Lecture on Sensor Networks

Communication in sensor networks

SMACS and EAR

SMACS (Stationary MAC and Startup procedure) builds local clusters similar to SMAC and T-MAC. The frame times of members of a common cluster are synchronized. They are divided into smaller time slots.

The nodes in a 1-hop neighborhood get to know each other during an initialization phase and agree on two time slots, one for each direction of the communication. The intention of the assignment of pairs of time slots is to achieve a collision-free allocation of mutual sending times.

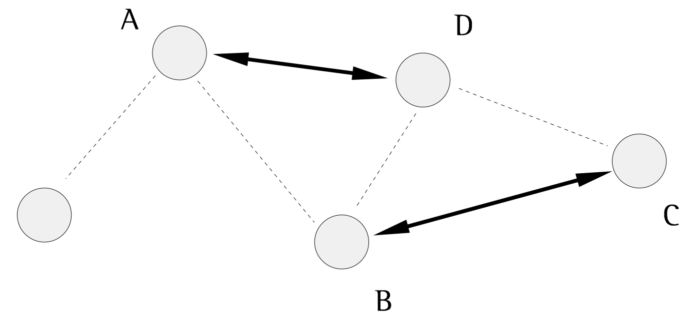
Problem: If a node allocated many time slots to its neighbors, there would be hardly any left for the communication between the other nodes. With n nodes we get n x (n-1) directed connections altogether.

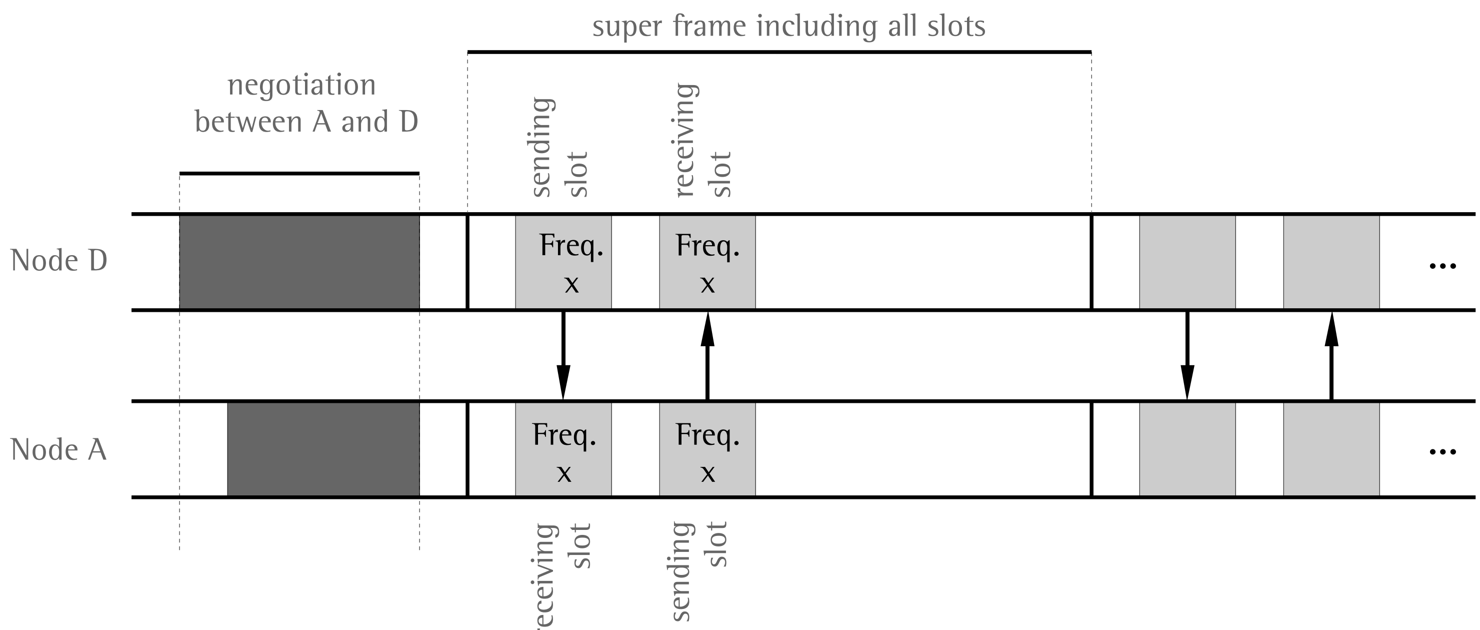
SMACS Solution: We assume that nodes have many channels available. Each pair of nodes chooses a pair (frequency, time slot), which has not been chosen yet by another pair of nodes. Coupling time slots with frequencies extends the space of communication possibilities.

Communication in sensor networks

SMACS and EAR

All nodes within a common sending range build communication pairs and agree pairwise on a common frequency. The sending slot for one node is the receiving slot for the peer node respectively.

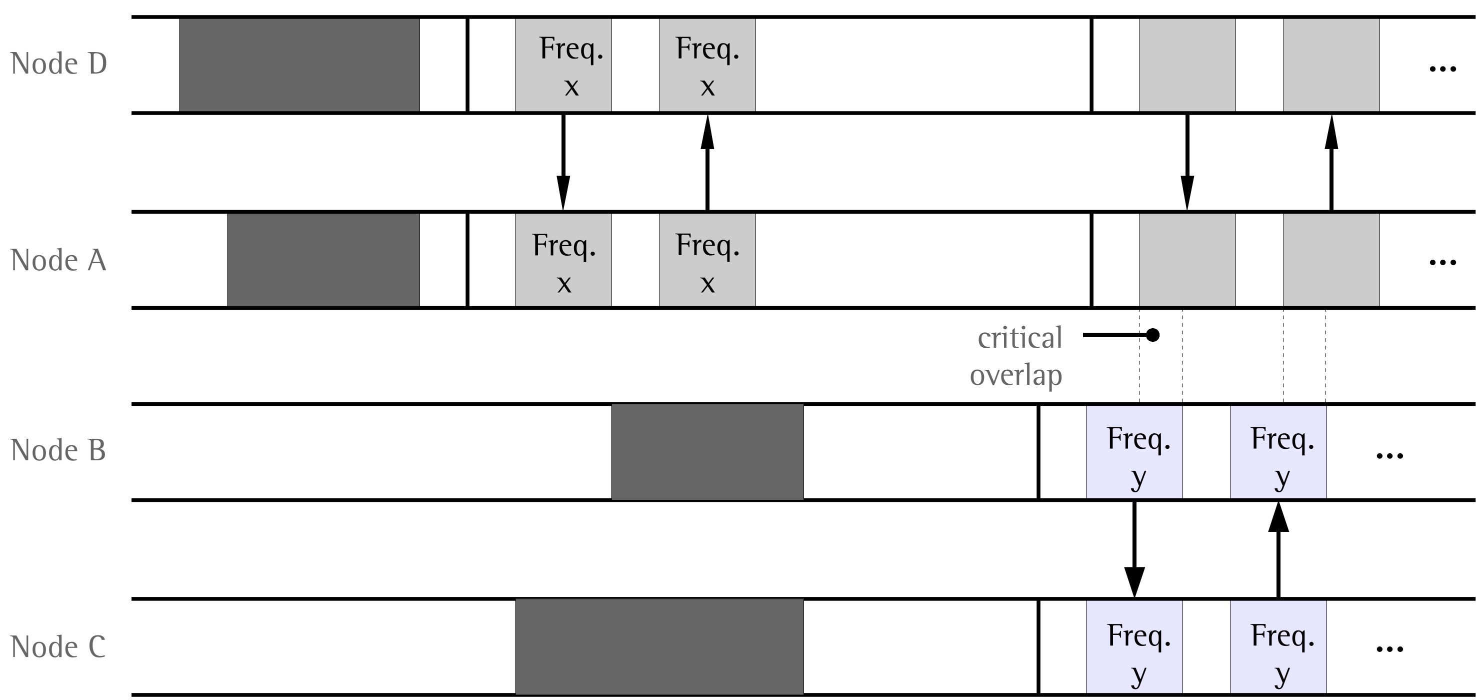




Communication in sensor networks

SMACS and EAR

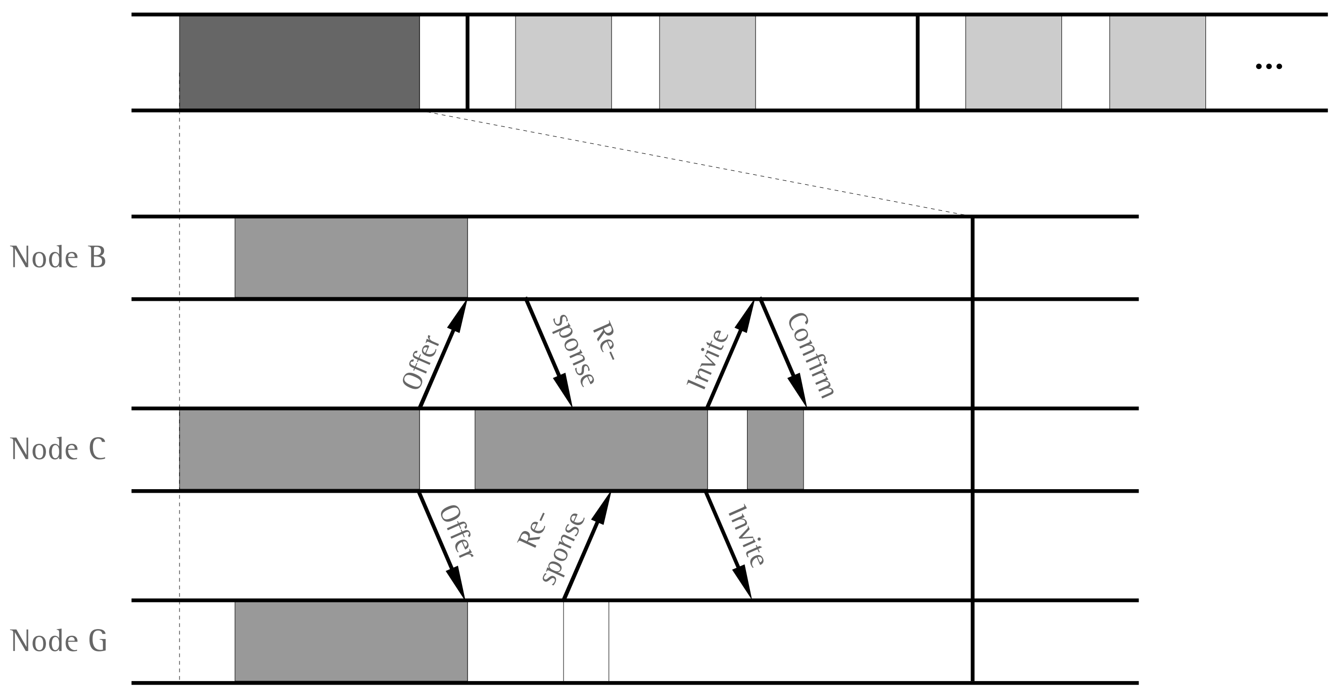
Due to overlaps in time between two clusters it is necessary to choose pairwise different frequency bands.



Communication in sensor networks

SMACS and EAR

Negotiation of slots and frequencies between three involved nodes B, C and G. B and C establish a connection while G loses the negotiation and has to wait for other invitations or invite someone itself.



Communication in senor networks

SMACS and EAR

Initialization of SMACS

Invitation offer	Nodes B, C and G are waiting to get invited to a group. C's random timer elapses first as can be seen on the previous slide. Thereupon C sends an invitation offer to all other nodes. The invitation contains its ID.
Response to invitation	Nodes G and B receive the message. After a random time they respond to the offer including their own ID and number of already connected nodes.
Actual invitation	After C waits for responses for a response interval it eventually picks a node, e.g. the one with a good radio signal or with the most connections and sends the actual invitation, including the ID of the invitee (node being invited) and its own schedule, containing its free and occupied slots. All other nodes receiving the invitation can tell from the ID that they are not the chosen ones. Furthermore, the invitee will adopt the start of the inviter's super frame if it has not yet adopted another one.
Confirmation	The invitee chooses a slot and frequency which does neither interfere with its own schedule nor with the one received and sends it back as a confirmation.

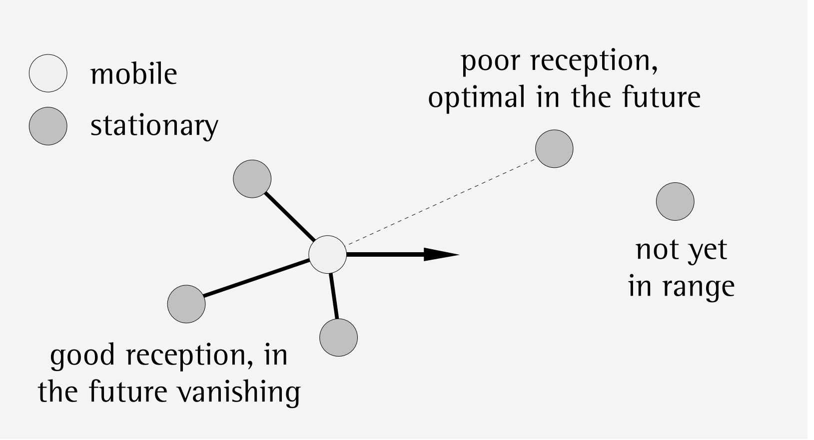
Comment: If the invited node has no slot entry yet the inviter can choose one on its own and add this information to the response (no negotiation necessary).

Communication in sensor networks

SMACS and EAR

Initialization of EAR

EAR (Eavesdrop-And-Register) is an extension of SMACS for mobile nodes.



After their initialization, the stationary nodes send an offer which is addressed to possibly existing mobile nodes.
The message is of the same type we got to know previously and is sent in larger intervals.
These mobile nodes listen passively for a while and collect potential partner nodes in a list.
The quality of the radio connection will be stored for every known station and can be updated while the node is moving.

Communication in sensor networks

SMACS and EAR

Initialization of EAR

As mobile nodes are only present for a short time, the initiative to join the network ought to origin from them. If such a node wants to join it sends a Mobile Invite to the stationary node of its choice including its schedule of free and occupied slots. After a confirmation by the static station of the type Mobile Response the connection is established if a common free slot was found.

If the mobile node is about to move away too far and the signal of another stationary node gets more dominant, the mobile node leaves with a Mobile Disconnect in favor of the new stationary node.

If the disconnect request gets lost, a static station is allowed to delete a mobile station from the list after a longer timeout.

Communication in sensor networks

Geographic Hash Tables

Principle: Data collected in the sensor network is kept (for a while) in the nodes. Only an explicit request provides information. Otherwise no effort for the transmission arises. The key to the information is a global coordinate. The nodes themselves are not addressed (personally). So the sensor network contains only information of the type:

global coordinate 1: date 1, date 2, ...
...
global coordinate n: date m-1, date m, ...

Result: (a lot of) information can be associated with a coordinate.

Communication in sensor networks

Geographic Hash Tables

Sensor nodes are usually not at a particular place and nodes can change their location. What happens with the information associated with the particular coordinates?

- Will information be lost, if nodes drop out at one place?
- How does a request to the network reach a certain place?
- Furthermore: The nodes (routers) should not have to save status information, if possible.

Information in the network can be...

...saved with `Put(coordinate, date)` and
...requested with `date = Get(coordinate)`.

In order to function the request has to reach the right place first. This happens using two different routing modes, namely Greedy Mode and the Perimeter Mode.

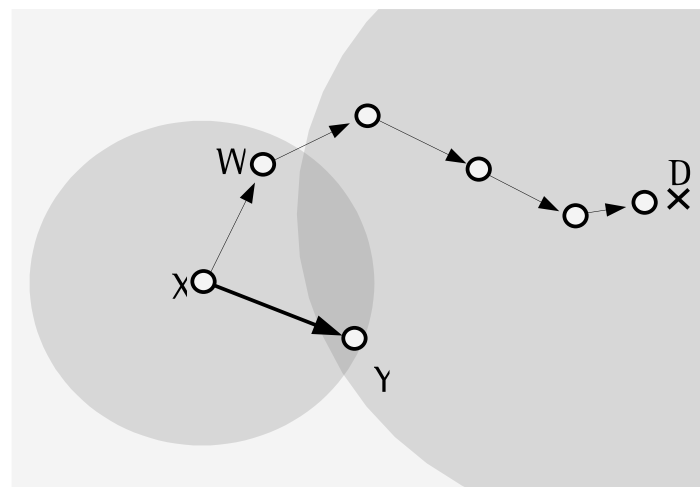
Communication in sensor networks

Geographic Hash Tables

Greedy Mode

Every node knows its neighboring nodes within its sending range (i.e. is also able to hear them) and their positions. Greedy forwarding means that a packet is always sent to the neighbor which has the shortest distance to the destination.

In densely populated networks this often yields a good path to the destination.
In sparsely populated networks, in particular if many nodes already dropped out, the direct way can lead into a dead end. In this example node Y was chosen for its shortest distance to the destination among the nodes in the range of X.



The only possible way to the destination however is node W.

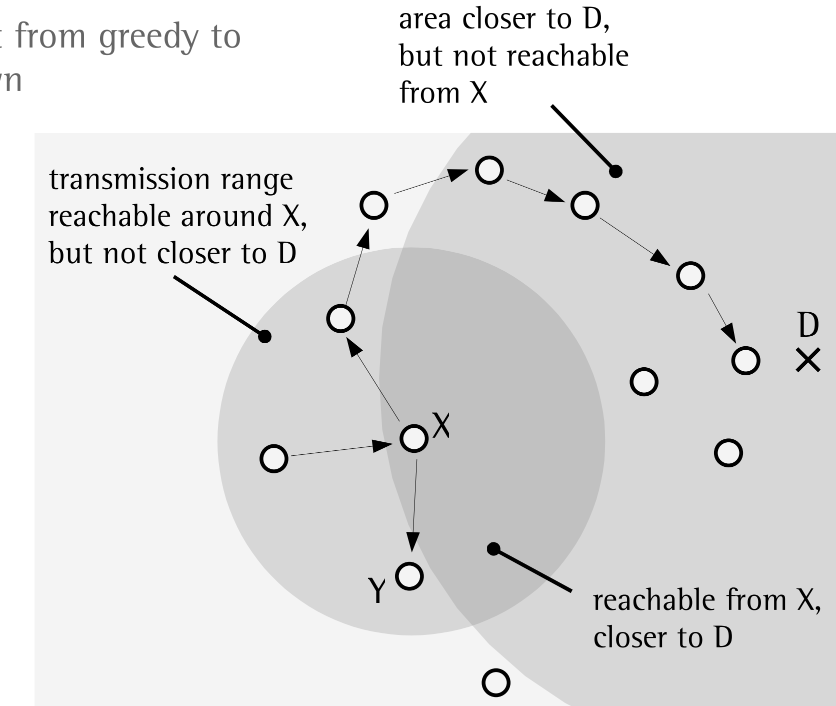
Communication in sensor networks

Geographic Hash Tables

Perimeter Mode

If getting nearer to the destination by forwarding a packet to another node is no more possible, a node has to switch to the so-called Perimeter-Mode. In this example this is what node X does.
It switches the mode-flag of the packet from greedy to perimeter and additionally writes its own position into the packet-header.
Then (depending on the convention) the packet is forwarded counter-clockwise to the next neighbor (see next slide). As soon as the distance to the destination is smaller than the distance between X and D, the packet's flag can be switched back to greedy mode.

Why?



Communication in sensor networks

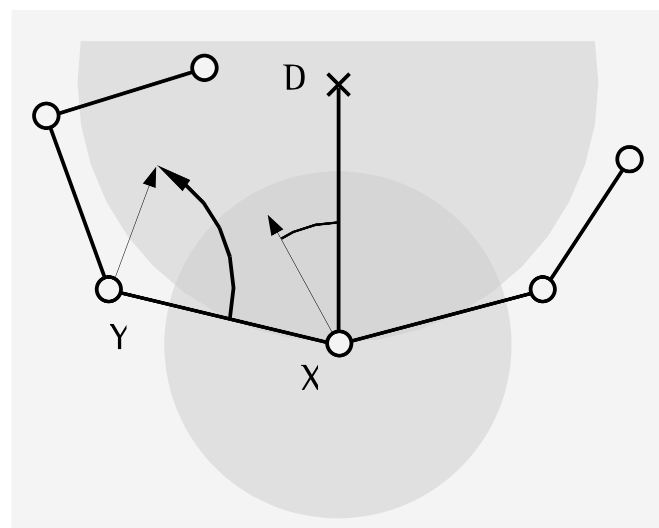
Geographic Hash Tables

Perimeter Mode

This can be done without any risk as the packet's distance to D is yet shorter than it used to be at node X. As a consequence, it will never be sent back again. If the packet however stays in perimeter mode, a cycle will be generated in either case. In this case it returns to the node X, which recognizes that it receives a packet in perimeter mode, which also has its origin in X itself according to the packet header. It is clear that it can not be delivered closer to node X.

Perimeter Start:

If a node recognizes that all its neighbors are further away from the destination D than itself, it switches the packet into perimeter mode and saves its own position in the packet header. While looking at the destination (= vector XD), we now search a neighboring node, e.g. counterclockwise. If there are several alternatives available, the one with the smallest angle to the vector XD is chosen. Rotations with more than 180 degrees are possible.



Communication in sensor networks

Geographic Hash Tables

Perimeter Mode

Regular Forwarding in Perimeter-Mode:

Starting from the recently reached vector Y, the side YX is again rotated counterclockwise until the next node is hit. If there is a node in the 1-hop neighborhood and closer to destination D than node X, the packet can be switched back to greedy mode.
If, in contrast, the packet returns back to node X via a link which has not yet been traversed, X will realize that it switched the packet into perimeter mode and concludes that no other reachable node has a shorter distance to the destination.

Keep in mind that absolutely no status information has to be kept in the nodes themselves.
If, however, the packet returns back over the same edge, on which it was already sent by X, it traveled through a dead end and will pass X like a normal node as a consequence (in order to escape the dead end).
Hint: The Perimeter-Mode only works in planar graphs.

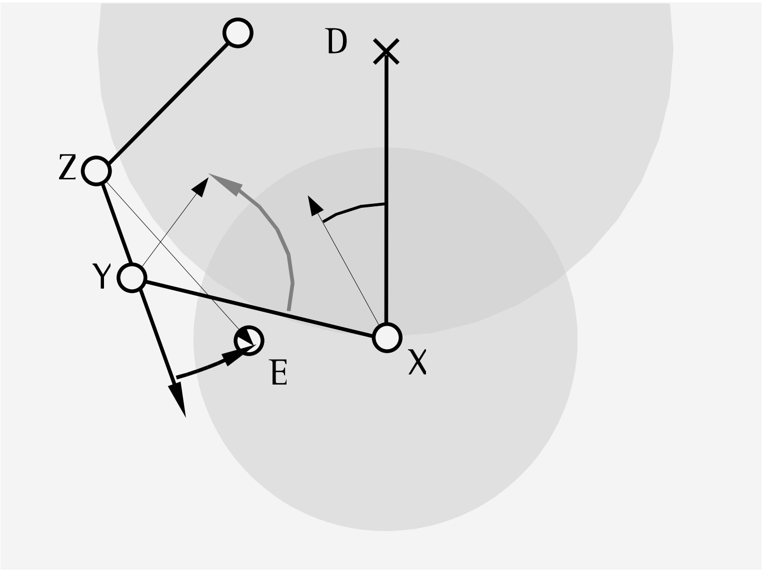
Communication in sensor networks

Geographic Hash Tables

Perimeter Mode

Problems with non-planar graphs

In this scenario another node E was added. This does not change the path of the packet between X and Y. Also the next choice of node Z continues as expected. But then the straight line through Z and Y is rotated counterclockwise according to the algorithm. This time, however, node E is hit first which is chosen as the next hop for the packet. The next step of the iteration will yield node Y again and creates a cycle Z -> E -> Y -> Z.



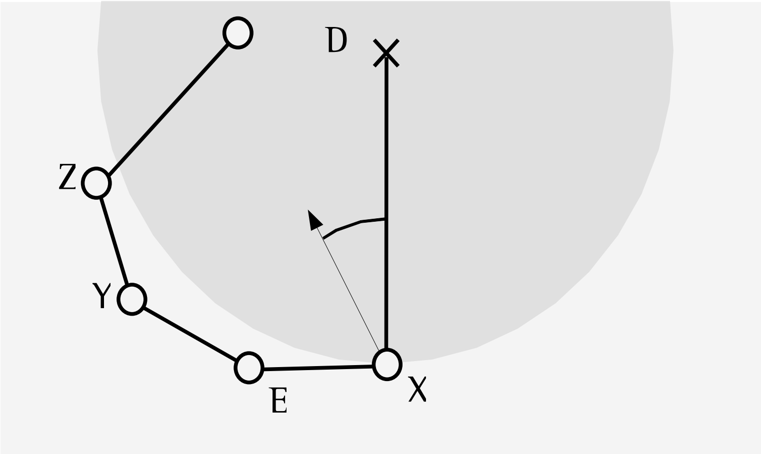
Communication in sensor networks

Geographic Hash Tables

Perimeter Mode

Problems with non-planar graphs

As a result it can be concluded that not every node in the neighborhood must be chosen. In fact the graph reflecting the natural connectivity has to be reduced to a planar graph before the choice of a node. In the example on the right the nodes are limited to a less densely meshed graph so that cycle free routing is achieved.



Communication in sensor networks

Geographic Hash Tables

Perimeter Mode

Problems with non-planar graphs

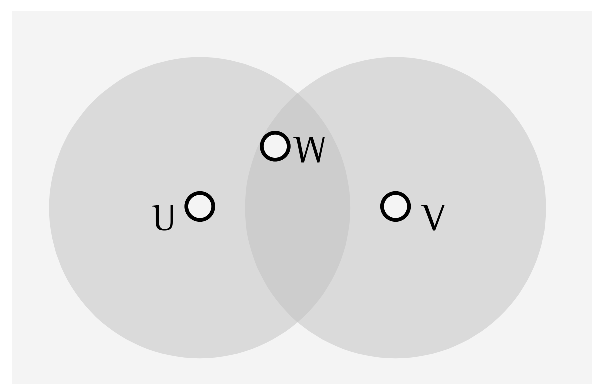
For all connections between two nodes U,V the following constraint has to be checked:

First of all the distance between U and V has to be calculated. Next, an iteration over all nodes will take place. For every node W it has to hold true that either the distance to node U or to node V must be larger that the distance between U and V or in more formal terms:

$$\forall w \neq u,v : d(u,v) \leq \max[d(u,w), d(v,w)]$$

If this can not be ensured the connection between U and V is deleted.

A visualization is provided on the right.
The two circles reflect the radio ranges of the nodes U and V. The node W inside the intersection would provide a second path from U to V. In this case the direct path between U and V will be deleted.



Communication in sensor networks

Geographic Hash Tables

Perimeter Mode

Problems with non-planar graphs

We can conclude that a larger number of short connections is preferred over a single direct connection. Intuitively the short connections are less likely to cross one another than the long ones.

Each node U can check locally, which local edges are allowed to be part of the subgraph and which have to be eliminated:

From the view of a node U the following check has to be made:

```
for all V
  for all W
     if (W != V) and
        d(U,V) > max [d(U,W),d(V,W)] then remove_edge(u,v)
    endif
  endfor
endfor
```

Communication in sensor networks

Geographic Hash Tables

The method present can be used to route packets to a geographical target, or more precisely, to the node which is closest to this target. The basic routing algorithm described before has been published as GPSR (Greedy Perimeter Stateless Routing in Wireless Networks). The following approach is the extension to Geographic Hash Tables, which is in particular using the perimeter-mode to forward information.

With GHT data should be associated with a geographical position. However, it is unlikely that a packet consisting of the tuple (coordinate, date) or a request with only a coordinate actually meets a node at the corresponding location. That is why the packet will always travel to the node closest to the desired location, switch into perimeter mode, orbit the target once and return back to the node, which is closest to the target.

The idea of GHT is to associate the information with all nodes, that are situated around the position on the perimeter. These nodes will be referred to as replica nodes. If a node drops out or if nodes are moving, we quickly find a node in the group of the replica nodes, that take responsibility for the information.

A node feels responsible for the data contained in a packet if it switched the packet to perimeter mode itself. This node is called a home node. It also takes responsibility for further information, which may later be associated with the same position and answers requests that refer to the position.

Communication in sensor networks

Geographic Hash Tables

If the home node gets the packet a second time after it had circled around the target coordinate the node may learn that it will eventually be the home node or that there is a better candidate who takes responsibility. The next slides will explain this in detail.

The term Geographic Hash Table is derived from the fact that information is stored in hash tables inside the nodes. The hash value is the coordinate which is associated with a particular kind of information, e. g. the temperature at the particular place.

Perimeter Refresh Protocol

From time to time the home node lets information circle around the coordinate with which the information is associated. This is basically the same process that takes place if information is associated with a coordinate for the first time.

In the context of this refresh the replica nodes can extend their databases with new information belonging to the particular coordinate.

Communication in sensor networks

Geographic Hash Tables

Perimeter Refresh Protocol (continued)

Change of the home nodes:
A current replica node receives a packet and recognizes, that itself is closer to the target of the information meanwhile than the current home node (whose position can be found in the packet header). It decides to become the (better) home node and writes its own position into the header. The former home node recognizes this when it gets back the packet. It has to accept the fact that there is a better home for the information. As a consequence, it degrades itself to be a replica node.

The home node drops out:
Every replica node updates a timer for every tuple (coordinate, date) in its database. If no refresh was encountered for a tuple for a predefined amount of time it can be assumed that the home node for the information moved away or dropped out. The replica node, whose timer elapses first sends the information with its own position onto the perimeter path. This does not necessarily mean that it automatically becomes a home node, since another replica node could be closer to the position (associated with the information) and could adopt the packet by writing its own location into the header.

Communication in sensor networks

Geographic Hash Tables

Perimeter Refresh Mode

GHT knows three types of Timers:

Refresh Timer T_r:
Every home node has a refresh timer T_r. Once it has elapsed the information is sent around the coordinate associated with the information in perimeter mode. Every node updates the packet if additional data emerges for the position in question and inserts data contained in the packet but not yet in the local database. Then T_r is set back to zero. Newly deployed nodes being on the perimeter mode are updated automatically.

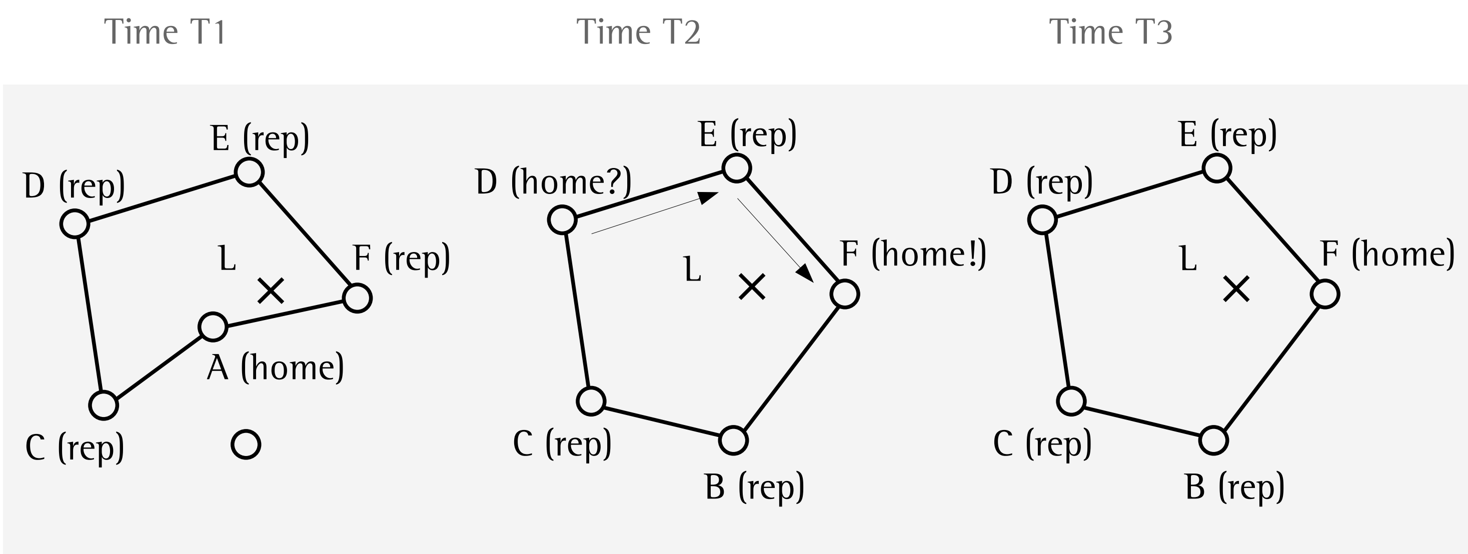
Takeover Timer T_t:
The takeover timer is used only in the replica nodes. If it elapses, the corresponding node will try to become the new home node to the associated information. Basically it applies that: $T_t > T_r$.

Death Timer T_d:
A long timeout is chosen for the timer T_d compared to the other timers. It elapses in a replica node, if it has not seen either its own refresh message or a foreign one.

Communication in sensor networks

Geographic Hash Tables

Example



At time T1 node A is the home node for its proximity to location L. Then A drops out and node D is the first node to become new home node as its takeover timer elapsed first.

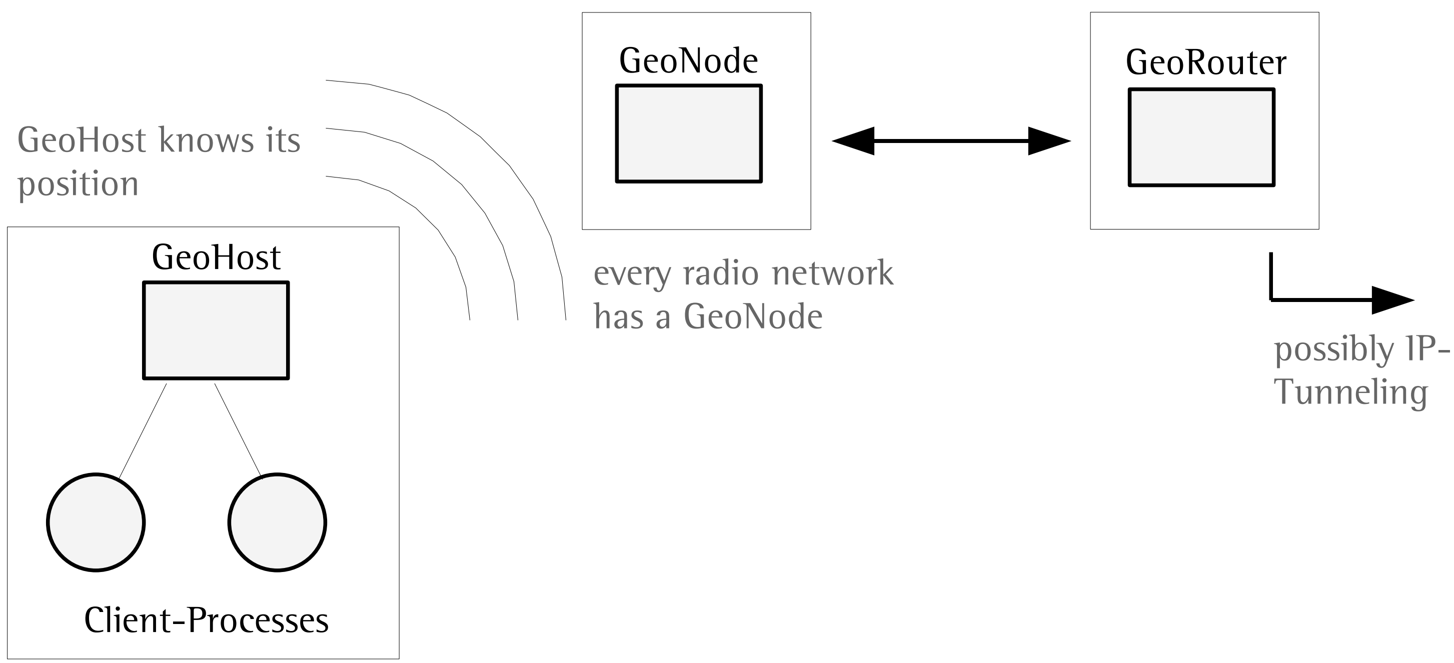
At T2 the packet arrives at F, which enters its position due to its shorter distance to L into the packet header. Later D sees the packet again, but with a shorter distance to L than to D itself, so it gives up its takeover attempt and forwards the packet again. F recognizes that it may become new home node because of the position it formerly wrote into the packet itself.

Communication in sensor networks

Geocast

A former suggestion for IP-based mobile ad-hoc networks in which routing is performed by means of spatial (world-) coordinates (but on the execution level) is known as GeoCast.

Principle: A packet knows its destination region, which can be encoded as circle or any polygon. Routers do not route their packets by IP-addresses, but choose in each case that router whose area overlaps with the area, which is addressed by the packet.



Communication in sensor networks

Geocast

GeoHost

Provides an interface for the client-processes as a kind of proxy. The GeoHost is aware of its geographical coordinates and it knows its responsible GeoNode, which has to be unique.

GeoNode

Accepts the messages upstream and forwards them to the next GeoRouter. Messages are cached downstream and sent repeatedly. Only one GeoNode is responsible per LAN respectively per radio network.

GeoRouter

The GeoRouter knows the polygon for which it is responsible. It consists of the aggregation of the polygons of the underlying GeoNodes or the underlying client-routers. The decision, which father or which child gets a message is detected by cutting the target polygon included in a packet with the area polygon of another router (children or parents in the hierarchy). The actual forwarding work is based on so-called IP tunneling. So routers do not have to be modified. Georouting is an application level routing approach.

Communication in sensor networks

Geocast

Sending Operation:
A client-process asks its GeoHost for the IP-address of the responsible GeoNode from which every host must not have more than one. In the packet the target area is encoded as polygon using degrees of latitude and longitude. The client sends the packet to the GeoNode who forwards it to its responsible GeoRouter.

The GeoRouter checks whether the region of one of its child (-router) overlaps with the target area in the packet. The packet is forwarded to all routers, with a partly overlap between the area included in the packet and the area the router is responsible for.

If the encoded region in the packet is not completely covered by the areas of the child routers, the packet has to be forwarded in addition to a parent node who processes the packet accordingly.

As IP-routers are not aware of GeoCast today, the routing has to take place at the application level of the routers which are connected via a conventional IP-based network. A routing-decision is only made between GeoCast routers. In between, the packets behave like normal IP-traffic. The communication between special routers over a conventional network is called tunneling.

Communication in sensor networks

Geocast

Routing characteristic:
In normal routers a lookup table is enough to detect to which neighboring router a packet with a certain IP-address has to be forwarded. This can happen e.g. via hashing very quickly. Often the free Berkeley DB (a (key, value) pair-oriented simple database) is used for fast lookup in IP routers.

In GeoCast however, regions have to be intersected with one another for every packet which is significantly more complex.

Main difference in the delivery process:
Not a single terminal system but a more abstract region is addressed, in which many, few or temporary no nodes are included. In particular the set of mobile nodes can change at any time.

So it makes sense to define a message for a target region and simultaneously for a time interval. This way new nodes entering the region are able to receive the message.

Communication in sensor networks

Geocast

Main difference in the delivery process (continued)

For this purpose every message arriving at a GeoNode is cached for a lifetime defined by the sender. From time to time the GeoNode sends a list of available messages to a global well-known group address, which is received by every GeoHost. With the knowledge about its current position the GeoHost is able to decide, whether the client-processes get notified. If a client-process is notified and interested in a kind of message, it can join the multicast group and will be supplied with novel information. The role of the Multicast-group stays somewhat unclear in the paper. Obviously not only single packets, but longer streams can also be received - otherwise joining the group would make no sense.

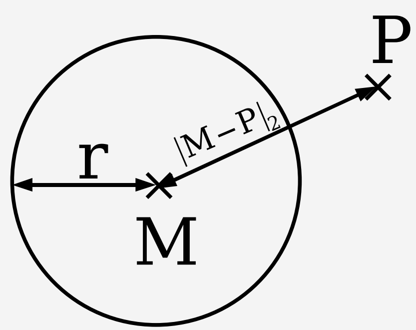
Advantage: The message gets a lifetime, which does not depend on the actual population of an area with receivers. One does not have this problem in end-to-end communication with dedicated participants. In the latter case unsuccessful deliveries can easily be signaled to the sender which is not possible in GeoCast.

Communication in sensor networks

Geocast

Intersection calculations between target- and domain areas:

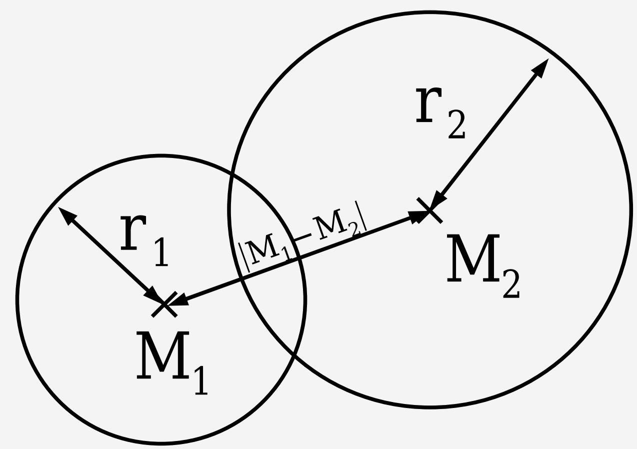
point – circle:



$$|M-P|_2 < r$$

This test makes sense for the GeoHost to detect whether it is in the domain of a GeoNode. The point is not well suited as a target "address" (nobody will be there).

circle - circle:



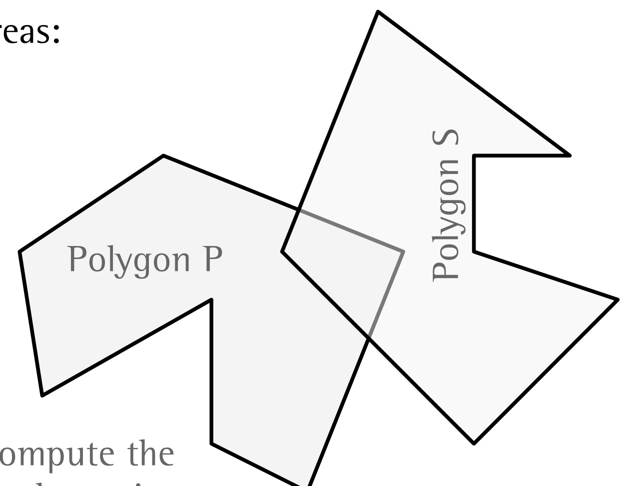
$$|M_1-M_2|_2 < r_1 + r_2$$

Here, one circle could be the domain of a GeoNode, the other one could be the area, a message is addressed to.

Communication in sensor networks

Geocast

Intersection calculations between target- and domain areas:

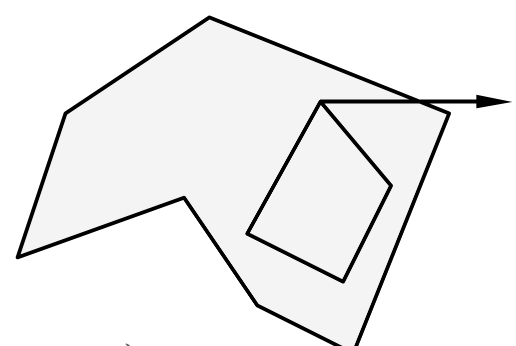


Polygon - Polygon:

(1) Intersection of each edge of P with each edge belonging to S

no intersection?

(2) Is one point from S in P? To determine the answer compute the number of intersections of a horizontal scanline through a point on the boundary of P in one direction. If the number is odd, S is within P, else outside.



No point contained?

(3) Is one point from P in S?

No point contained?

Then the polygons have no intersection. Unfortunately this (common) case requires all complex tests. However bounding boxes can be used as preliminary test.

Communication in sensor networks

Geocast

Calculation of line-line intersections

(1) Line S in parameter form:

$$\vec{x} = \vec{S}_1 + x\vec{s}$$

(2) Line P in normal form:

$$[\vec{x} - \vec{P}_1]\vec{n} = 0$$

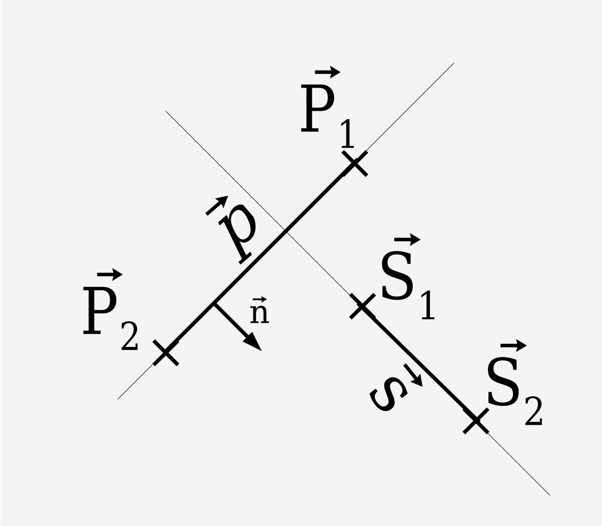
in 2D the normal on p is easily determined as:

$$\vec{n} = (-p.y, p.x)^T$$

(1) in (2)

$$[\vec{S}_1 + x\vec{s} - \vec{P}_1]\vec{n} = 0$$

$$\vec{S}_1\vec{n} + x\vec{s}\vec{n} - \vec{P}_1\vec{n} = 0$$



$$x = \frac{p.x(P_1.y - S_1.y) - p.y(P_1.x - S_1.x)}{-s.x\,p.y + s.y\,p.x}$$

Is x in [0;1]? If so, the same intersection with the straight line P in parameter form has to be executed.

Communication in sensor networks

Rumor Routing

"Rumor Routing Algorithm for Sensor Networks" by Braginsky and Estrin

How to make information available in a sensor network?

a) Flood every event in case of few events and many interested nodes

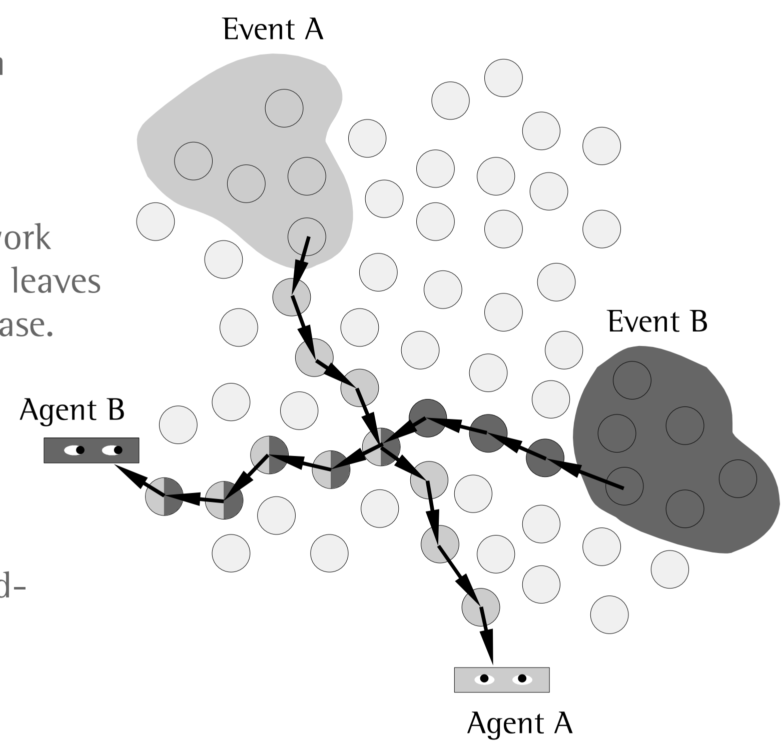
b) Flood the request in case of many events but few requests

c) Use rumor routing for all scenarios in between

Principle of Rumor Routing:

An event sends out agents which travel the network from node to node on a random path. Each visit leaves information about the event in the node's database. After a predefined TTL the agent stops.

A requester also sends out an agent. After some time it will hopefully come across the path of the informing agent by checking the node's databases. It can then travel the backward-references the first agents left in the nodes to reach the event.



Communication in sensor networks

Rumor Routing

Crossing agents can also adopt information from nodes and spread their own knowledge as well as the novel one. Nodes in the proximity of an agents could adopt the information as well, which stabilizes the path.

Critical review

+ Only a small number of nodes have to adopt the same information
+ Only a small number of nodes have to process the request
- When or whether requested information can be delivered is a random process.
- The failure of nodes can interrupt the path to the event (depending on how broad it is).
- The actual behavior of a node is very different from what is shown on the right (see simulation).