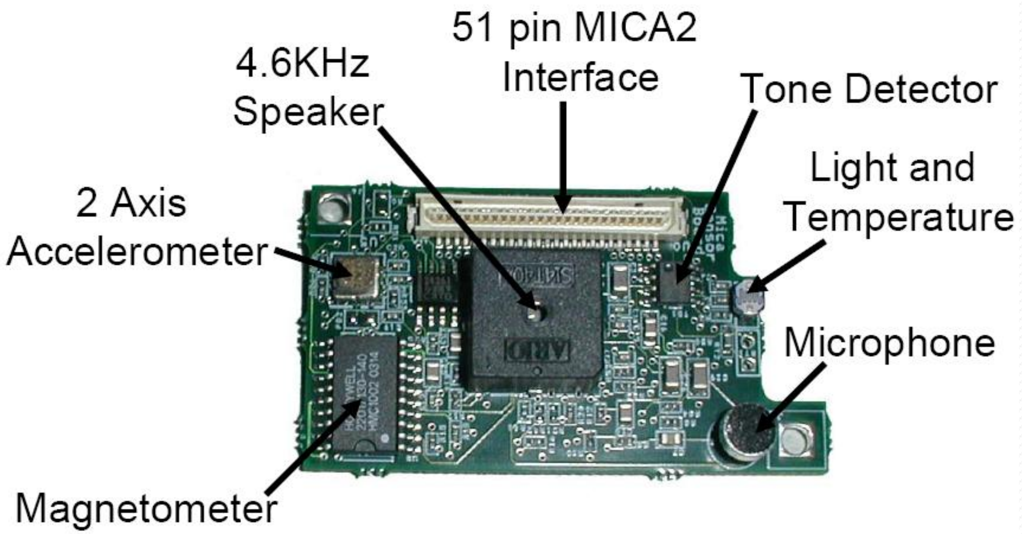
Mica Motes

Mica2/MicaZ/Iris:

- Low-power CPU
  - ATMEL 128L (8 bit, 8Mhz)
- Program memory: 128 KB Flash memory
- Data memory: 4 KB RAM – 512 KB Flash memory
- Radio compatible with IEEE 802.15.4
  - 2,4 GHz, 250 Kbps
  - Communication range: up to 100 m. (in open fields)
- Battery pack: 2 AA 1,5 V batteries
- Transducers on a separate board
  - Several transducer boards are available
- The cost of a professional kit with 6 MicaZ is about 3000 $

Mica Motes: transducer board

- Example: MTS 300 CA
  - Light
  - Temperature
  - Microphone
  - Sounder
  - Accelerometer 2 axis
  - Magnetometer 2 axis
- Other boards include:
  - GPS
  - Humidity
  - Pressure
  - Additional analog and digital inputs



Mica Motes: sink

- Several types of sinks:
  - Boards connecting a sensor to a PC through a serial line (USB, ethernet)
  - Microsystems (stargate) acting as bridges between a IEEE 802.15.4 network and ethernet, wifi,…

Mica Motes: sink

- Stargate
  - Intel PXA255 Xscale 400 Mhz
  - Linux embedded
  - WiFi, ethernet, IEEE 802.15.4 interfaces
  - Hosts a Mica Mote

Mica Motes: IMote2

- High performance, low consumption CPU
  - Marvell PXA271 XScale Processor
  - 13MHz to 416MHz with Dynamic Voltage Scaling
  - 256kB SRAM, 32MB SDRAM and 32MB of FLASH memory
  - XScale DSP
- Designed for multimedia applications (control of cameras,…)
- Radio compatible with IEEE 802.15
  - 2,4 GHz, 250 Kbps
  - Range: up to 100 m.
  - Interoperability with Mica Motes

Mica Motes: IMote2

- Trasduttori on a separate board
  - Boards with different transducers are available
- Sensor board (ITS400CA):
  - Accelerometer 3 axis
  - Temperature and humidity
  - Light
  - ADC "general purpose"
- Battery pack: 3 x AAA 1,5 V
- Cost of a basic kit with 3 sensors: about

SUN Spot

- Produced by SUN
- Based on Java
  - Supports a Java virtual machine
- Currently distributed to research purposes
- High performance, low-power CPU
  - Marvell PXA271 XScale Processor
  - 13MHz to 416MHz with Dynamic Voltage Scaling
  - 256kB SRAM, 32MB SDRAM and 32MB of FLASH memory
  - XScale DSP
- Radio compatible with IEEE 802.15.4
  - 2,4 GHz, 250 Kbps
  - Range: up to 100 m.

SUN Spot

- Transducers and additional inputs on a separate board :
  - 2G/6G accelerometer 3-axis
  - Temperature, Light
  - 8 tri-color LEDs
  - 6 analog inputs
  - 5 I/O general purpose pins
- Battery pack: 3 x AAA 1,5 V
- Not clear the business model
  - Mainly for the show?

Protocols for Sensor Networks

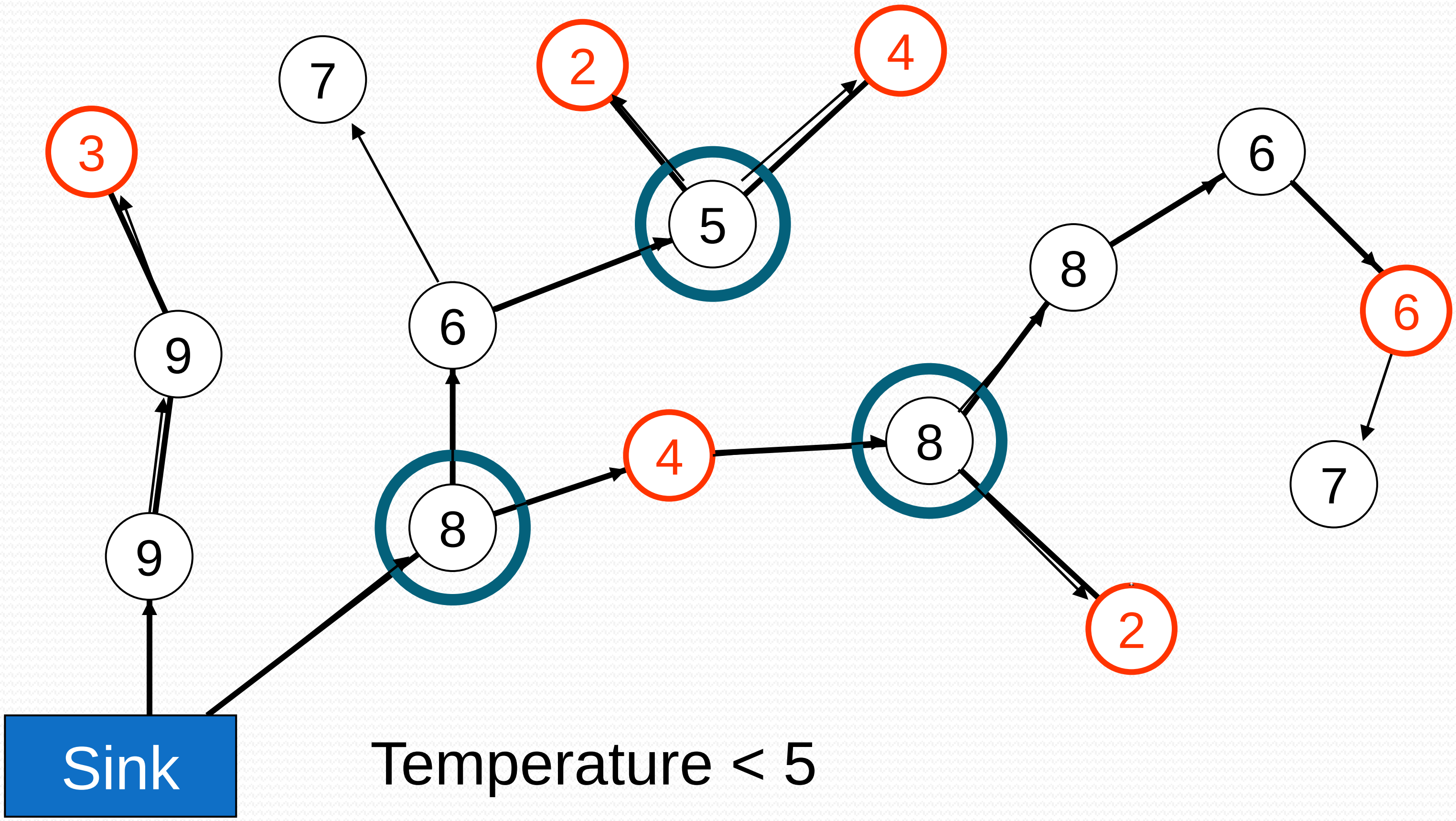
WSN: data centric vs node centric

- Important considerations:
  - Sensor networks are mostly data centric
  - Attribute-based addressing and location awareness
  - Data aggregation can be useful but it might prevent collaborative effort
  - Energy efficiency is a key factor
- Traditional routing protocols are not practical:
  - Large routing tables
  - Size of packet headers
- Node IDs are less meaningful than their capabilities
  - From identity-based to data-driven routing

Protocols for Sensor Networks

Data centric routing

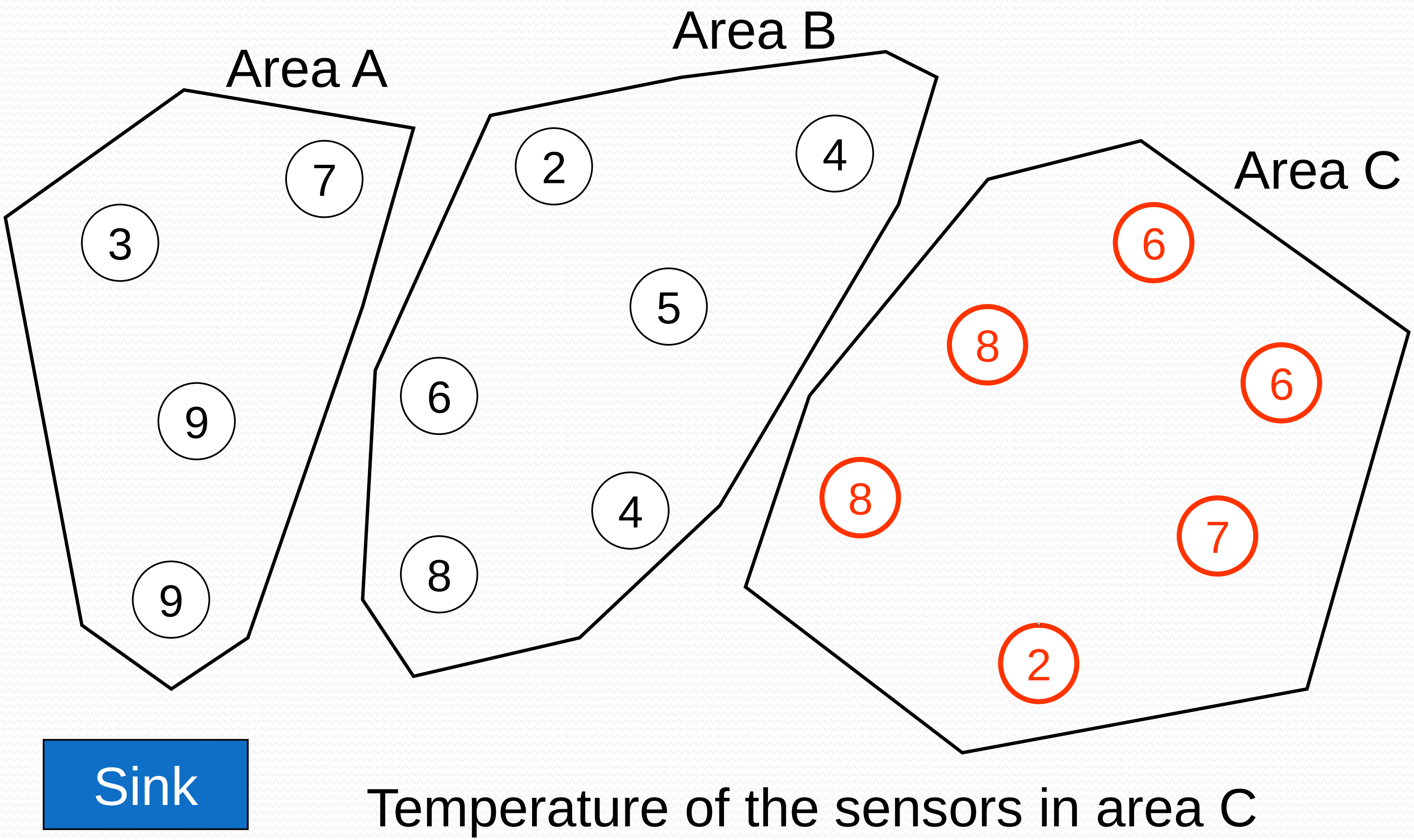
Aggregation:



Temperature < 5

Protocols for Sensor Networks

Location Awareness

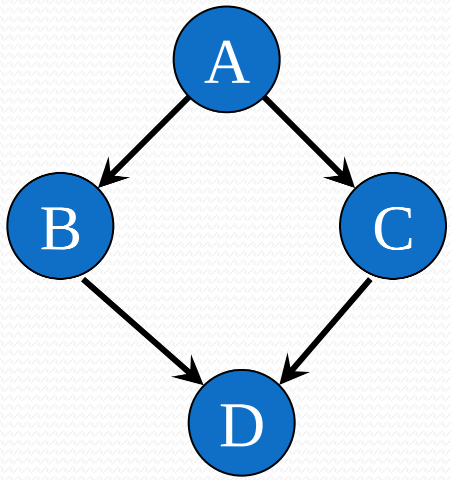


Temperature of the sensors in area C

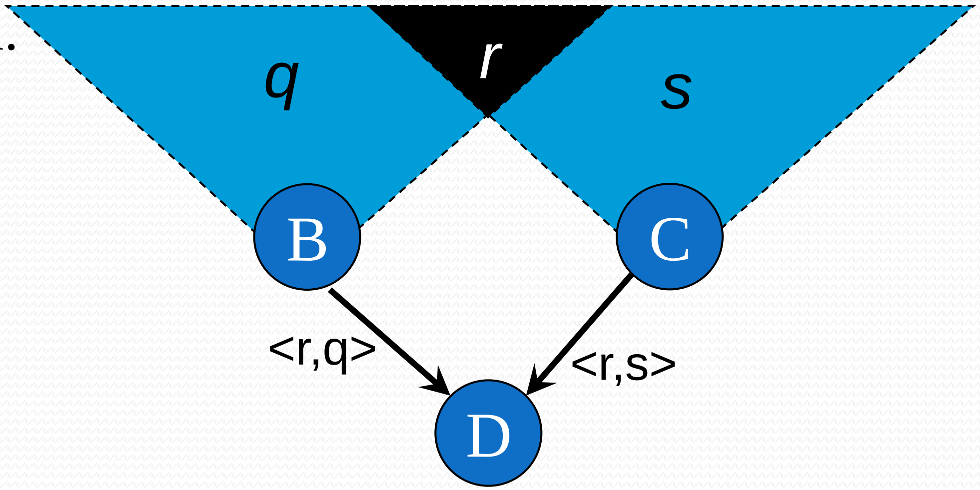
Protocols for Sensor Networks

Drawbacks of flooding-based data dissemination:

- The implosion problem:
  - node A starts by flooding its data to all of its neighbors.
  - Two copies of the data eventually arrive at node D.
  - The system wastes resources in one unnecessary send and receive.



- The overlap problem:
  - Two sensors cover an overlapping geographic region.
  - The sensors flood their data
  - Node C receives two copies of the data marked r.



MAC Protocols

Design guidelines

- MAC layer for WSN should also implement energy efficiency strategies
- The objectives is to:
  - Reduce the radio duty cycle
  - Maintain network connectivity
- Tradeoffs energy vs latency & bandwidth
- Three approaches to energy efficiency:
  - Synchronization of nodes (e.g. S-MAC, IEEE 802.15.4)
  - Preamble sampling (e.g. B-MAC)
  - Polling (e.g. IEEE 802.15.4)

Design guidelines

- Synchronization of the nodes:
  - If the nodes are synchronized they can turn on the radios simultaneously.
  - When the radios are active the network is connected
  - When the radios are inactive there is no network
  - The radios have a low duty cycle: inactive for most of the time
- Who decides the duty cycle?
- How does this affects the latency?

Synchronization: S-MAC

- Medium access control for sensor network
  - Implemented over TinyOS and mica motes
- Exploits nodes synchronization
  - Under this respect it is also a network organization protocol
  - Only local synchronization, NOT global
  - Nodes Alternate listen and sleep periods
    - During sleep time the sensor cannot detect incoming messages

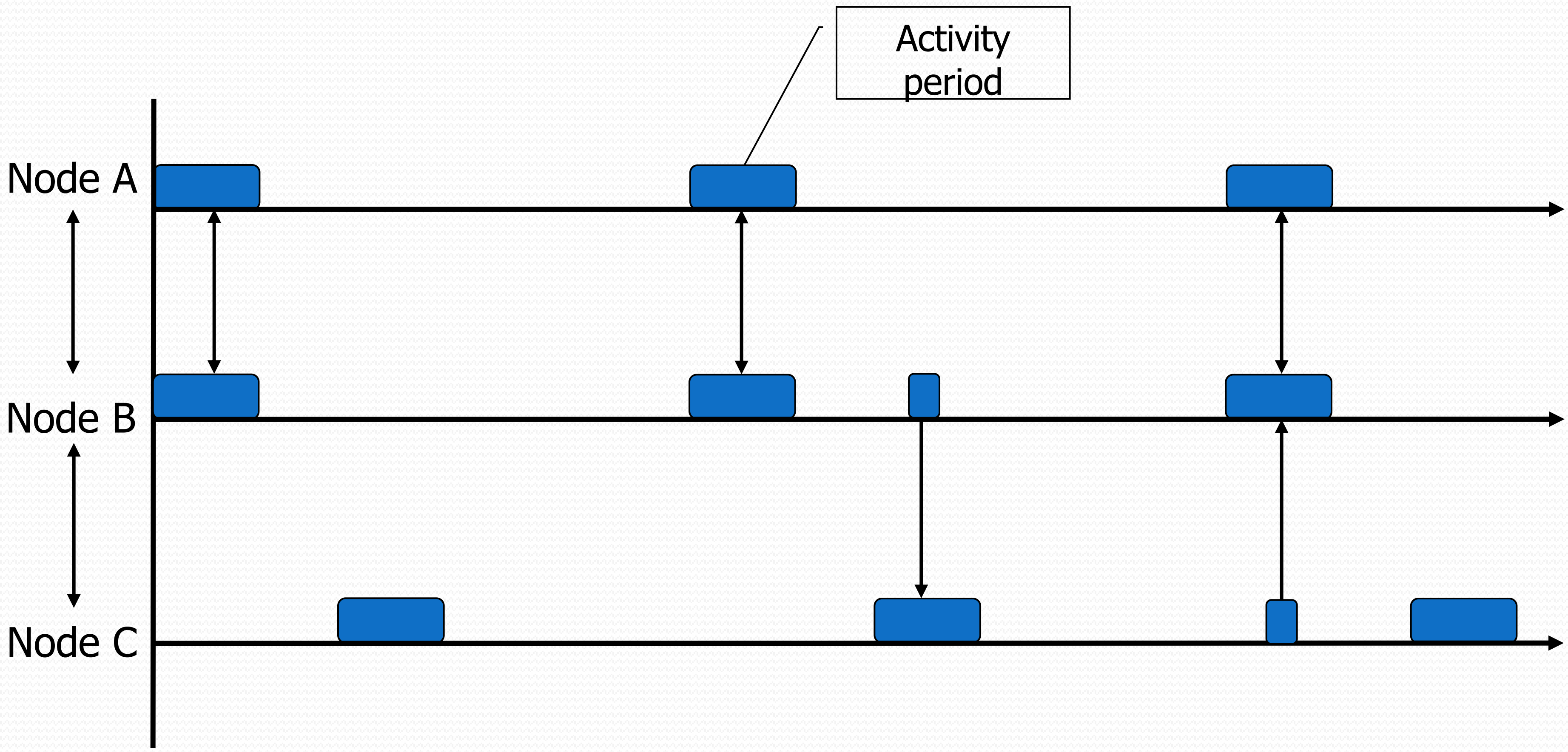
Synchronization: S-MAC

- Adjacent sensors synchronize the listen periods
  - By means of periodical (local) broadcasts of SYNC packets
    - A SYNC packet contains the schedule (sleep/wakeup periods) of the sensor
  - If a node detects adjacent sensors with pre-defined listen period it use the same period
  - Otherwise it chooses its own period
    - The chosen period is advertised to the neighbors by SYNC packets
  - A sensor may revert to someone else's schedule if its own schedule is not shared with anybody else.

Synchronization: S-MAC

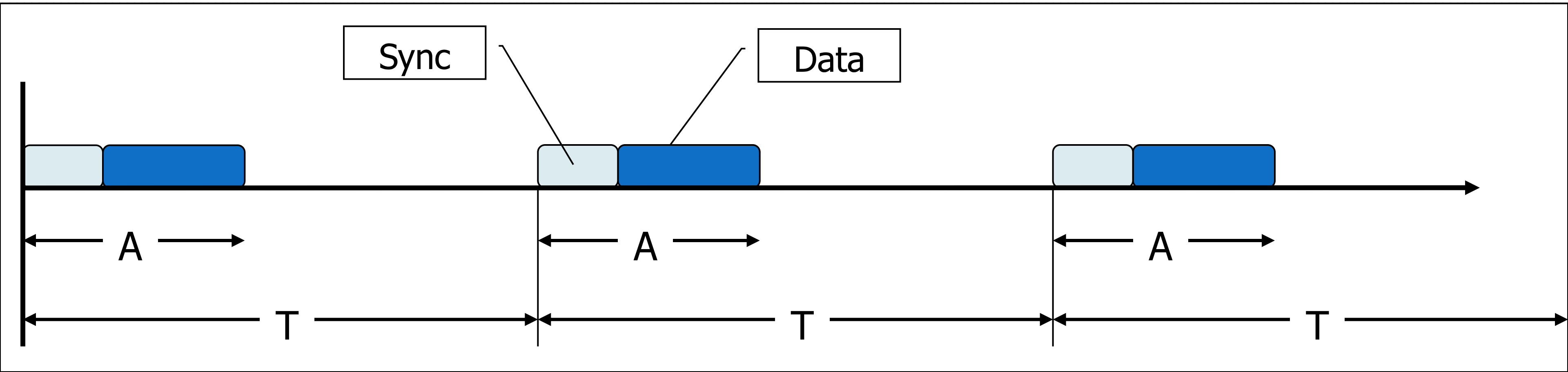
- A sensor receives packets from the neighbors during its listen period
- A sensor A can send a packet to sensor B only during the listen period of B
  - Sensor A may need to turn on its radio also outside its listen period
  - Sensor A should know the listen period of all of its neighbors
    - It listens the SYNC packet of its neighbors once it is turned on

Synchronization: S-MAC



Synchronization: S-MAC

- Packets are sent during the listen period of the receiver
  - Carrier sense before transmission
  - If the channel is busy and a node fails to get the medium, the packet is delayed to the next period
  - Collision avoidance based on RTS/CTS



Synchronization: S-MAC

- Issues:
  - Latency
    - To be sent across a multihop path a packet may have to wait (in the worst case) for the listen period of each intermediate node
    - It is mitigated by the fact that (hopefully) a number of sensor will converge towards the same schedule (not guaranteed anyway)
  - Maintain synchronization
    - Clock drifts may affect synch.
    - Depending on the topology it may be impossible for a sensor to have a listen period compatible with its neighbors
    - Need for protocols to maintain schedules

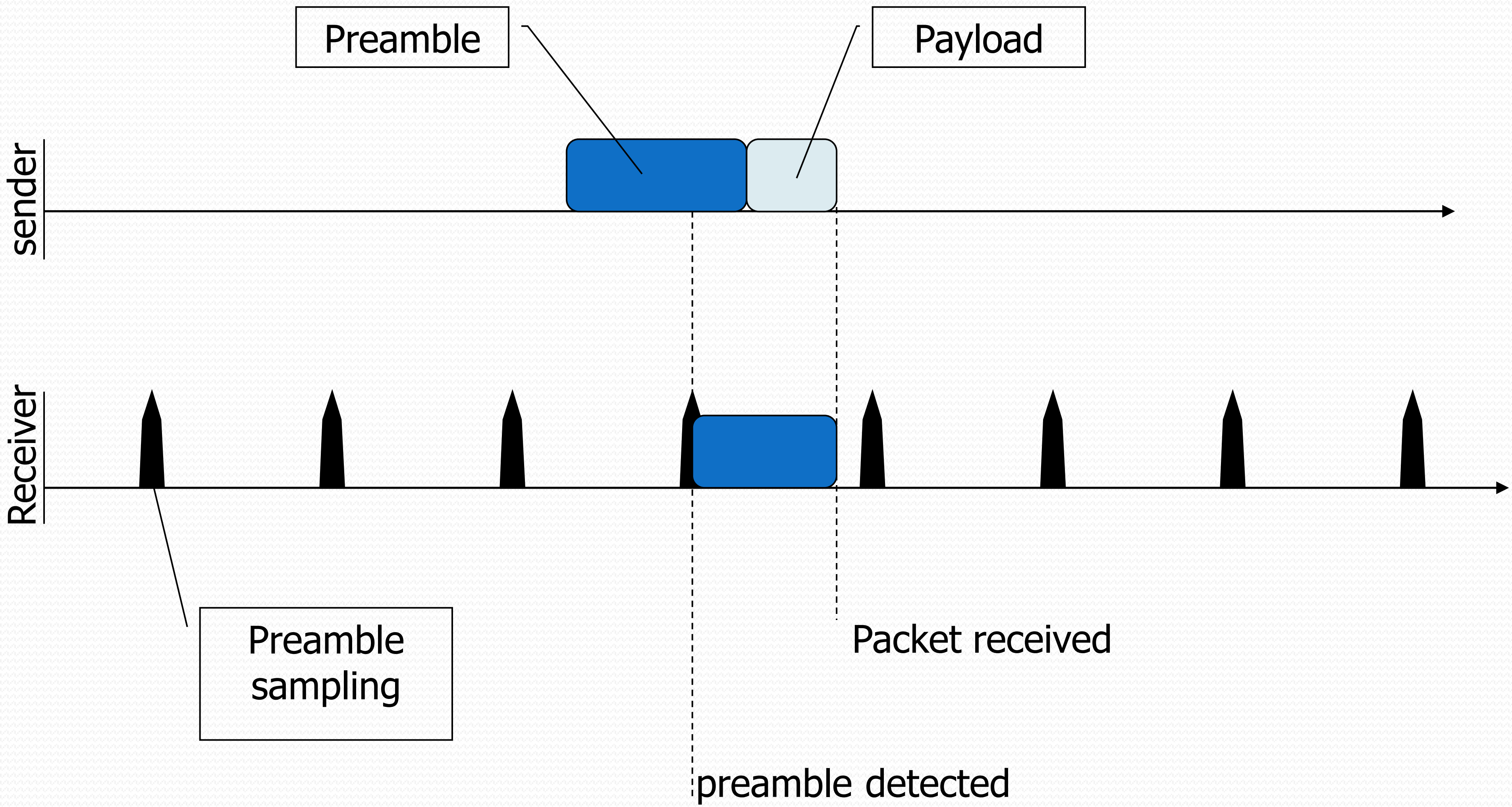
Preamble sampling: B-MAC

- Medium access control for sensor network
  - Implemented over TinyOS and mica motes
  - It does not exploit sensors' synchronization
- A sender sends whenever it wants
  - The sent packet contains a very long preamble in its header
- The receiver activates its radio periodically to check if there is a preamble "on the air"
  - This activity is called preamble sampling

Preamble sampling: B-MAC

- If the preamble sampling detects a preamble:
  - keep the radio on to receive the packet
  - Otherwise: turn off the radio
- The idea is:
  - Spend more in transmission but save energy in reception
  - The preamble sampling should be very short and cheap
  - the cost of radio activations/deactivation on the receiver side are amortized by lower rates of sampling
- To work properly the preamble should be longer than the sleep period

Preamble sampling: B-MAC



Preamble sampling: B-MAC

- Advantages:
  - It is not a network organization protocol
  - It is simple to use and configure
    - In practice it is transparent to the higher layers
- Issues:
  - In the long run preamble sampling is not negligible
  - In some cases it may result more expensive than using some form of synchronization

Polling

- It is a technique that can be combined with synchronization
  - Used by IEEE 802.15.4
- Requires an asymmetric organization of the nodes:
  - A master node that issues periodic beacons
  - Slave nodes that can keep the radio off whenever they want.
- If a message for node a slave arrives to its master
  - The master stores the message and advertise its presence in the beacon
  - When the slave turns the radio:
    - waits for the beacon
    - recognizes that there is a pending message
    - Requests the pending message to the master

Network protocols: Directed Diffusion

Directed Diffusion

- Intanagonwiwat et Al., MobiCom 2000
- Coordination protocol to perform distributed sensing of environmental phenomena
- The sensor network is programmed to respond to queries such as:
  - "How many pedestrians do you observe in the geographical region X"
  - "Tell me in what direction that vehicle in region Y is moving"
- Directed diffusion is datacentric
  - All communications are for named data
  - Data generated by sensors are named by attribute-value pairs.
  - A node requests data by sending *interests* for named data.

Directed Diffusion

Basic elements of Directed Diffusion:

- Data is *named* using attribute-value pairs.
- A sensing task is disseminated in the network as an *interest* for named data.
- The dissemination of interests sets up *gradients*
  - gradients "draw" events (*i.e.*, **data** matching the interest).
- Data matching the interest flow towards the sink of interest along multiple paths.
- The sink *reinforces* one, or a small number of these paths.

Directed Diffusion

- Interests are named by a sequence of attribute-value pairs that describe the task.
  - Example of a simple animal tracking task:

```
type = four-legged animal          // detect animal location
interval = 20 ms                   // send back events every 20 ms
duration = 10 seconds              // .. for the next 10 seconds
rect = [-100, 100, 200, 400]       // from sensors within rectangle
```

  - Coordinate may refer to a GPS-based coordinate system
- The data sent in response to the interest is also named using a similar naming scheme.
  - Example :

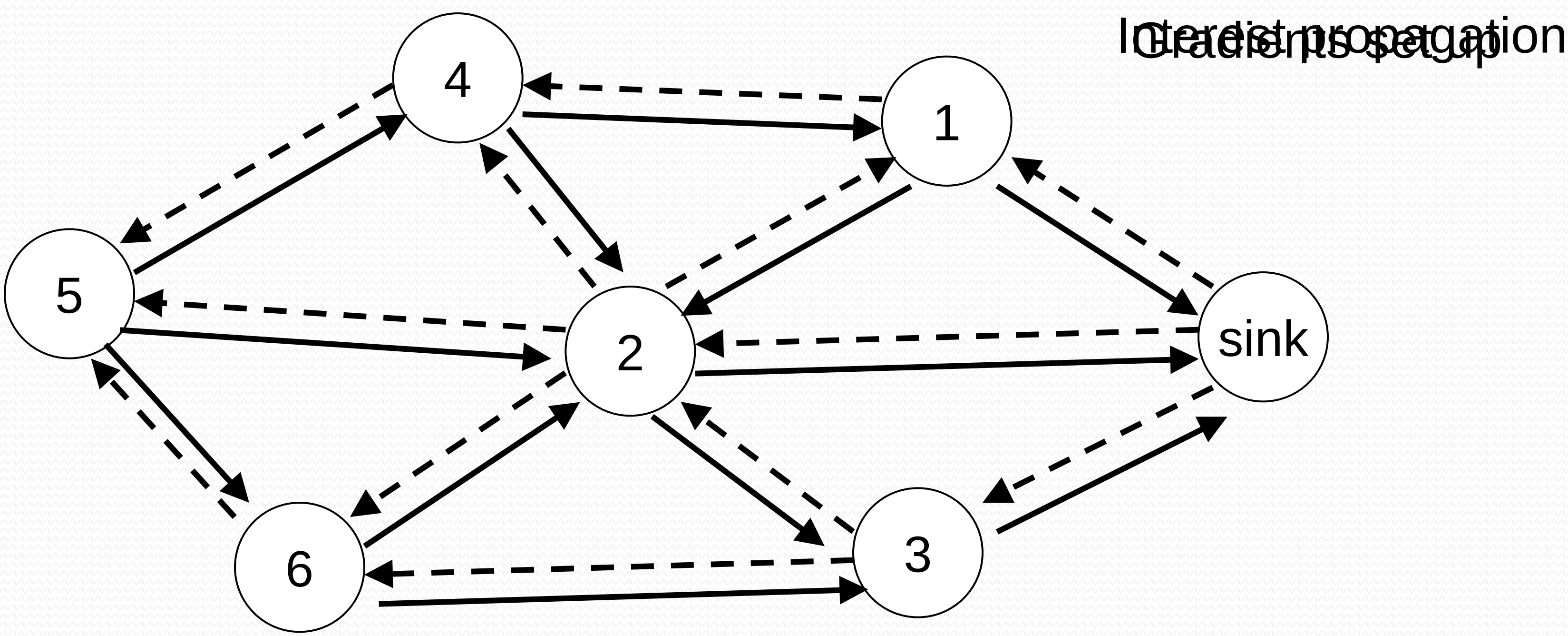
```
type = four-legged animal          // type of animal seen
instance = elephant                // instance of this type
location = [125, 220]              // node location
intensity = 0.6                    // signal amplitude measure
confidence = 0.85                  // confidence in the match
timestamp = 01:20:40               // event generation time
```

Directed Diffusion

- Interests are periodically generated by the sink
  - The first broadcast is **exploratory**
  - The next broadcasts are **refreshes** of the interest
    - Necessary because dissemination of interests is not reliable
  - Nodes receiving an interest may forward the interest to a subset of neighbors
    - nodes must be assigned with a unique ID
- Directed diffusion works also in presence of multiple sinks

Directed Diffusion

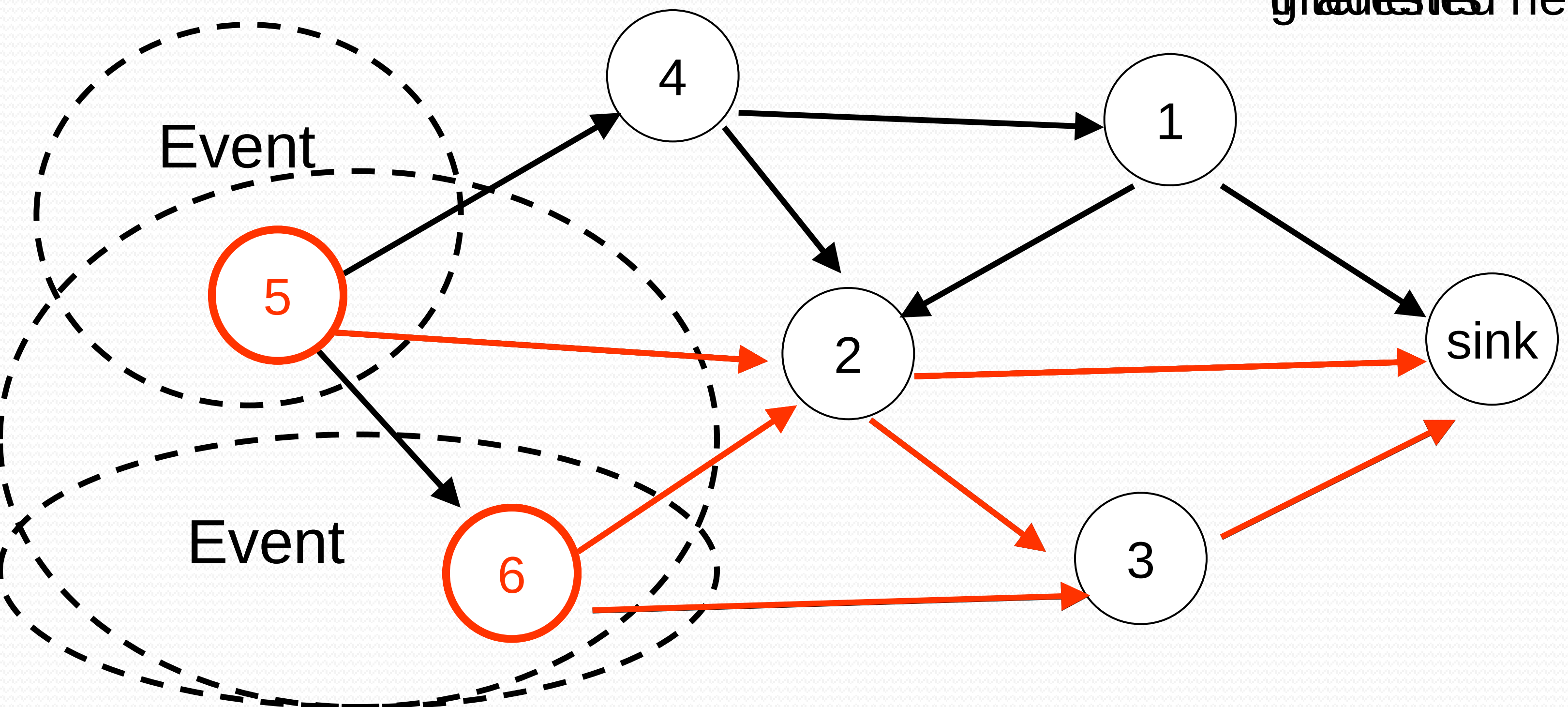
- Nodes cache received interests
  - Different interests with same time interval, area, and type (but, for example, different sampling rate) are aggregated
  - Interests in the cache expire when the duration time is elapsed
  - Each interest in the cache is associated with a **gradient**, i.e. the node from which it was received
    - Gradients might be associated with different sampling rate
  - Note that the same interest may be received from different nodes



Directed Diffusion

- A gradient is a **direction** and a **data rate**
- Gradients are used to route data matching the interest toward the sink whom originated the interest
  - A data may be routed along multiple paths
  - Data is routed along a single path if a preferred gradient is used

Examples of data propagation:



Directed Diffusion

- A sensor node which detects an event matching with an interest in the cache:
  - Start sampling the event at the largest sampling rate of the corresponding gradients
- The node sends sampled data to neighbors interested in the event
  - This information is stored in the gradients associated to the interest in the cache
  - If a gradient g has a lower rate then the others, data along g is sent with lower rate
- Neighbors forward the data only if a corresponding interest (with a gradient) is still in the cache
  - However if that data has already been sent it is dropped

Directed Diffusion

Reinforcements

- Used when the sink start receiving data matching an exploratory interest from node u
- The sink reinforces node u to improve the quality of received data
  - Exploratory interests use a low sampling rate
  - Reinforces of interests specify an interest with larger sampling rate
- In turn, a node receiving a reinforce of an interest reinforces one of its neighbors
  - Reinforces are propagated through the path along which the data flows

Directed Diffusion

Drawbacks

- Assumes that the sink is permanently connected to the network
  - the network does not operate autonomously
- Sensors do not process data (apart aggregation)
  - The sensors just send the data matching the interests to the sink
  - Does not exploit processing and storage capabilities of the sensors

Flexible network design needs flexible routing

Greedy Perimeter Stateless Routing
(GPSR)

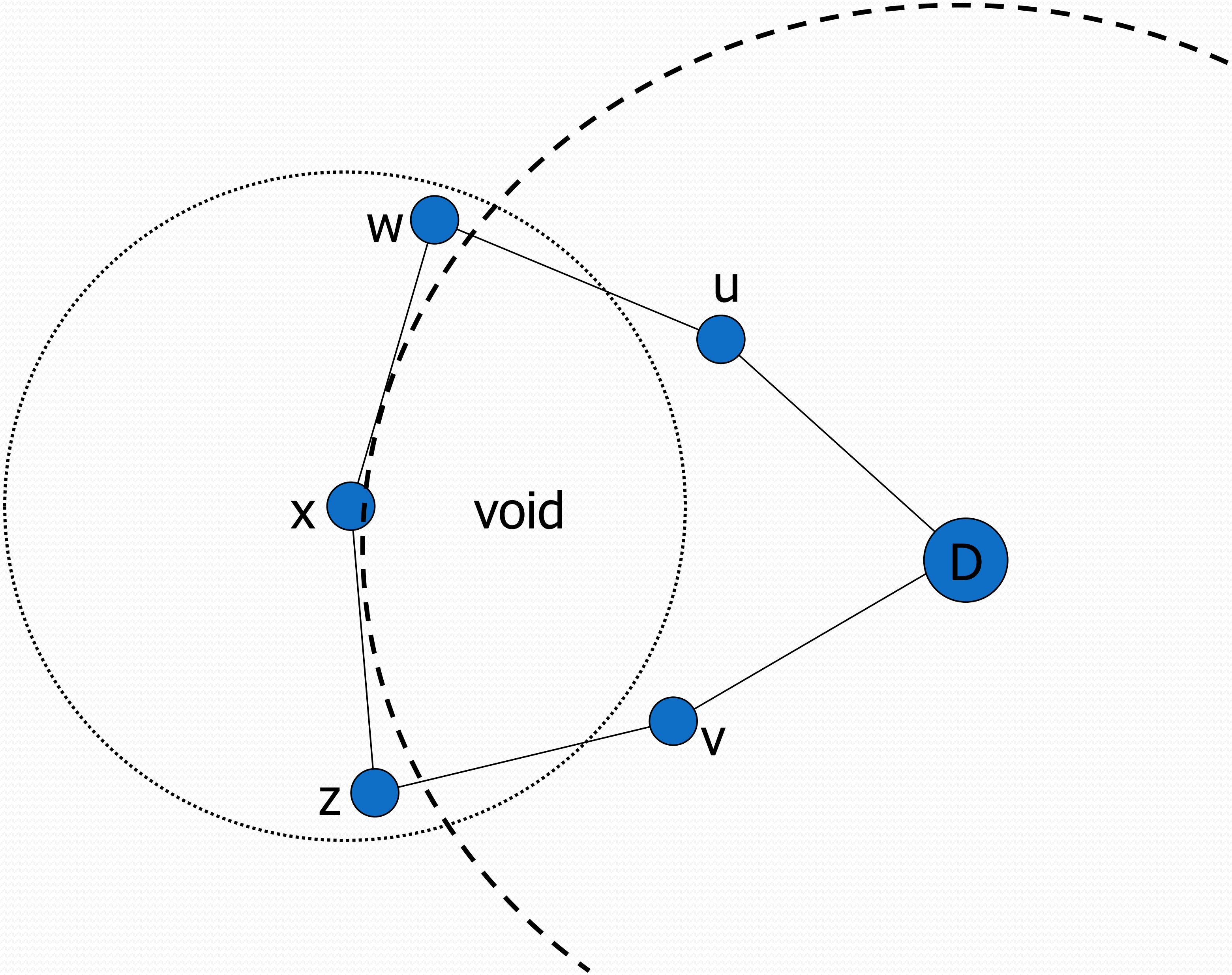
Routing with GPS: GPSR

- Karp & Kung, Mobicom 2000
- Assumptions:
  - The nodes are deployed on a two-dimensional space
  - Nodes are aware of their position and of the position of their neighbors
    - For example the nodes are equipped with GPS
  - The source knows the coordinate of the destination
    - Packet headers contain the destination coordinate
- The protocol is scalable:
  - No need for route discovery
    - Few control packets
  - Nodes maintain only local information
    - Large route caches or routing table are not necessary
  - Packet headers do not need to store routes

GPSR

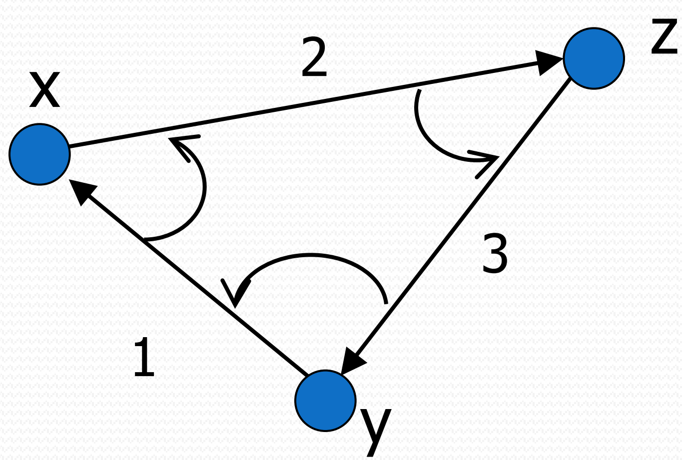
- GPSR comprises two modes:
  - Greedy forwarding
  - Perimeter forwarding
- Greedy forwarding
  - Consider a packet with destination D
  - the forwarding node x select as next hop a neighbor y such that:
    - y is closer to D than x
    - Among neighbors y is the closest to the destination
  - Greedy forwarding fails if the packet encounters a “void”

GPSR



GPSR

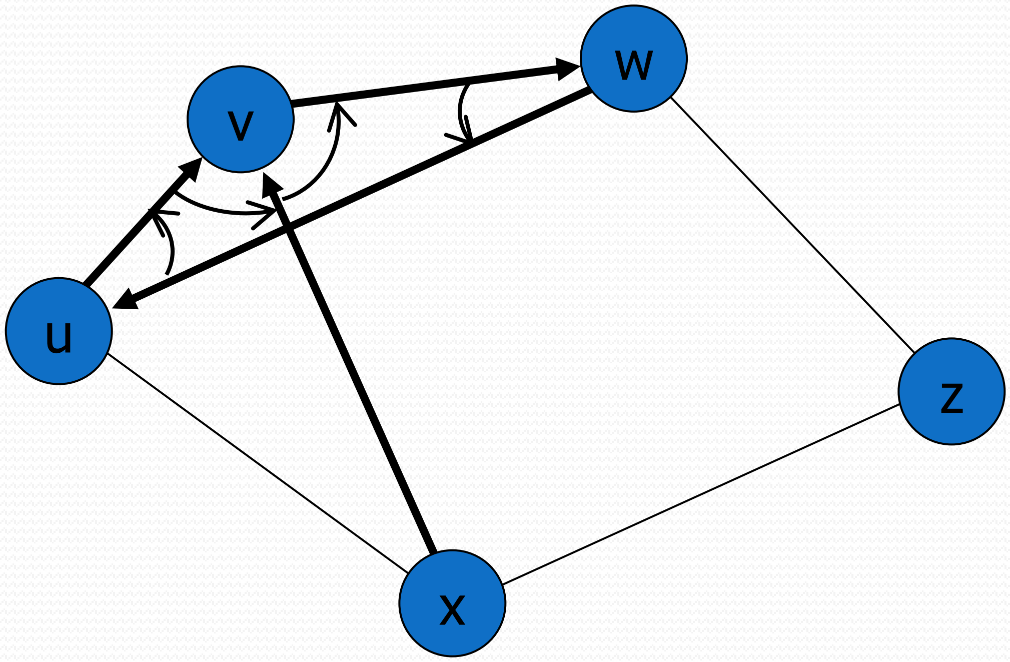
- Perimeter mode forwarding is executed when greedy forwarding finds a void
  - Routes around the void
  - Based on the right hand rule
    - When arriving from y to x
    - Selects as next edge the one sequentially counterclockwise from edge (x,y)
    - Traverses the interior of a closed polygonal region (face) in clockwise edge order
  - Intuitively it explores the polygon enclosing the void to route around the void
    - In the previous example it would produce x – w – u – D



GPSR

- However, graph G corresponding to the sensor network is a non-planar embedding of a graph
  - Edges may cross
  - the right hand rule may take a degenerate tour of edges that does not trace the boundary of a closed polygon
  - In the example, from x to v the right hand rule produces the path

x – v – w – u – x

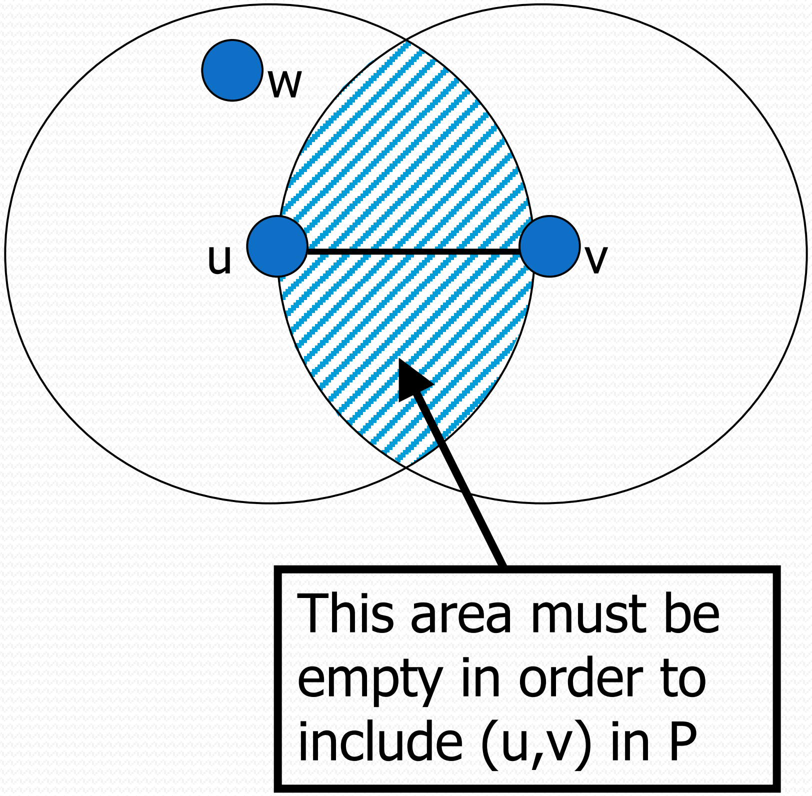


GPSR: graph planarization

- For this reason GPSR applies the perimeter mode to a planar graph P obtained from G
  - *Relative Neighborhood Graph* of G
  - *Gabriel Graph* of G
  - Properties:
    - If G is connected then P is connected
    - P is obtained from G by removing edges
    - P is computed with a distributed algorithm executed along with the perimeter mode packet forwarding

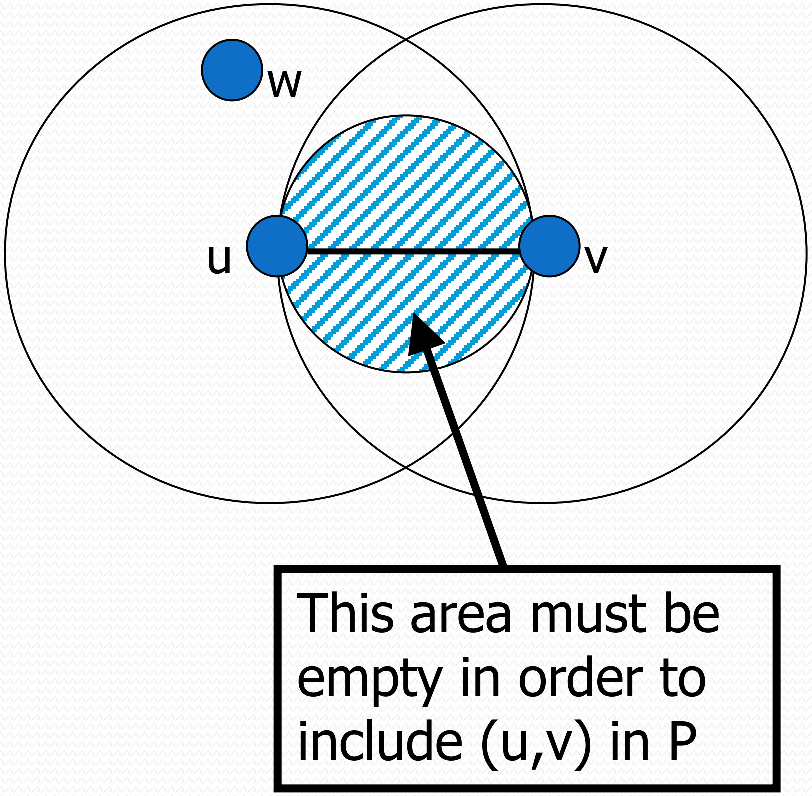
GPSR : graph planarization

- *Relative Neighborhood Graph* (P) of G:
  - Edge $(u,v) \in P$ iff
    - $(u,v) \in G$
    - $d(u,v) \leq Max(d(u,w),d(v,w))$ for each $w \in N(u) \cup N(v)$
- Consider the forwarding node u:
  - u considers each neighbor $v \in N(u)$
  - edge (u,v) is kept iff the above property is satisfied

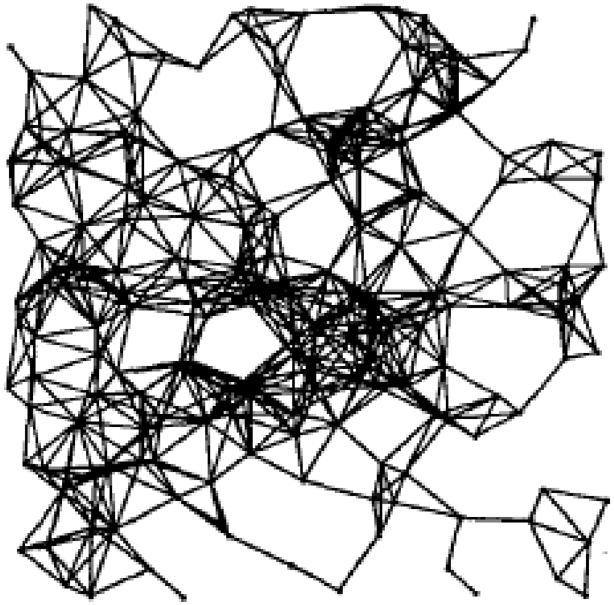


GPSR : graph planarization

- *Gabriel Graph* (P) of G:
  - Edge $(u,v) \in P$ iff
    - $(u,v) \in G$
    - $d^2(u,v) \le [d^2(u,w) + d^2(v,w)]$ for each $w \in N(u) \cup N(v)$
- GG constructed with a distributed algorithm as RNG
- RNG is a subgraph of GG
  - RNG has lower link density
- RNG or GG are both suitable to GPSR



GPSR : graph planarization



Full graph, Gabriel Graph and Relative Neighborhood Graph

GPSR: perimeter mode

- Packet header in perimeter mode:

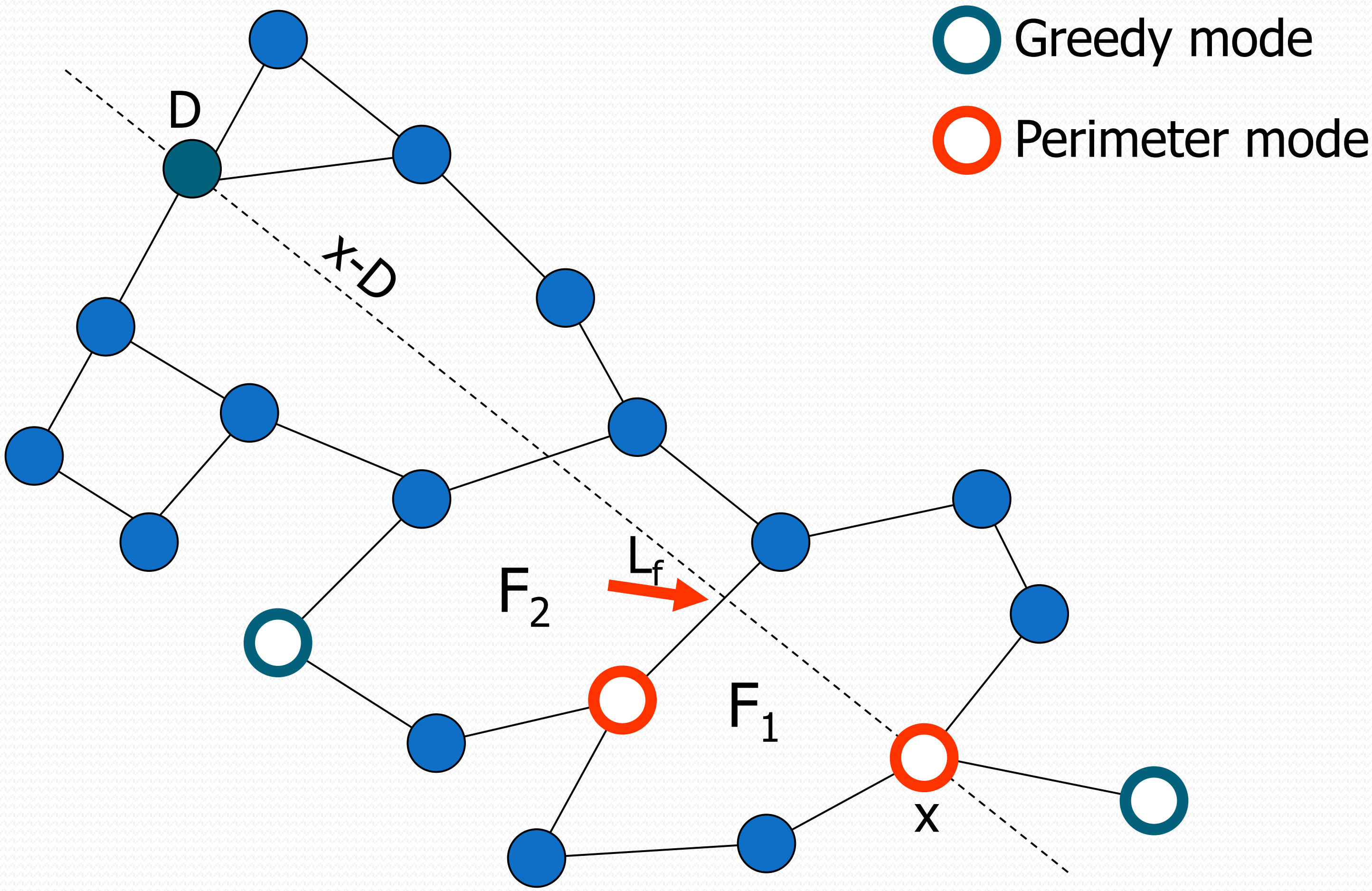
Field	Function
D	Destination Location
x	Location where packet entered in perimeter mode
L_f	Point on x-D where the packet entered current face
e_o	First edge traversed on current face
M	Packet mode: greedy or perimeter

- Let x be the node where the packet enters in perimeter mode
  - Consider the line x-D
- GPSR forwards the packet on progressively closer faces on the planar graph, each of which intersects x-D

GPSR: perimeter mode

- A planar graph has two types of faces:
  - Interior faces
    - Closed polygonal regions bounded by the graph edges
  - One exterior face
    - The unbounded face outside the outer boundary of the graph
- On each face GPSR uses the right hand rule to reach an edge which crosses x-D (and that is closer to D than x)
- At that edge GPSR moves to the adjacent face crossed by x-D
  - Each time it enters a new face the packet records:
    - In L_f the point on the intersection between x-D and the current edge
    - In e_o the current edge
- However … GPSR returns to greedy mode if the current node is closer to D than *x*
  - Perimeter mode is intended to recover from a local maximum…

GPSR: perimeter mode



GPSR: perimeter mode

- If D is reachable from x (G is connected) then GPSR always finds a route
  - Only if the network is planarized with RNG or GG
- if D is not reachable:
    - Either D lies inside an interior face F_i
    - Or D lies in the exterior face F_e
  - The packet will reach the face (either F_i or F_e)
  - Then it will tour around the face until it travels again along the edge e_o
  - At that point the packet is discharged

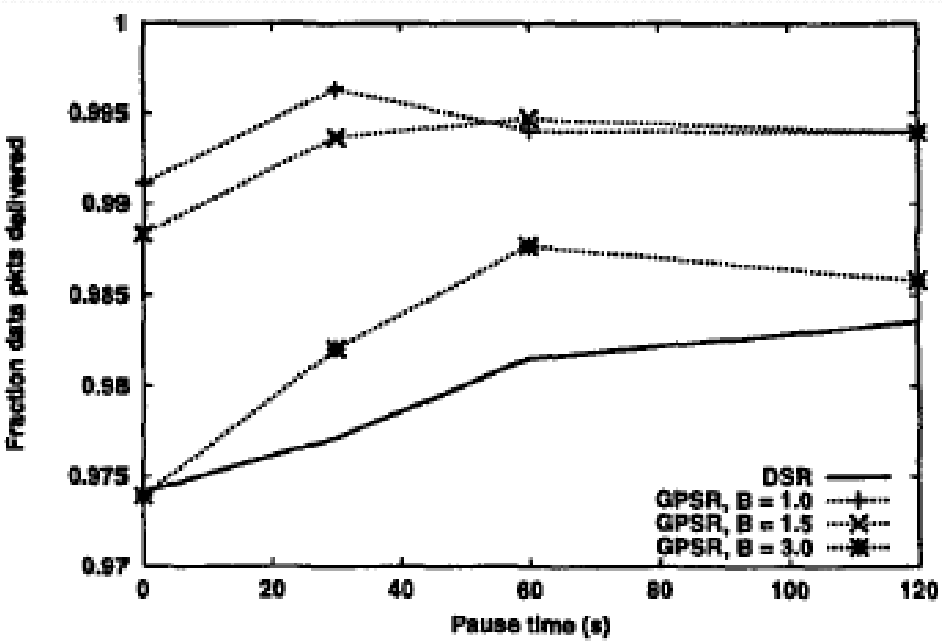
GPSR

- GPSR and mobility:
  - GPSR relies on updated information about the position of the neighbors
  - It need a freshly planarized graph
    - Using stale planarized graph may result in performance degradation
  - Performing planarization at topology changes is not sufficient
    - Nodes may move within a node's transmission range
    - This may change the selection of links operated by GG or RNG
  - Proactive approach: nodes periodically (at each beacon interval) communicate their position their neighbors
    - This information is used to keep updated the list of neighbors and to force planarization

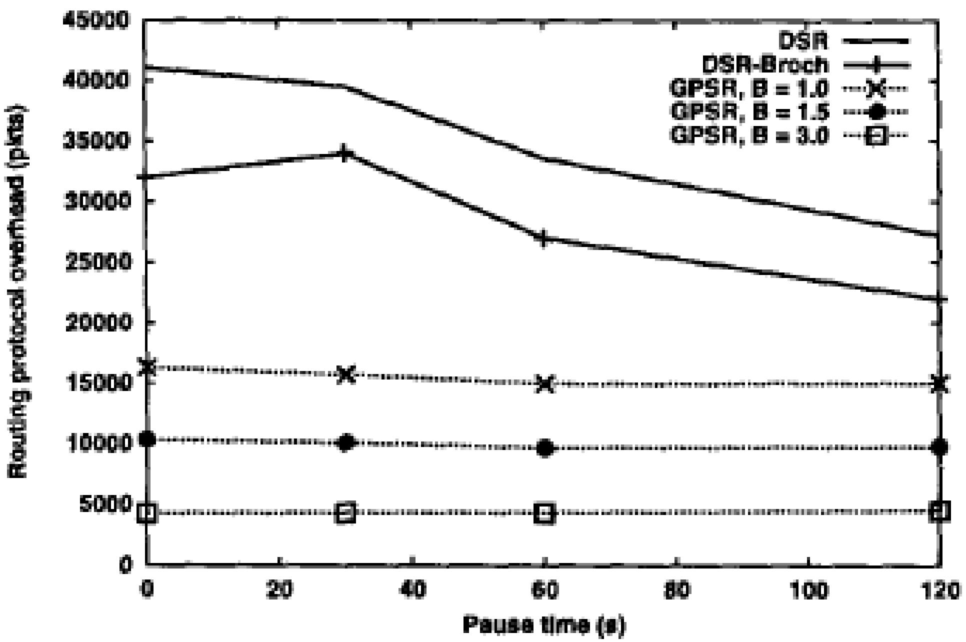
GPSR - simulation

- Decreasing the beaconing time the delivery rate of GPSR
- Routing overhead (beacon packets) is independent of mobility
  - Beacons are proactive

Delivery rate

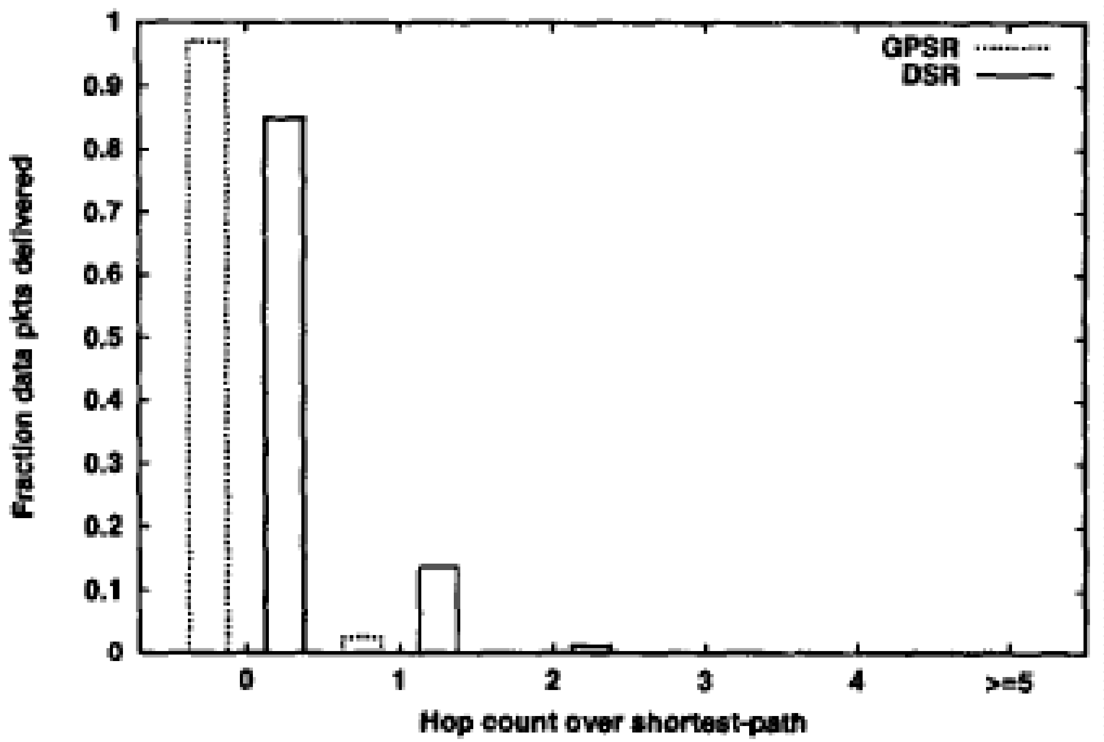


Routing overhead



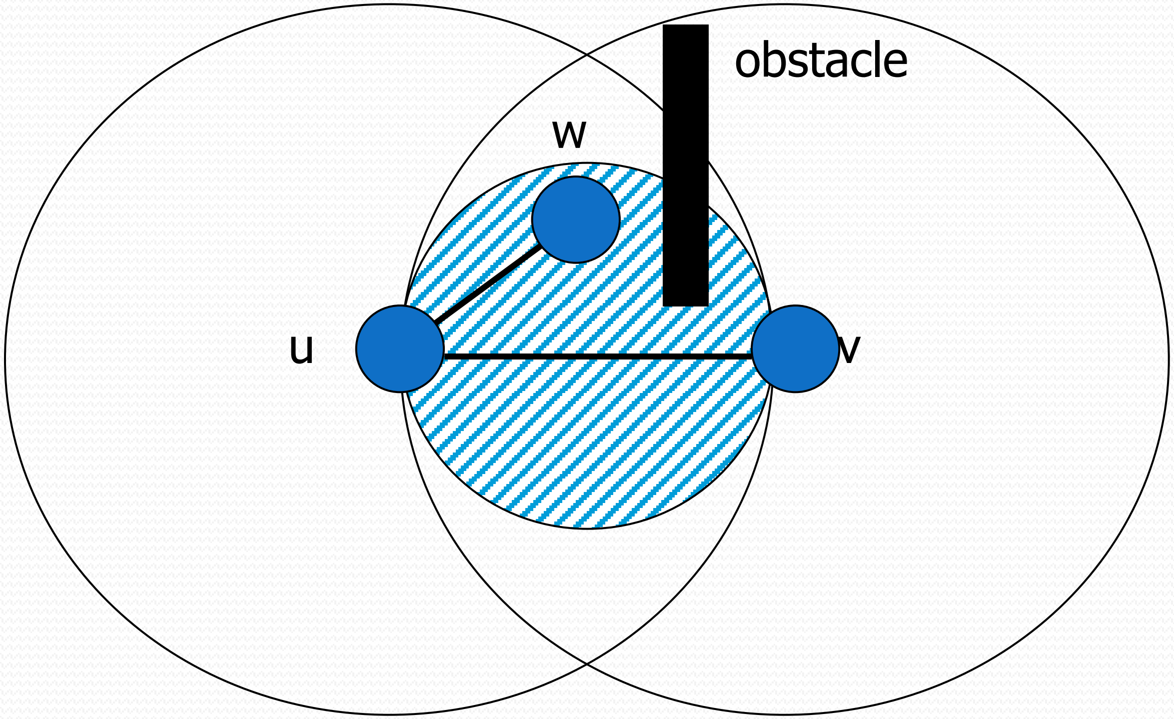
GPSR - simulation

- Path length: nearly optimal if the network is dense
  - 95% of packet delivered through the shortest path VS 85% of DSR
    - Difference due to the caching of DSR, some paths in the cache may be no longer optimal
  - Intuitively greedy routing approximates shortest paths



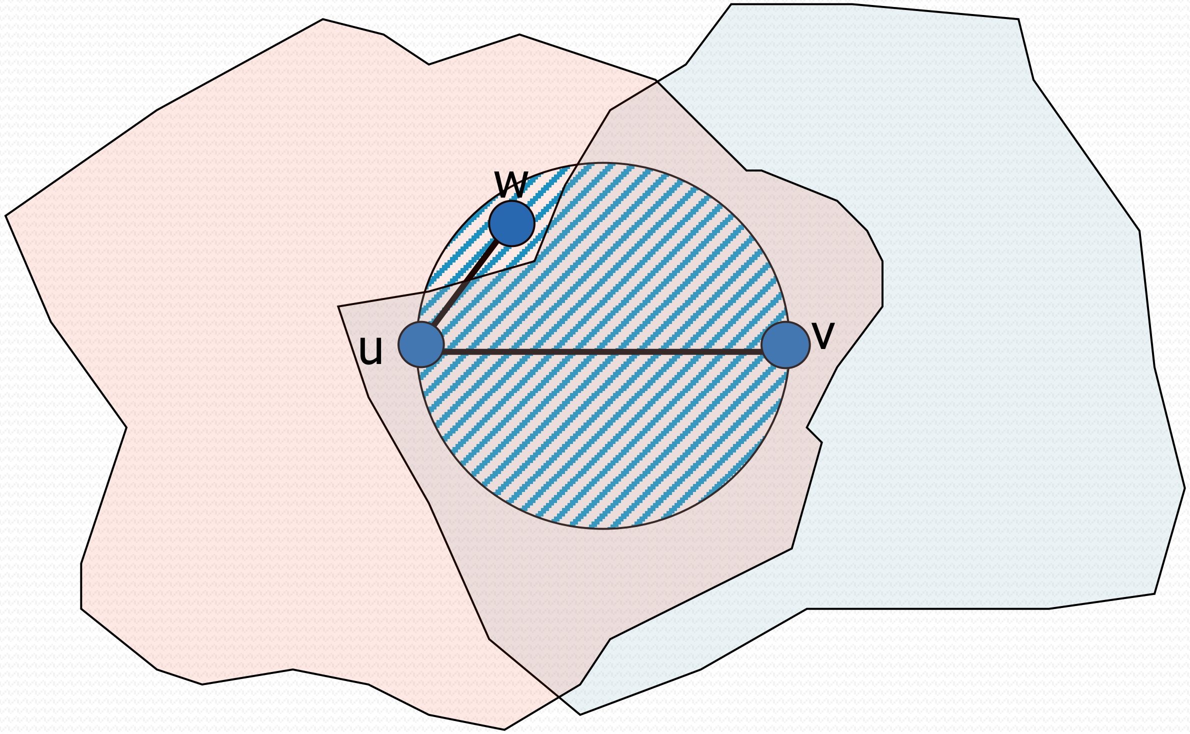
GPSR - drawbacks

- Planarization failures due to unidirectional links:
  - Because of obstacles



GPSR - drawbacks

- Planarization failures due to unidirectional links:
  - Because the assumption of unit disk graph does not hold

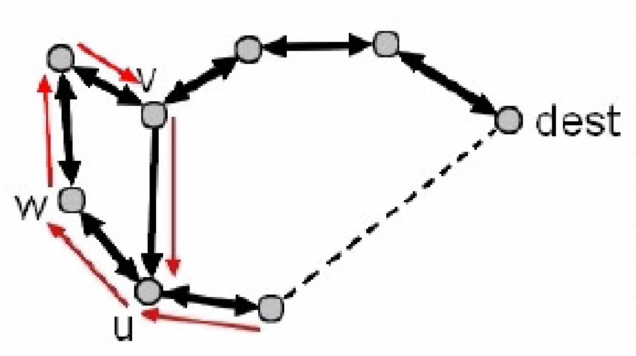


Failure of GPSR

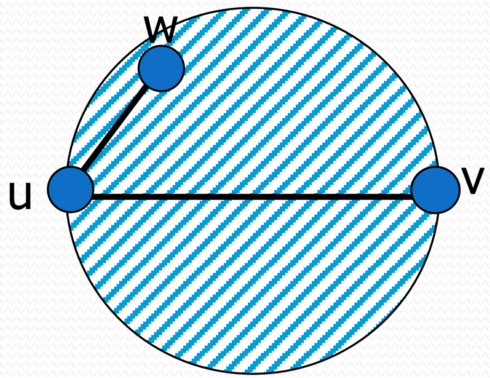
- Exercise:
  - Construct an example in which obstacles or non-circular transmission range produce loops in the GPSR packet forwarding
  - Hint: construct a graph in which not all the links are bidirectional
  - 5 minutes…

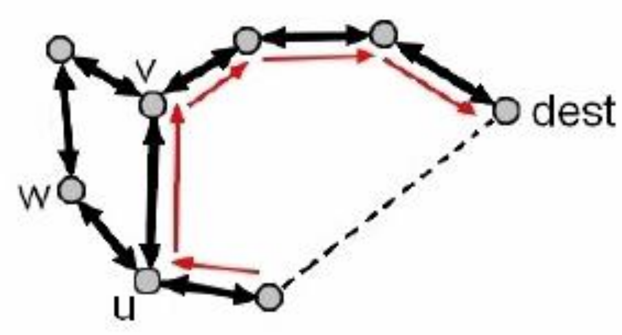
GPSR with Mutual Witness

- The presence of unidirectional links may lead to loops:



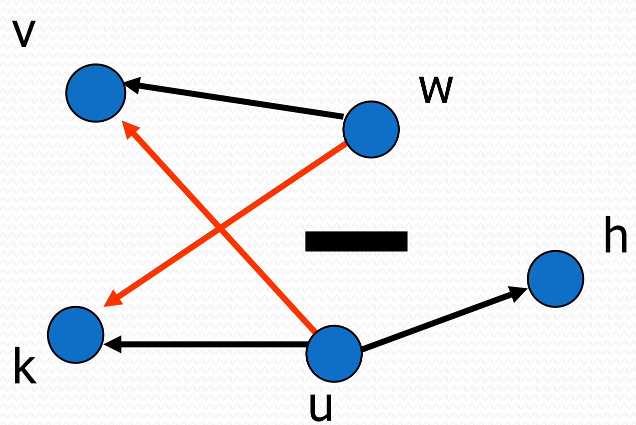
- Mutual witness extends the planarization algorithm of GPSR:
  - If the link w–v does not exists then keeps link u–>v (link v->u is kept by v anyway)
  - Only bidirectional links: no more loops.





Failures of GPSR with MW

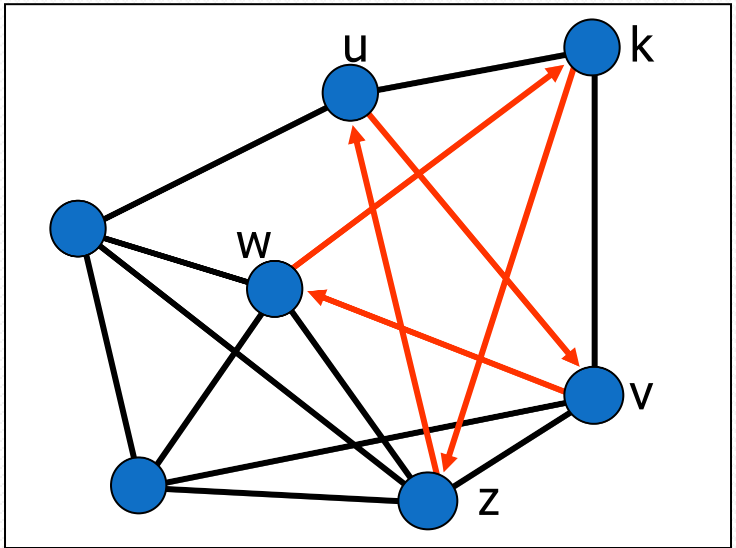
- There are cross links which are undetectable by Mutual Witness
  - The cross of links u-v and w-k are not detectable
    - u and v use w as witness for link u-v
    - w and k use u as witness for link w-k
  - Thus MW would take both u-v and w-k



- CLDP (Cross Link Detection Protocol) to detect all the cross links

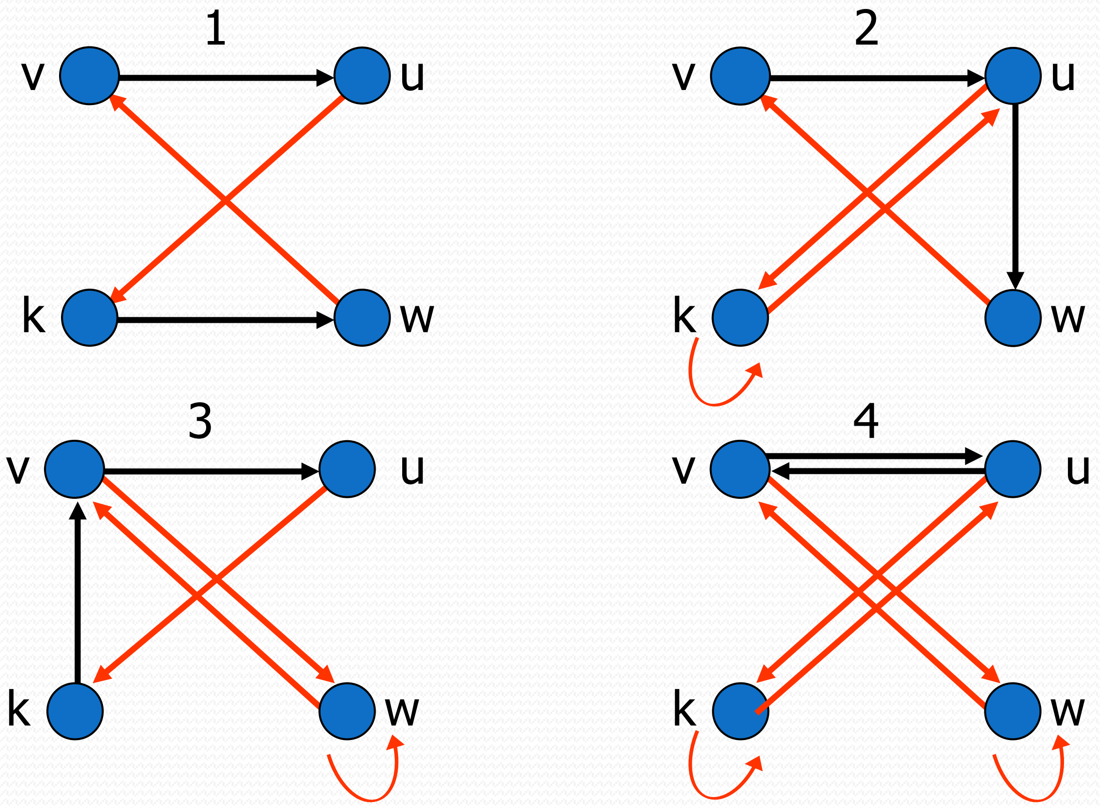
GPSR with CLDP

- CLDP operates on the full graph (no preliminary planarization)
- Each node sends a probe through each of its outgoing links
- The probe crosses the graph using the right hand rule
- Each node controls the coordinates of the nodes crossed by the probe:
  - If it finds a link crossing the current link it records the information in the probe
- If cross links are detected the source node may decide to remove one of the crossing links
  - In the figure the first cross links detected by the probe are u–v and w–z. Any of the two can be removed.



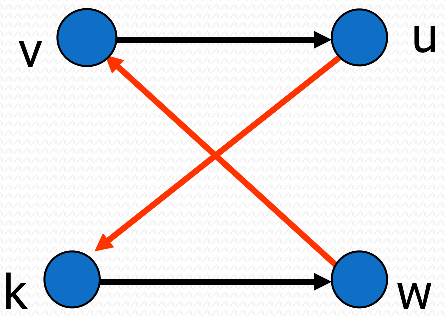
GPSR with CLDP

- However a link removal may result in network disconnections
- For this reason the probe counts the number of times it crosses a link.
- If a link had been crossed only once then it can be removed
  - there exist a loop and thus it is possible to reach any node in the loop by an alternative path
- Four cases of CLDP: node w sends a probe to v



GPSR with CLDP

- Link removal may require additional communications between nodes
- To reduce the overhead CLDP uses some rules (let us assume that node v tests outgoing link L which crosses link L'):
  - If L' cannot be removed then v removes L
  - If both links can be removed then v removes L (which requires less communications)
  - If nor L neither L' can be removed then both links are kept.
  - If L cannot be removed then v removes L' although it requires additional communications.
- In the figure node v would remove link v–w (this requires only one communication from v to w).



GPSR with CLDP

- It should be observed that a probe can be used to identify and remove only one pair of cross links.
  - removing a link implies a change in the topology
  - Removing more than one link per probe may result in network disconnections
- If there exists several cross link then a node should send a probe on the same link until no cross link are detected.

Considerations on GPSR

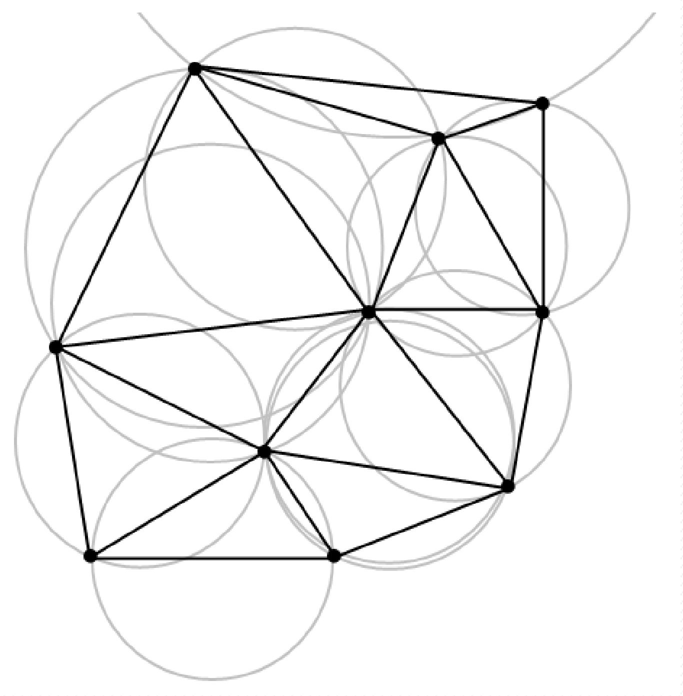
- GPSR (and GPSR+MW) does not guarantee delivery in real settings
- GPSR + CLDP very complex
- In theory GPSR + CLDP works in 3D and in complex indoor settings, but in practice?
  - Some links might be intermittent…

Other geographic routing protocols

- GPSR is one of a large number of different geographic routing protocols
- Some protocols keep the line x-D as reference for the routing protocol:
  - GPSR
  - Greedy-Face-Greedy (GFG)
  - Compass Routing II
- Some others start again the process each time they change face:
  - Greedy Other Adaptive Face Routing (GOAFR+, GOAFR++)
  - Greedy Path Vector Face Routing (GPVFR)
- All of them require planarization

Other geographic routing protocols

- All of them require planarization
  - GG
  - RNG
  - Delaunay triangulation
    - Picture from wikipedia



Other geographic routing protocols

- Not all of them work properly with any planar graph:
  - GPSR (without greedy) may loop with Delaunay or arbitrary, planar graphs
  - GPVFR may loop with arbitrary planar graphs
  - GOAFR+ may fail with Delaunay or arbitrary, planar graphs
  - GFG and GOAFR++ works well with any planar graph.
- Routing with guaranteed delivery without constraints is still an issue

Data-Centric Storage (DCS) and Geographic Hash Tables (GHT)

DCS & GHT

- Ratnasamy et Al., MONET 2003
- Focus:
  - The sensor network can operate in an unattended mode
  - Samples and Records information about the environment
  - Need for:
    - Data-dissemination techniques to extract data
    - Data-centric storage
  - Based on:
    - Geographic routing protocols
    - Peer-to-peer lookup systems

DCS & GHT

- Data-centric storage:
  - Events are named (keys) and corresponding data are stored by names in the network
  - Queries are directed to the node that stores events of that key
- Two operations supported by DCS:
  - Put(k,v) stores the observed data according to its key k
  - Get(k) retrieves whatever value associated to key k

DCS & GHT

- Design criteria of a DCS
  - Node failures: battery exhaustion, HW failures, …
  - Topology changes: due to node mobility, failures, …
  - Scalability: number of nodes, network density
  - Energy constraints
  - Persistence: a stored pair (k,v) must remain available despite failures and topology changes
  - Consistency: a query for key k must reach the node where pairs (k,v) are actually stored
  - Scaling in database size: storage should not overburden a node as the number of pairs (k,v) increase

DCS & GHT

Geographic hash table built on top of the GPSR routing protocol:

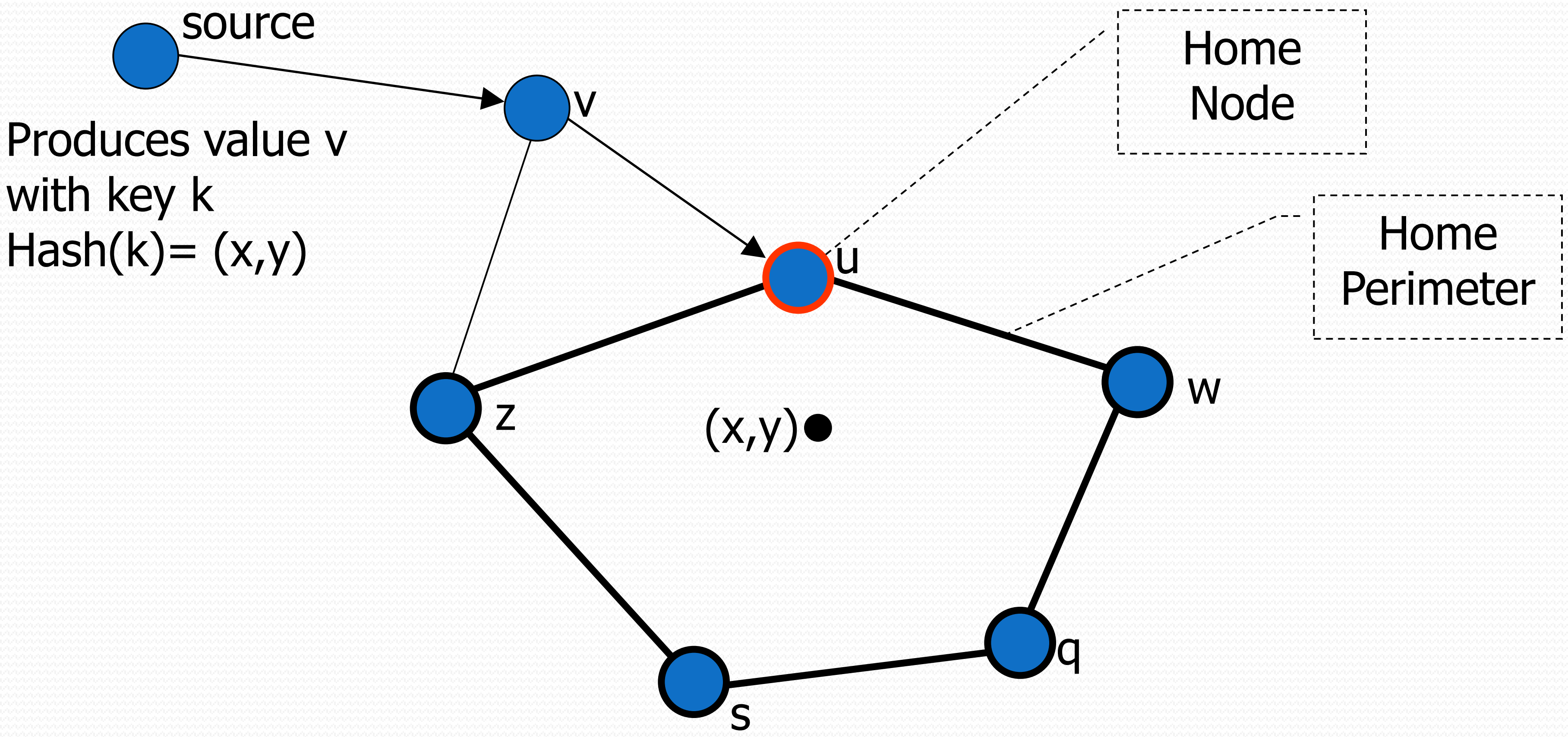
- Put(k,v):
  - (x,y) = hash(k);
    - Hash(k) returns a pair of coordinates (x,y)
      - (x,y) should be included in the network boundary, it is assumed this information is known to each node
  - Send <k,v> to point (x,y) using GPSR
    - (k,v) is stored by the node u which is the closest to coordinates (x,y)
- Get(k):
  - (x,y) = hash(k);
  - Send a request to point (x,y) using GPSR
    - Queries related to key k are routed to (x,y)

DCS & GHT

- Mobility or failure of sensor u may result in unavailability of stored value v
  - GHT uses a Perimeter Refresh Protocol (PRP) to provide persistence and consistency
    - PRP selects one node as home node for key k
    - PRP replicates v on the nodes in the perimeter around (x,y)

DCS & GHT

- Home node & Home perimeter



DCS & GHT

Perimeter Refresh Protocol (PRP)

- Accomplish replication of key-value pairs
- GHT routes the packet (k,v) around the perimeter enclosing (x,y)
    - Where (x,y)=Hash(k)
  - The perimeter is identified by GPSR
- (k,v) is stored in the home node
- Each node in the home perimeter stores a replica
  - Nodes on the home perimeter are said replica nodes

DCS & GHT

Perimeter Refresh Protocol (PRP)

- The home node u of key k generates periodical refresh packets, each of which:
  - is sent to coordinate (x,y)=Hash(k)
    - Note that the home node might have moved
  - contains (k,v)
  - tours around the perimeter around (x,y)
- The refresh packets preserve consistency: the home node should be the closest to (x,y)

DCS & GHT

Perimeter Refresh Protocol (PRP)

- If the refresh packet reaches a node v closer to (x,y) than the old home node u
  - v generates a new refresh packet towards (x,y)
    - It is eligible as a new home node for k
- When the refresh packet reaches again its source v
  - v becomes the home node for k
  - v sets up a refresh timer for k
  - v replicates (k,v) in the new perimeter

DCS & GHT

Perimeter Refresh Protocol (PRP)

- However the home node may fail
  - Need to enforce persistence
- Hence each replica node sets up a takeover timer
  - The timer is reset when a refresh packet is receved
- When the timer expires the replica node generates a refresh packet for (k,v) towards (x,y)

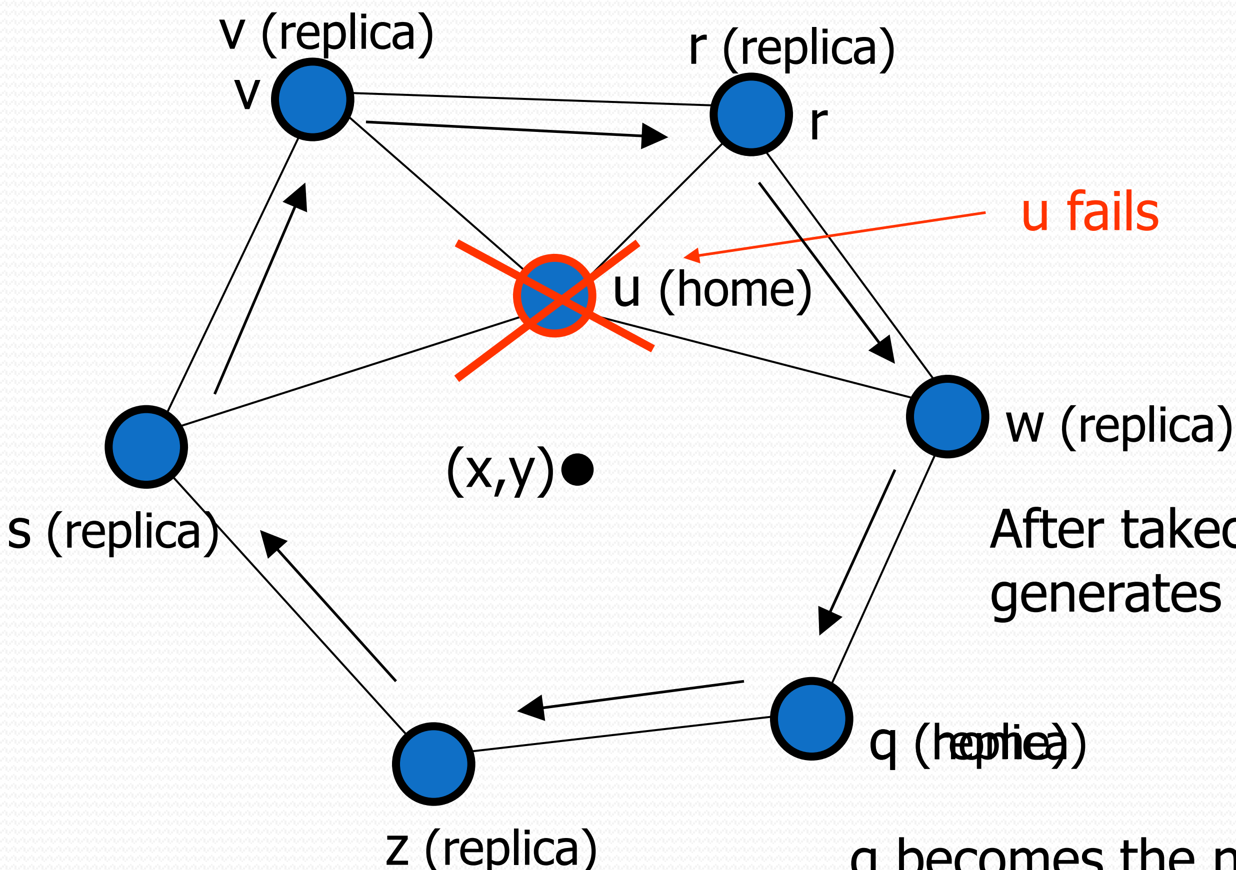
DCS & GHT

Perimeter Refresh Protocol (PRP)

- Pairs (k,v) are not cached forever
  - If a home (replica) node moves it might not be associated to key k anymore
  - Discharging (k,v) should not affect availability
- Home and replica nodes use death timers
  - Each pair expires after a death timeout
  - The death timeout should be larger than the refresh and takeover timeouts

DCS & GHT

Perimeter Refresh Protocol (PRP)



After takeover time w generates a refresh

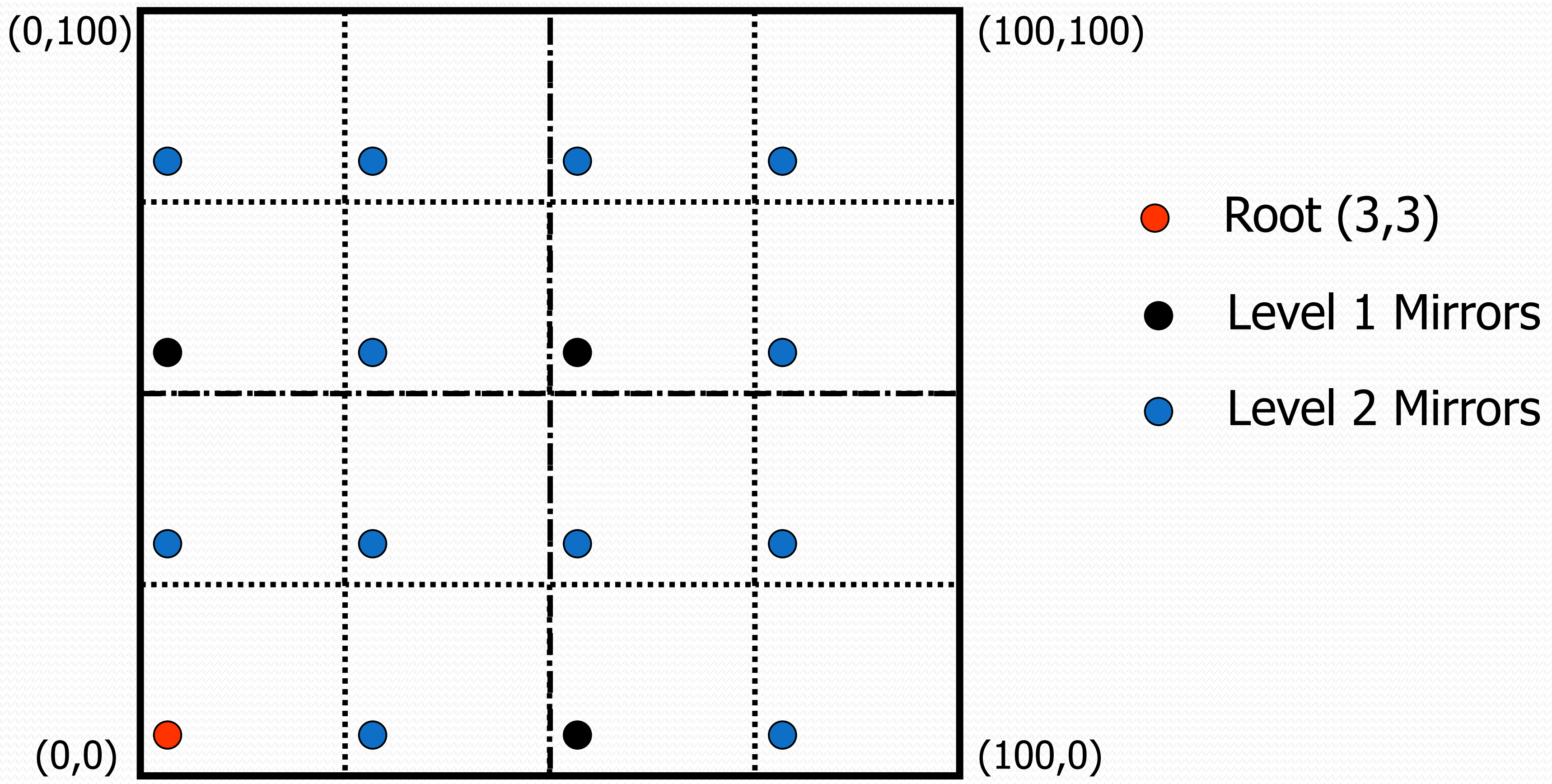
q becomes the new home node

DCS & GHT

- A home node for key k might be overburdened if too many values with key k are produced
- Structured replication (SR)
  - uses a hierarchy (of depth d) of event names
  - Hash(k) is the root of the hierarchy
  - To each key k are associated a root and 4^d-1 mirrors
  - A node stores the pair (k,v) to its closest mirror of Hash(k)
    - The mirror informs its ancestors that it stores values with key k
  - Retrieval of a value involves queries to the root and (possibly) all mirrors
    - The query is first directed to the root
    - The root forwards the query to the interested descendant mirrors
    - Trades storage overhead with communication

DCS & GHT

- An example of Structured Replication with d=2



Summary of DCS-GHT

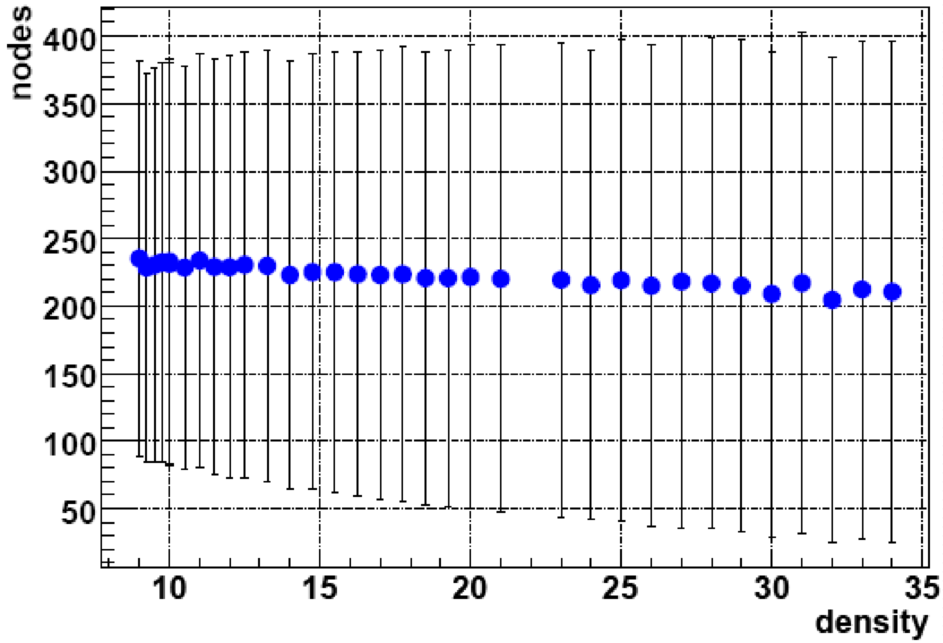
- Data Centric Storage based on Geographic Hash Tables
  - nodes should be aware of their coordinates
  - Nodes should know the network boundary
- Built on top of GPSR
- Perimeter Refresh Protocol to enforce persistence and consistency
- Structured Replication to enforce scaling in database size

Drawbacks of DCS & GHT

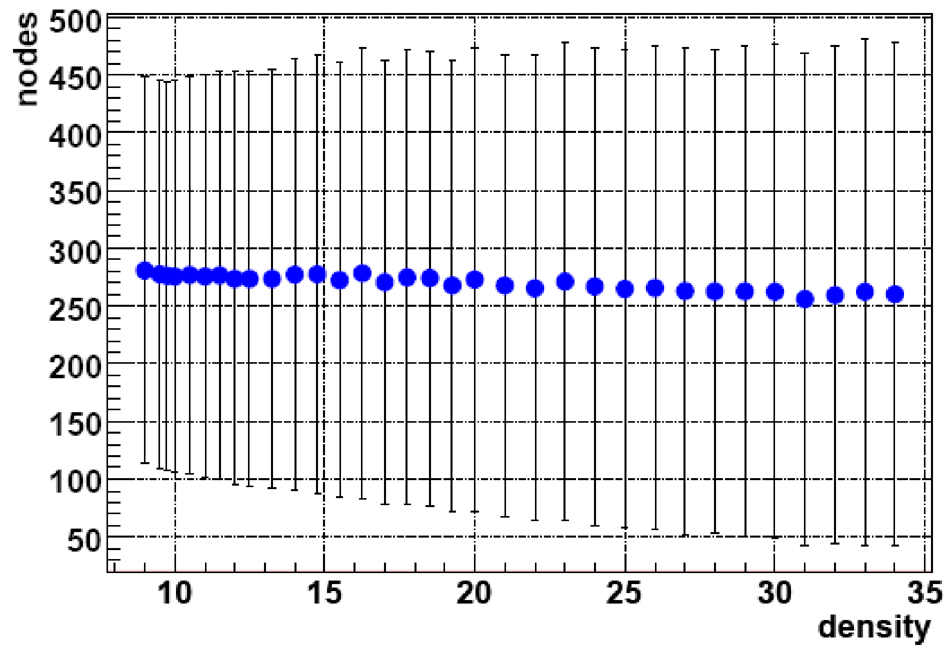
- No control on the degree of data replication
  - Data replicated in the home perimeter
  - Size of the home perimeter is unknown a priori
  - Home perimeter size may vary significantly
    - what happens if hash(k) returns a point outside the boundary of the network?

Drawbacks of DCS & GHT

- Mean and variance of GHT perimeters for different network densities, Gaussian distribution
    - Networks with 3000 to 20000 nodes
  - Mean and variance of perimeters (number of nodes) measured with
  - a) planarization with gabriel graph
  - b) planarization with RNG



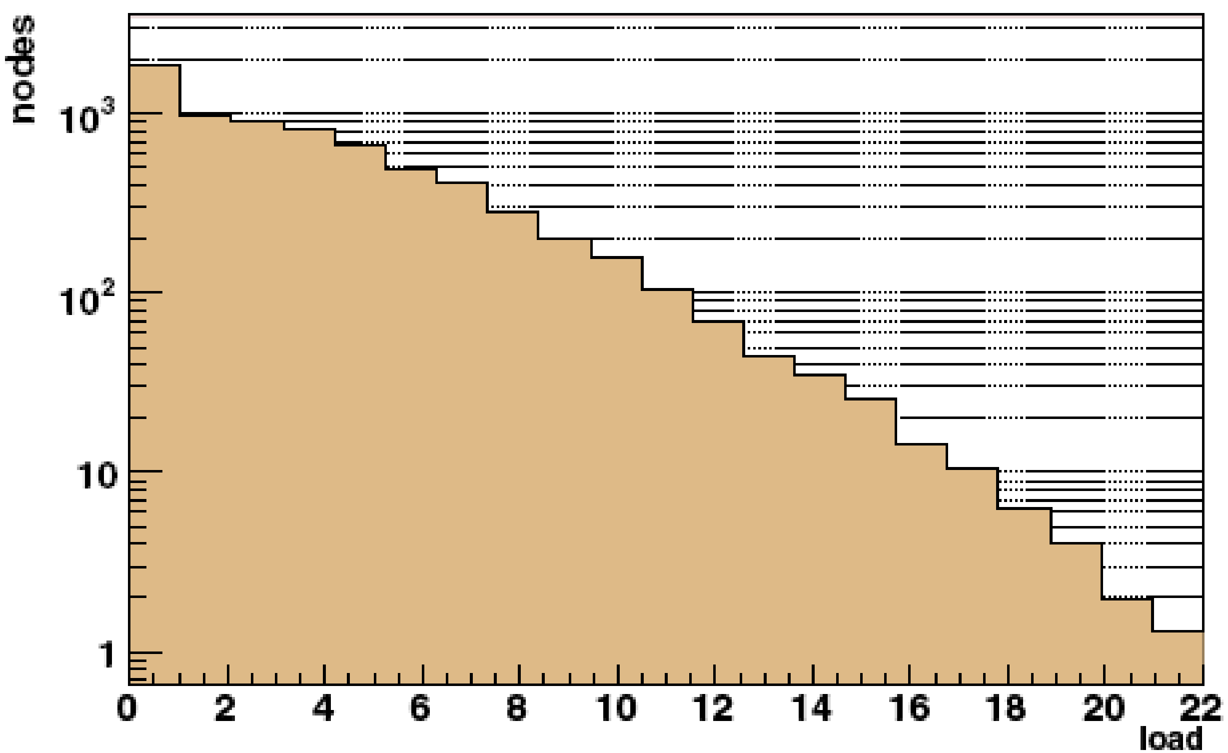
(a) GG



(b) RNG

Drawbacks of DCS & GHT

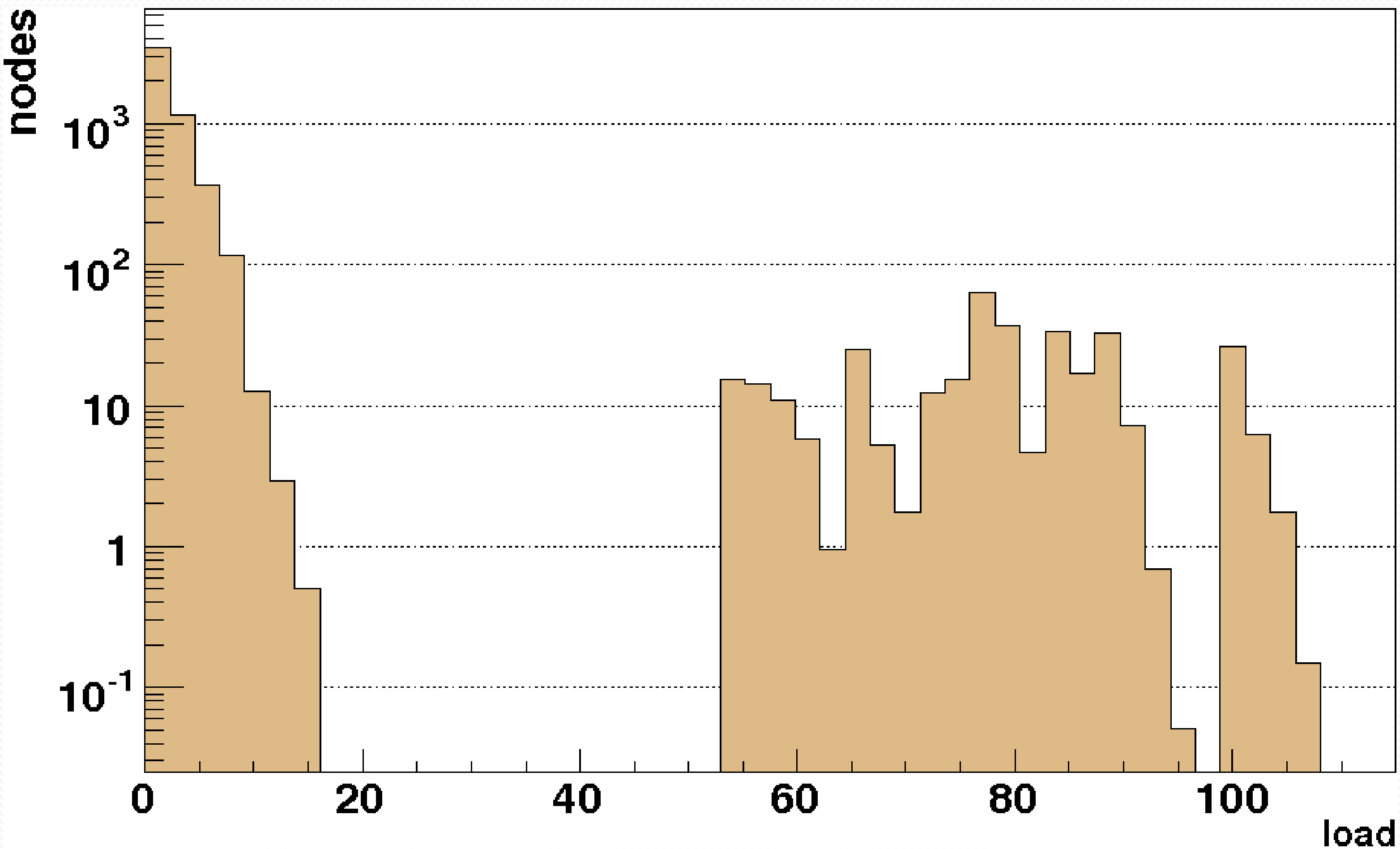
- Average load of sensors



(a) GHT with uniform distribution

Drawbacks of DCS & GHT

- Average load of sensors with gaussian distribution of sensors



Other DCS approaches

- Many other systems for data centric storage have been proposed so far:
  - Q-NiGHT & LB-DCS – to overcome the load balancing issues
  - CHR, GLS – exploits clustering for scalability
  - GEM – uses node labels rather than coordinates
  - RR – uses regions rather than coordinates to relax the requirements for localization accuracy
  - … and many others
- DCS is still focus of research

Physical and virtual Coordinates

Geographic Routing and Localization

- Traditional routing protocols for ad hoc networks are not practical:
  - Large routing tables or path caches
  - Size of packet headers
- Geographic routing appears to be the best option
- Coordinates are mandatory
  - To support geographic routing
  - Support the implementation of a data centric storage (DCS with GHT)
  - Provide a relation between sensed data and locations

Sensor Coordinates

- Coordinates can be obtained by equipping nodes with GPS
  - Additional cost
  - Not always feasible (for example indoor)
- When no GPS system is available:
  - Either a few anchor nodes know their position
    - other nodes compute coordinates with a variety of methods
  - Or virtual coordinates are used
    - typically they are hop-distances dependent

Sensor Coordinates without GPS

Approximate physical coordinates of the sensors

- A few nodes (anchors) know their exact position
  - by means of special hardware (GPS)
  - or because they are deployed in precise, well known positions
- The anchors broadcast beacons
- The other nodes estimate their positions with distributed protocols
  - Nodes may estimate their distance from the anchors
    - Time of arrival techniques
    - Signal strength
  - Or they may estimate their position based on triangulation
    - angle of arrival

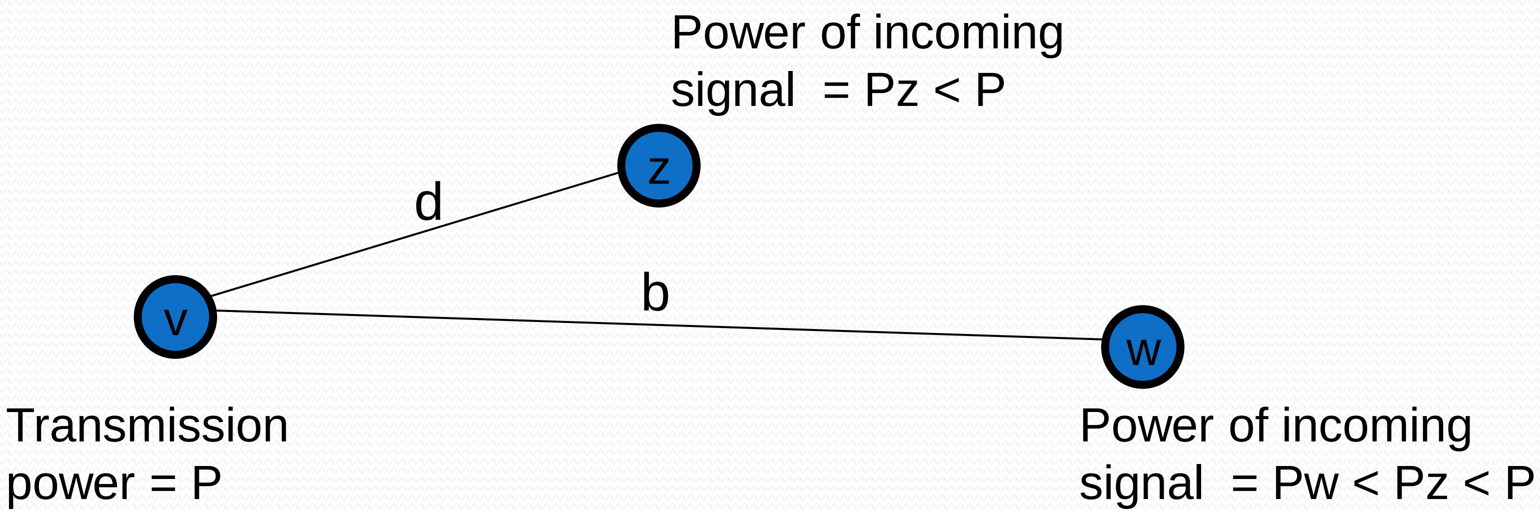
Range and Range-Free techniques

Two basic techniques:

- Using special *Ranging* hardware (e.g. signal strength, T.O.A., etc).
- Range Free Technique.
  - Cost-Effective: No special hardware for ranging.
  - **Topology based (Hop counting) techniques**.

Range techniques

- Received Signal Strength (RSS):
  - Uses power of signal to estimate distances
  - Power of the signal decays with an exponential rule

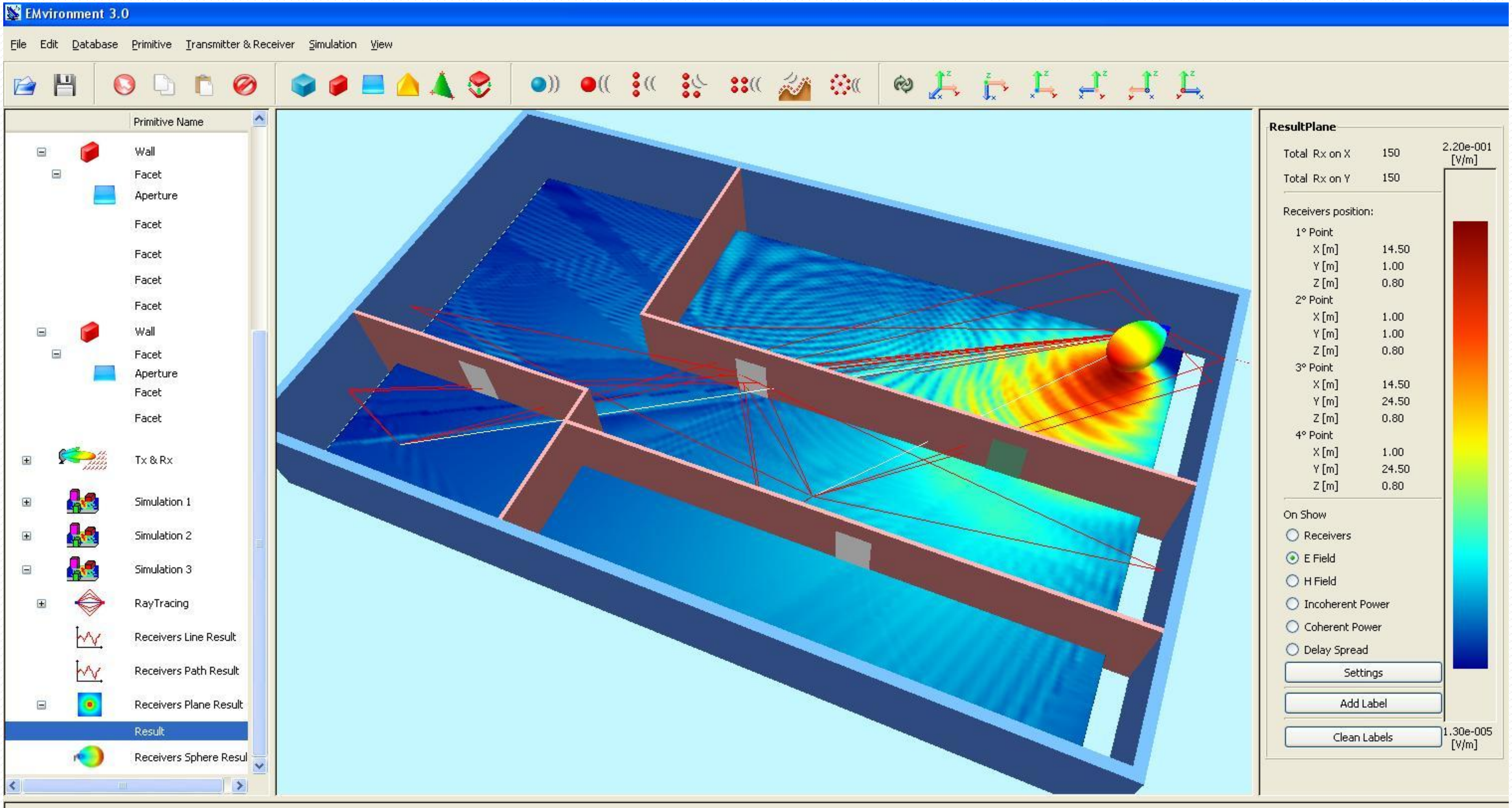


Received signal strength

- Signal attenuation depends on the environment.
- There are many models that relate distance with transmission and received power.
- The one slop model states that the path loss at distance d L(d) is:
  - $L(d) = l + 10\, a \log_{10}(d)$
- where
  - l is attenuation of signal at a reference distance (for example 1 m)
  - a is the path loss (typically in the range [2,4])

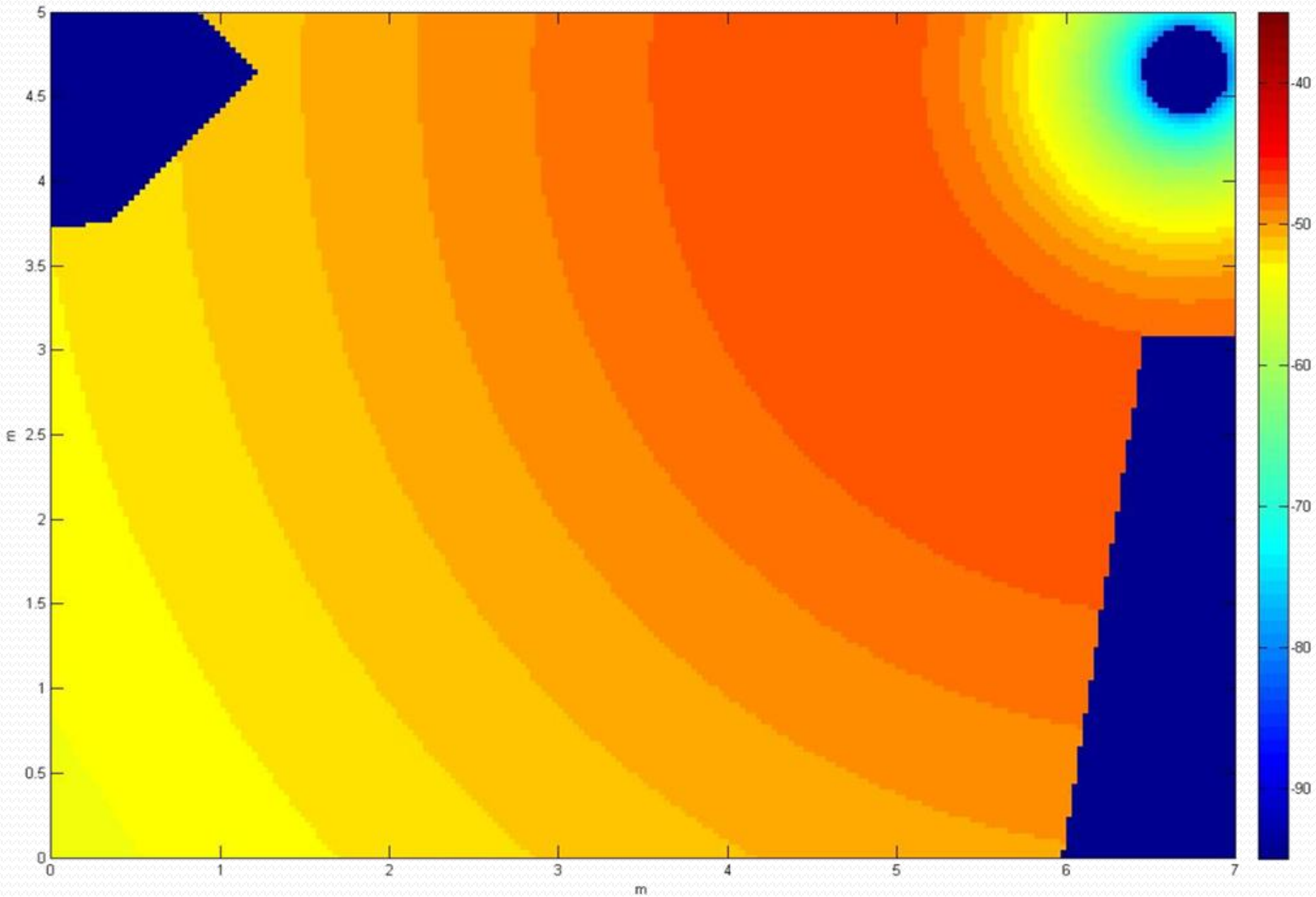
Received signal strength

- When used in indoor environments the quality of RSS worsens significantly



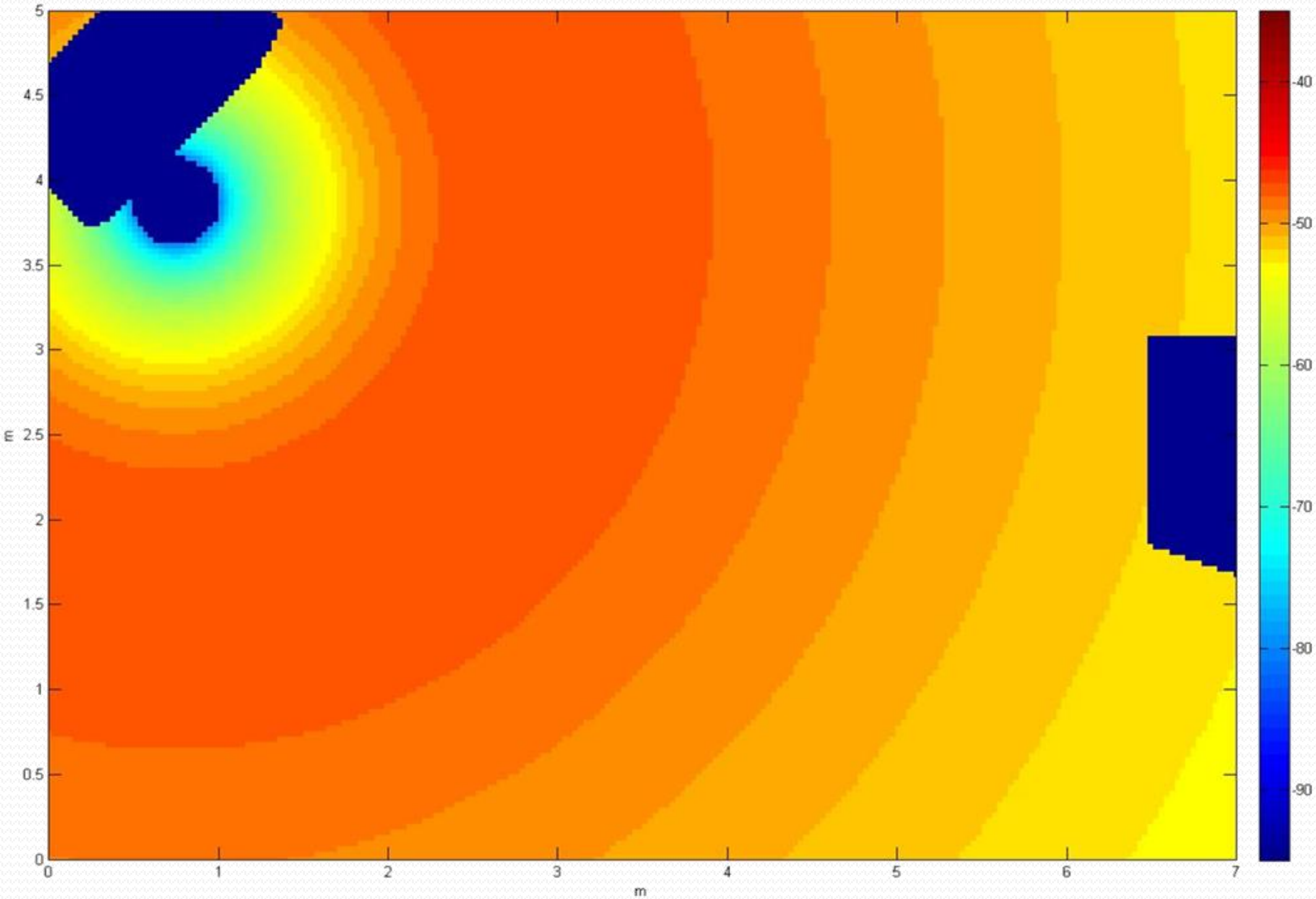
Received signal strength

- Ideal situation
  - (courtesy of F.Potortì, A.Corucci, P.Nepa, F.Furfari, P.Barsocchi1, A.Buffi)



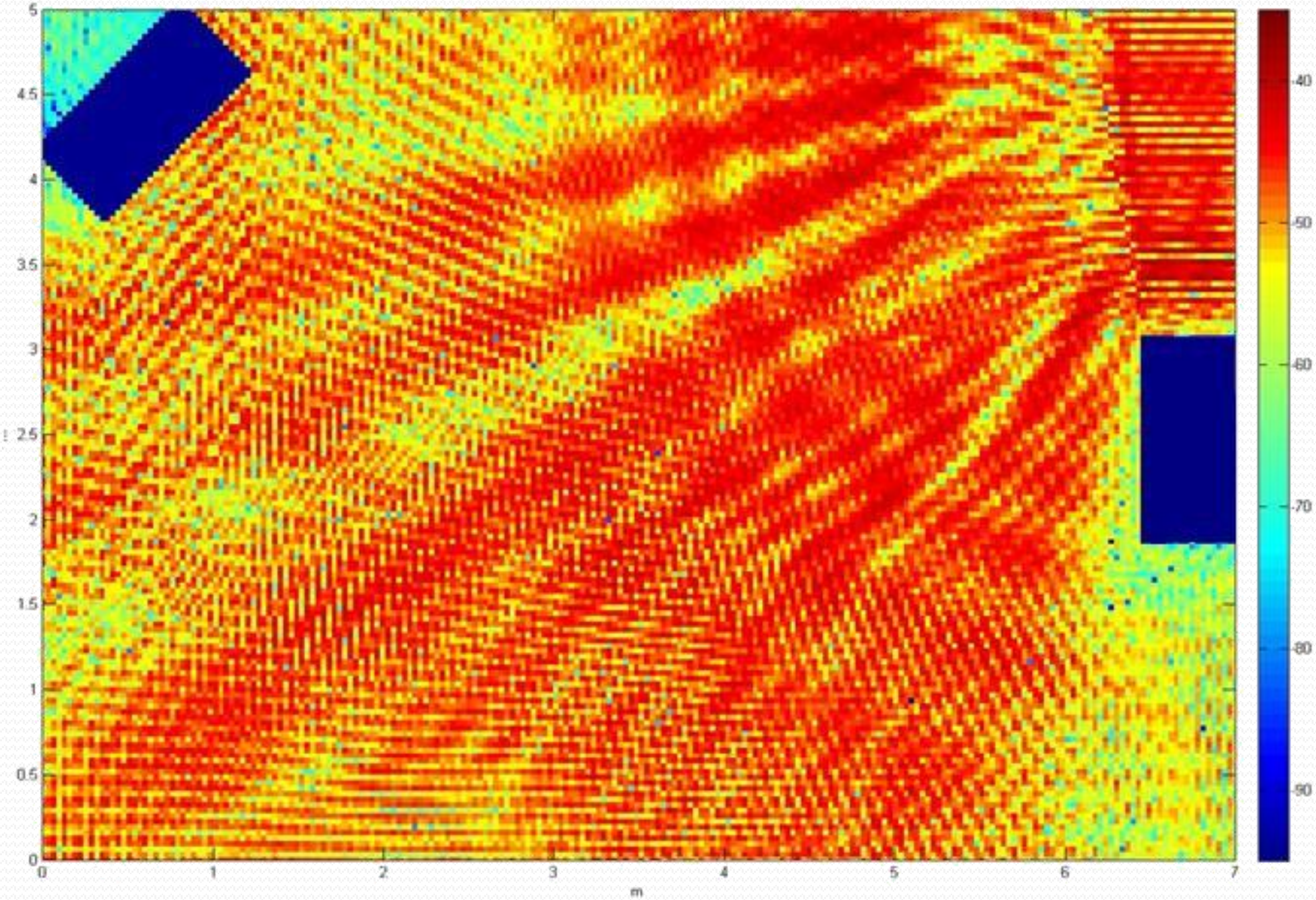
Received signal strength

- Ideal situation:



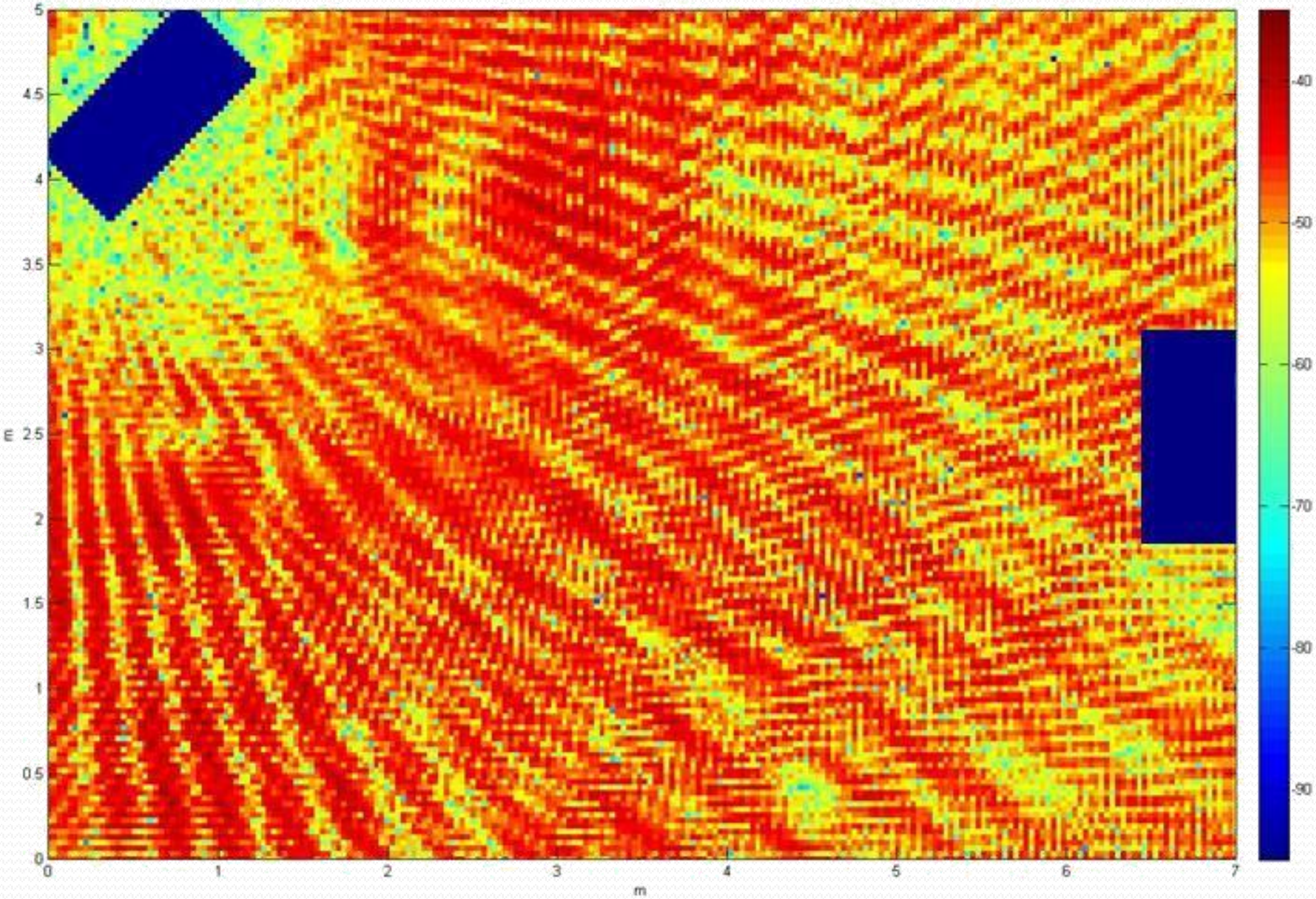
Received signal strength

- Realistic situation (with 3° order reflections):



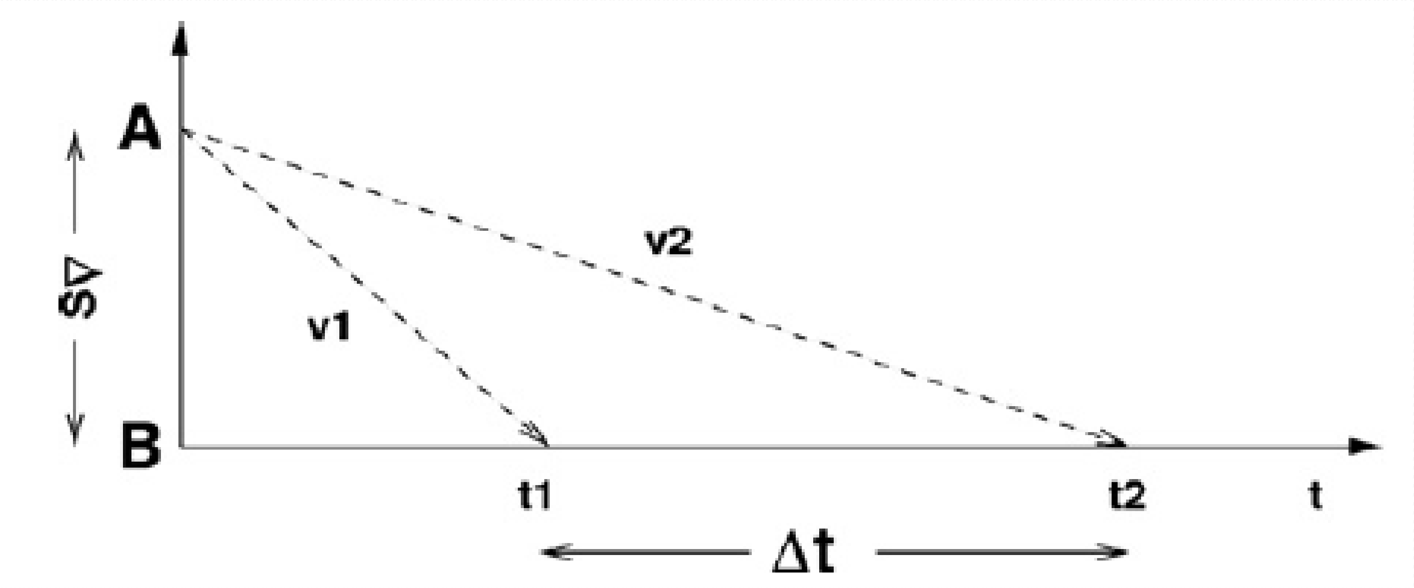
Received signal strength

- Realistic situation (with 3° order reflections):



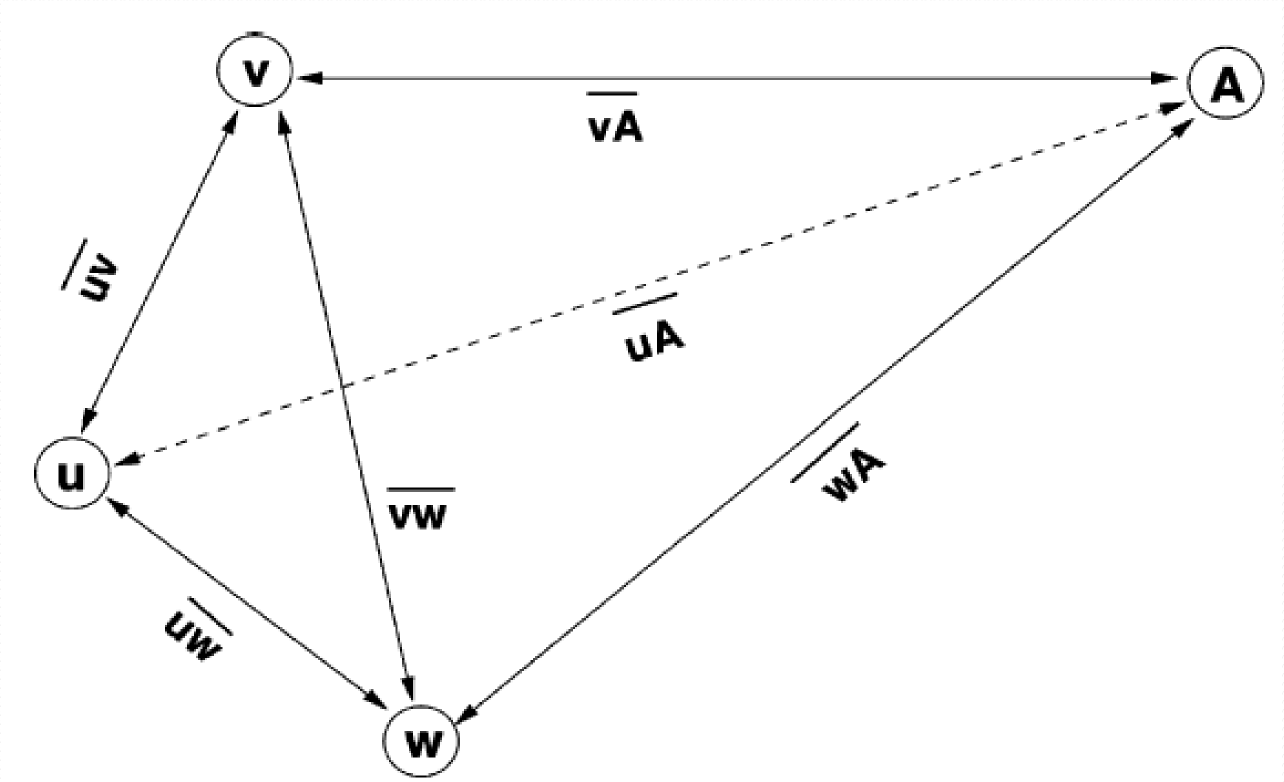
Range techniques

- Time of different arrival:
  - Uses two different kind of signals (e.g. radio and audio), measures the difference in the time arrival and estimate the distance based on speed of signal propagation



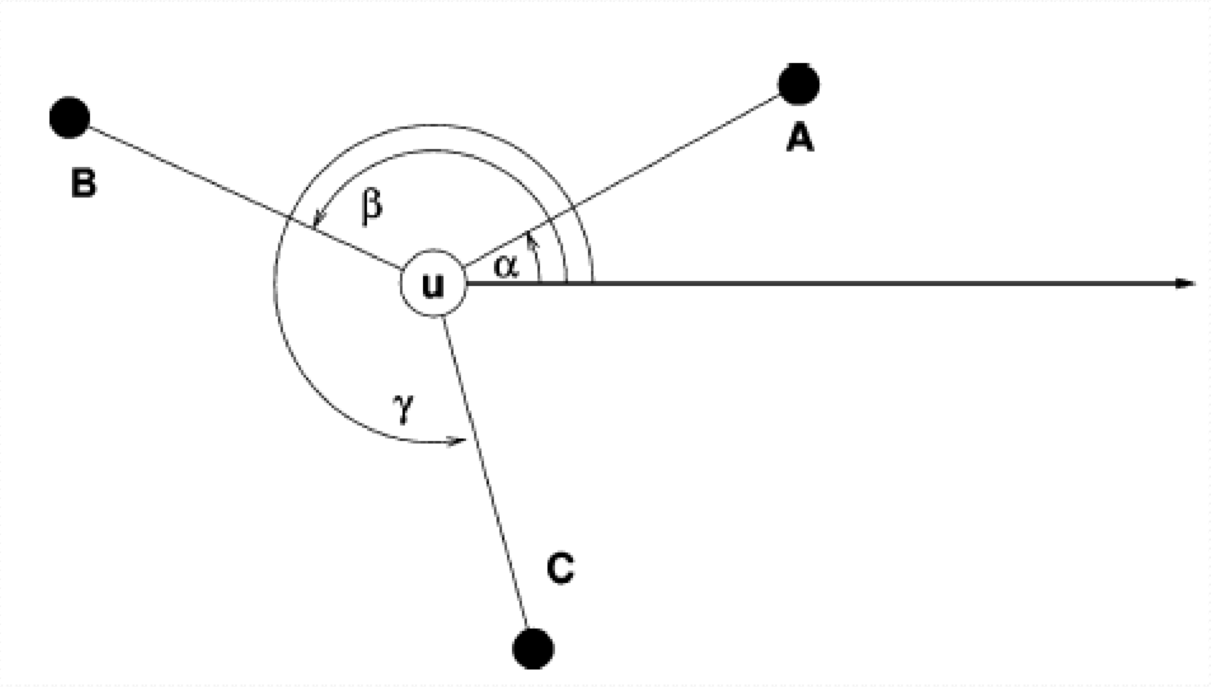
Range techniques

- Once the relative distances of nodes are known estimates the relative positions
  - Upon receiving distances vA and vw from v and wA from w, node u estimates its distances to v (uv) and w (uw) and uses trigonometry to estimate its distance to A (uA)



Range techniques

- Angle of arrival:
  - Uses directional antennas to estimate the angle of arrival of the incoming radio signal
  - Node u measures the angle of arrival of messages received from nodes A, B and C as α, β and γ according to a local angular system



Range techniques

Drawbacks:

- Need special antennas to estimate angle of arrival
- Signal strength or time of arrival techniques may be affected by external perturbations
  - Interferences
  - Walls/obstacles
  - Multipath
- Evaluation of the coordinate system accuracy

Range-Free Techniques

Virtual coordinates

- Unrelated to the physical coordinates of the sensors
- Typically based on hop distances
- Can support efficiently geographical routing
  - With sparse networks might be better than physical coordinates
- Correspondence between virtual and physical coordinates left to the sink node

Sensor Coordinates without GPS

Sensor Coordinates without GPS

- Geographic routing without location information (Rao et al.) MOBICOM 2003
- Investigates three cases:
  - Perimeter nodes are know & they know their location
  - Perimeter nodes know they are on the perimeter but they don't know their location
  - Nodes know neither their location, nor whether they are on the perimeter
- For each case they give a protocol which assign virtual coordinates

Sensor Coordinates without GPS

Perimeter nodes are known & they know their location

- Given node i let:
  - N_i be the set of its neighbors
  - x_i its x-coordinate
  - y_i its y-coordinate
- Initially each node (except perimeter node) is assigned coordinate (100,100)
- Node i approximates its virtual coordinates iteratively:
  - $x_i = \mathrm{SUM}(x_k : k \in N_i) / \# N_i$
  - $y_i = \mathrm{SUM}(y_k : k \in N_i) / \# N_i$
- The iteration is repeated d times

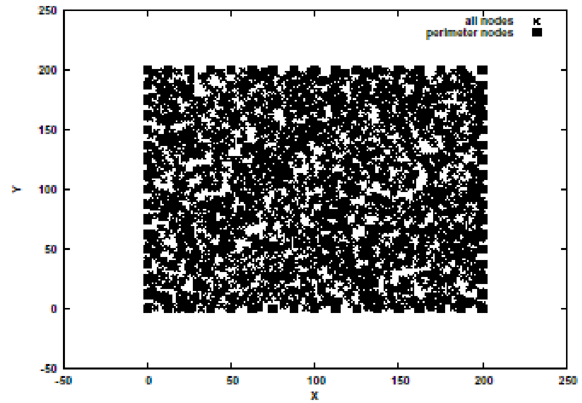
Sensor Coordinates without GPS

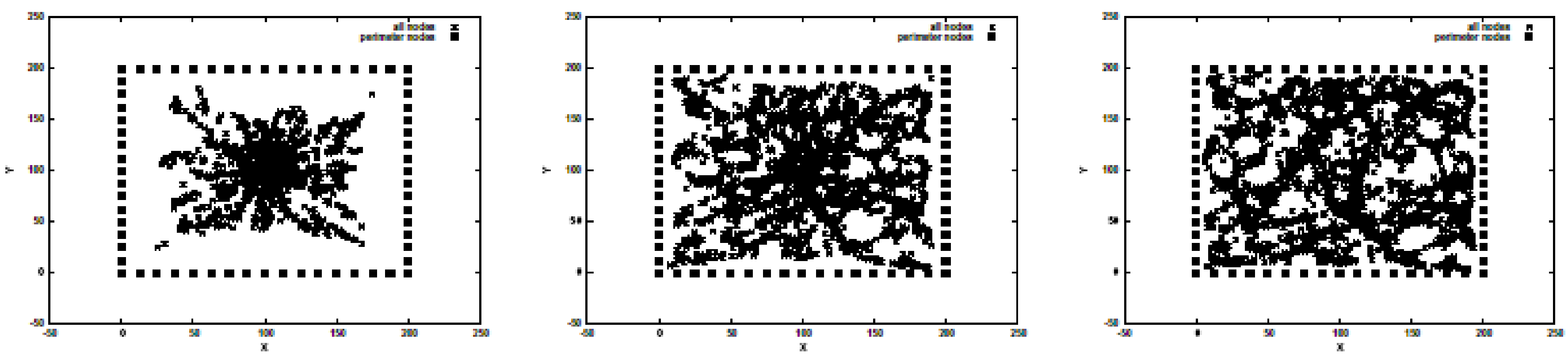
Perimeter nodes are known & they know their location

- After d iterations evaluate:
  - success rate of greedy routing over the virtual coordinates
  - Average path length

Initial position of nodes

a) After 10 iterations
b) After 100 iterations
c) After 1000 iterations





(a) (b) (c)

Sensor Coordinates without GPS

Perimeter nodes are known & they know their location

- Simulation with d=1000 (and 16 neighbors per node)
  - Virtual coordinates:
    - Greedy routing success rate: 99,3%
    - Average path length: 17.1
  - Physical coordinates:
    - Greedy routing success rate: 98,9%
    - Average path length: 16,8
- It is not necessary that all perimeter nodes participate to the protocol
  - If only 8 perimeter nodes participate:
  - Greedy routing success rate: 98,1%
  - Average path length: 17.3

Sensor Coordinates without GPS

Only perimeter nodes are known

1. Each perimeter node broadcasts an HELLO message to the entire network
   - Each perimeter nodes knows its hop distance with the other perimeter nodes
   - This vector distance is the perimeter vector
2. Each perimeter node broadcasts its perimeter vector to the entire network
   - Each perimeter node knows the hop distance between any pair of perimeter nodes

Sensor Coordinates without GPS

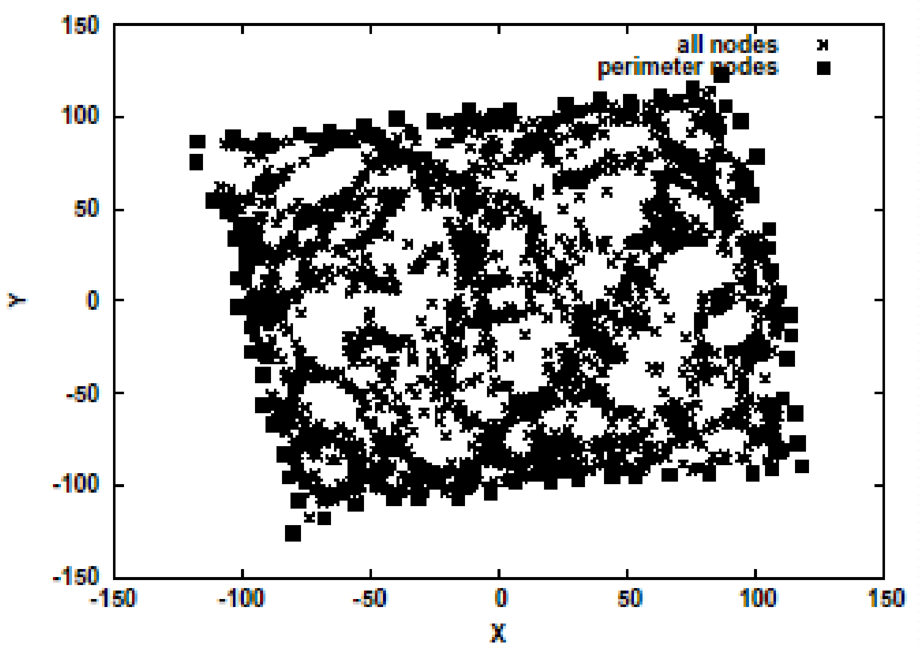
Only perimeter nodes are known

3. Each perimeter node computes a triangulation to compute the virtual coordinates of the other perimeter nodes
   - Such as to minimize
   - SUM $_{i,j \text{ in the perimeter}}$ (hopdistance(i,j) – dist(i,j))^2
     - dist(i,j) is the euclidean distance over the virtual coordinates
     - hopdistance(i,j) is the distance computed in phase 1
4. Then the previous protocol is applied
   - However the nodes can be assigned initial coordinates taking into consideration the information available from the previous steps

Sensor Coordinates without GPS

Only perimeter nodes are known

- With d=10 (and 16 neighbors per node) achieve same performance than previous protocol with d=1000
  - Greedy routing success rate: 99,2%
  - Average path length: 17.2
  - Due to a better initialization of non-perimeter nodes

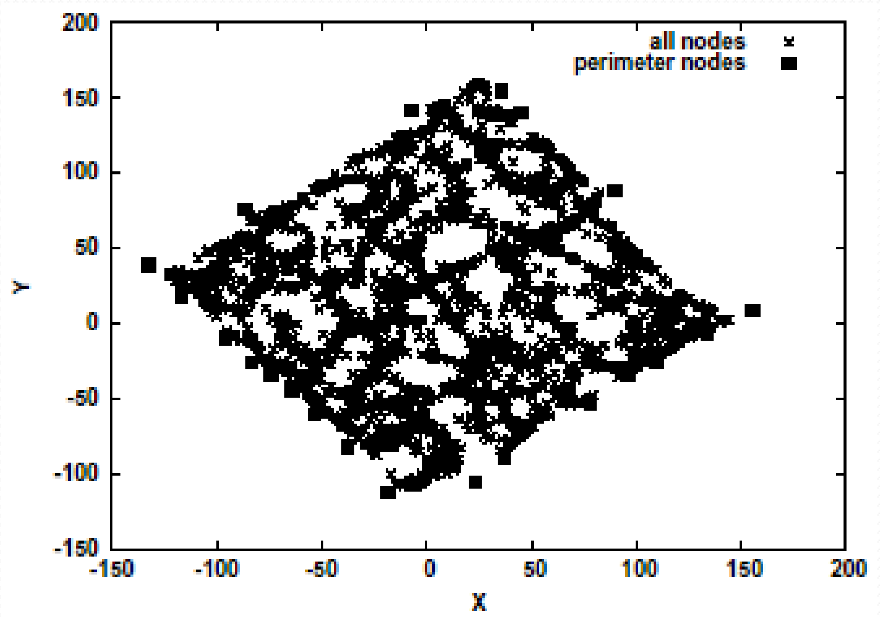


Sensor Coordinates without GPS

No Location Information

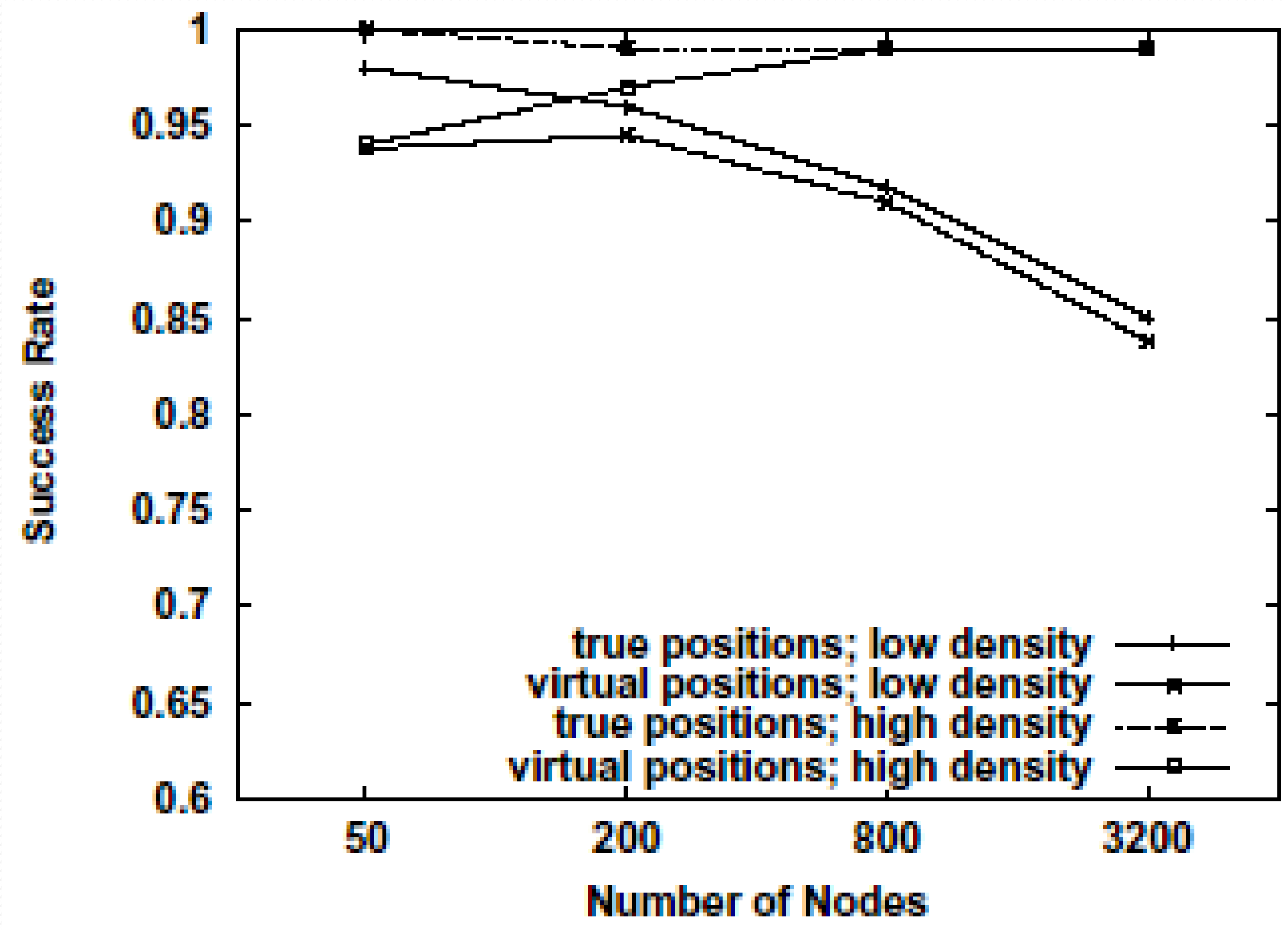
- Add a preliminary phase:
  - Two bootstrap nodes broadcast a beacon
  - The nodes that, within two hops, are the farthest from the bootstrap nodes are classified perimeter node
  - Applies the same protocol as before
- With d=10:
  - Greedy routing success rate: 99,6%
  - Average path length: 17.3

Example of virtual coordinates



Sensor Coordinates without GPS

Success rate of greedy routing with virtual and physical coordinates



Sensor Coordinates without GPS

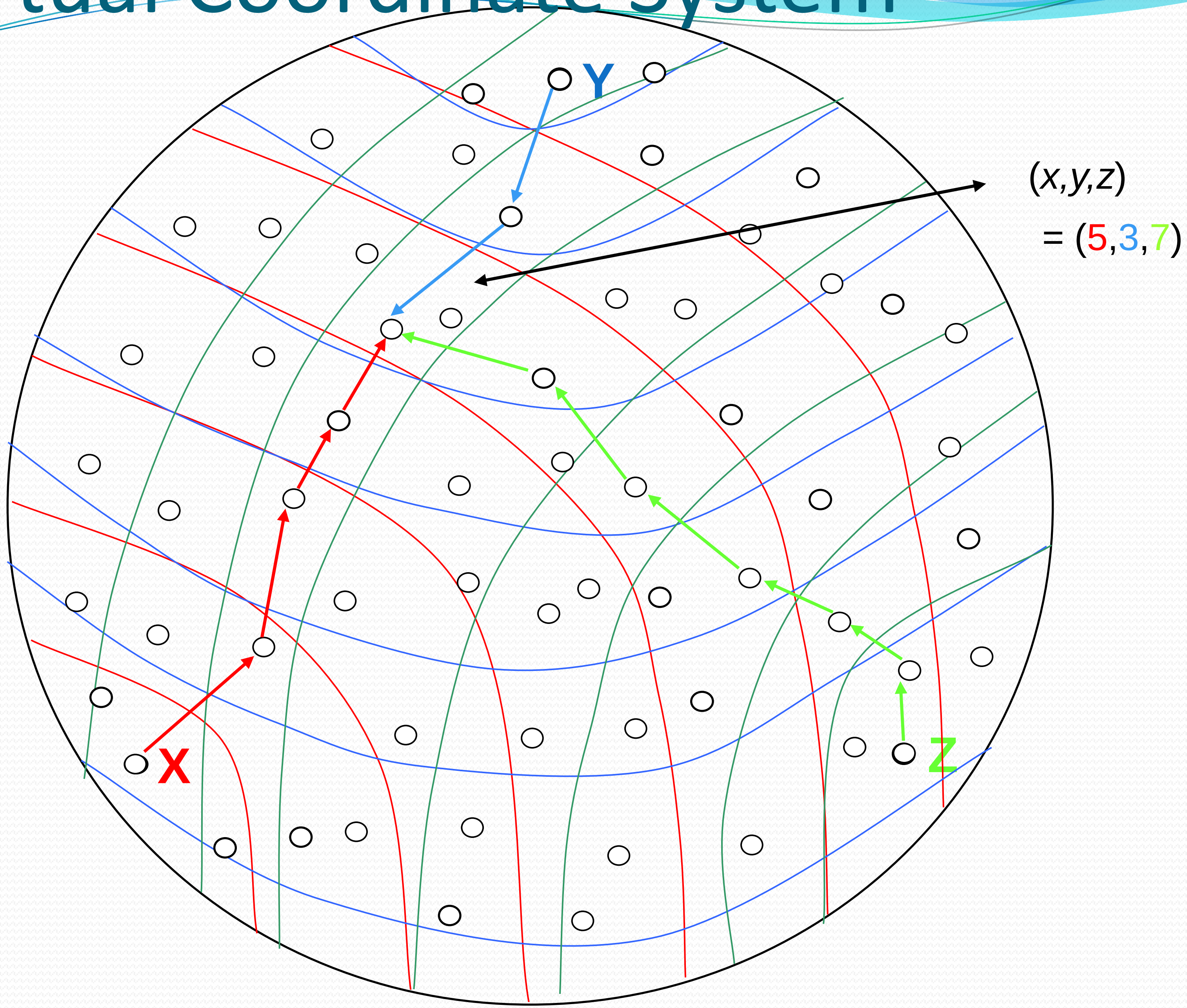
- The protocols are resilient to message losses
  - Due to redundancy of information in the perimeter vectors
- The virtual coordinates can be mapped onto a circle
  - Gives well defined area for implementing a distributed hash table

Virtual Coordinate Assignment protocols

Virtual Coordinate System

- VCAP, Infocom 2005
- A *virtual coordinate system* is defined by electing three *anchor* nodes (X,Y,Z).
- Each coordinate is a triplet (x,y,z), x is the minimum hop distance between the node and the X anchor.
- The coordinates are virtual because they are not related to the physical position of a node (Euclidean coordinates).
- Different nodes may have the same coordinates.

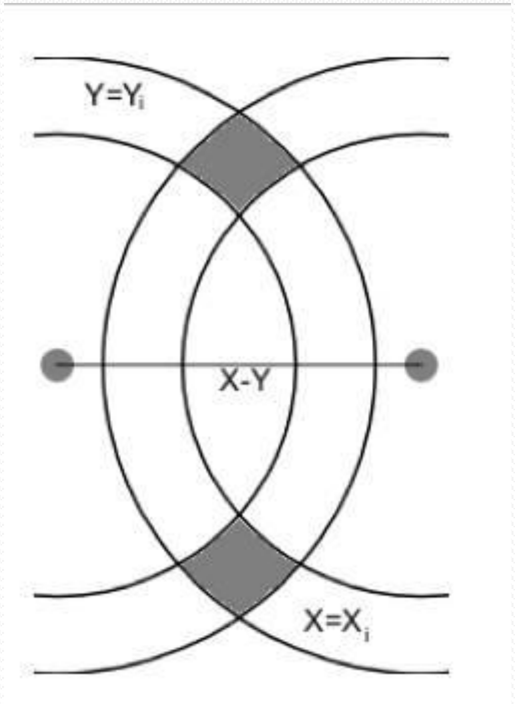
Virtual Coordinate System



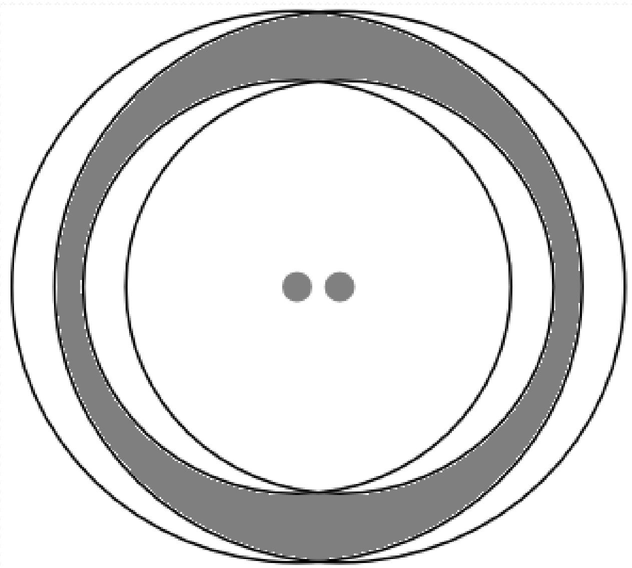
Virtual Coordinate System

Why three anchors?
Why on the border?

- Otherwise the coordinate system may be inconsistent, that is, distant nodes may have the same coordinate.



a) Inconsistence with one coordinate



b) Anchors too close: distant nodes share the same coordinate

Virtual Coordinate System

The zones

- With the virtual coordinate system, the routing protocol exploits geographic routing to reach the destination’s zone and proactive routing within the zone.
- Hence, in order to efficiently support routing, the size of a zone must be small.
- The size of the zones depends on the position of the anchors and on the network density
- The zones are minimized if the anchors are as far as possible from each other
    - The anchors must be close to the network border

Virtual Coordinate System

Virtual Coordinate Assignment Protocol

- The sink node begins the VCap protocol
  - It broadcasts a W_SET message containing an hop counter to identify nodes on the network border
  - Nodes in the network use the hop counter in the W_SET message to determine their hop distance from the sink
- Anchor X is elected by a broadcast-based distributed protocol
  - Nodes with maximum hop distance from the sink (in their local neighborhood) are eligible as X.

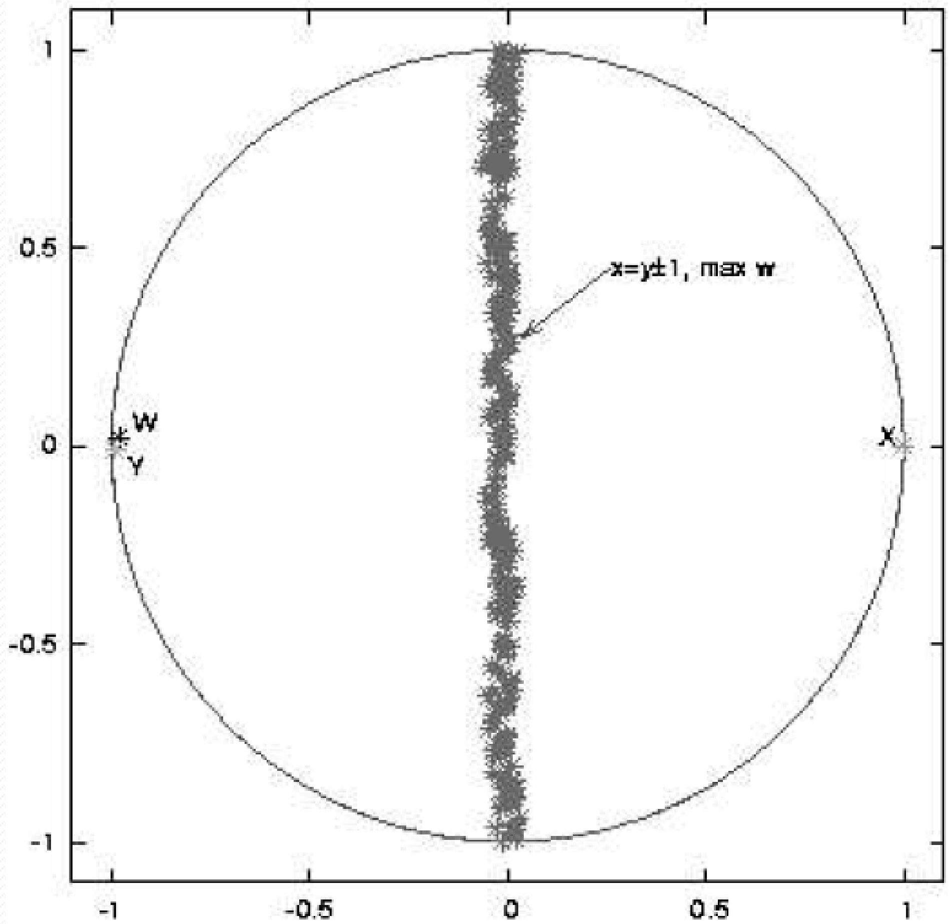
Virtual Coordinate System

Virtual Coordinate Assignment Protocol

- The election of anchor X:
  - Among the nodes with maximum hop distance from the sink, it is elected as X the node with maximum ID.
- The election of anchors Y and Z exploit a similar protocol
  - Heuristics enforcing the property that X, Y and Z are as far as possible from each other are used.
    - Example: Z should be on the border and equidistant from X and Y

Virtual Coordinate System

Virtual Coordinate Assignment Protocol



Feasible positions for anchor Z

Routing over virtual Coordinates

- The coordinate system is 3D
  - No GPSR or similar
- Mainly greedy strategies
- If greedy fails:
  - Use heuristics such as: move to the anchor nearest to the destination
  - OR use CLDP

Routing over virtual Coordinates

- The performance of greedy can be improved using coordinate smoothing
  - the coordinates are averaged with the neighbors‘ coordinate
  - Increases the resolution of the coordinate system
    - By reducing the size of the zones
  - Improves the performance of greedy routing
  - Reduce the effect of errors (due to packet losses) in the coordinate system
- The coordinate system can be embedded in a 2D coordinate system to support GPSR
  - The actual embedding strategy is important, the coordinates properties should be preserved

Virtual coordinates performance

- Depends on the actual routing strategy
- Greedy on 3D coordinates without smoothing provides a poor performance
- However with suitable heuristics the reachability approaches 100% as the network density increases
  - It cannot guarantee 100% delivery unless GPSR or CLDP are used
  - Reachability with virtual coordinates is comparable to reachability with Euclidean coordinates
    - In some cases virtual coordinates perform better

Conclusions

- Up to now most of these solutions are confined to the academy
  - Static configuration of the network
  - The virtual coordinates need to be assigned a priori
  - With mobility, failures, join and disconnections of nodes the virtual coordinate system degrades rapidly
  - Guaranteed delivery can be achieved at a high cost (CLDP)