Routing Protocols

Nandini Mukherjee

Dept. of Computer Science and Engineering, Jadavpur University

Goal

To disseminate data from sensor nodes to the sink node in energy-awareness manner, hence, maximize the lifetime of the sensor networks

Issues

- Large number of sensors
- Nodes have only modest processing power and memory
- Power is an expensive resource
- Unreliable network links
- Classical protocols require large routing tables to be maintained at each router
- Topology change is a relatively frequent occurrence in the sensor network – so DV-type protocols are not suitable

Sensor Vs Ad - Hoc Networks

- Addressing schemes for sensor nodes differ from those for ad-hoc networks
- Sensor networks are used for data gathering
- Ad - Hoc networks are used for distributed computing
- Data flows from multiple sources to a single destination
- Redundancy in data traffic
- Sensor nodes are prone to failure
- Localized strategies are more attractive in WSN

Routing protocols

- WSN has a data-centric view of the network
- Packets to be routed using destination attributes
- Necessary matchmaking to be performed to bring together those nodes having information with those desiring to use it
- Both pull and push models apply
  - The network must be a database that can be queried about the world state (pull)
  - An entity can actively initiate an action when something of interest is sensed (push)

Data-centric Protocols

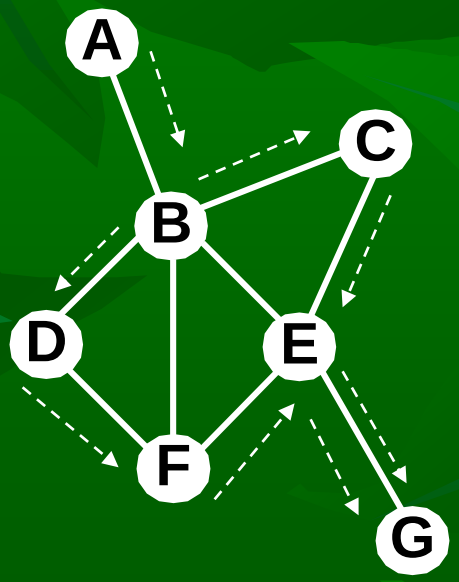
- Major characteristics
  - The ability to query a set of sensor nodes
  - Attribute-based naming
  - Data aggregation during relaying
- For example:
  - Flooding & Gossiping
  - SPIN
  - Directed Diffusion
  - Rumor Routing

Flooding & Gossiping

- In flooding, sensor broadcasts packets to all its neighbours till destination is reached or packets' ttl == 0
- In gossiping, sensor sends packets to a randomly selected neighbour which does the same
- Pros
  - Simple
  - No routing, no state maintenance
- Cons
  - Implosion
  - Overlap
  - Resource blindness
  - Delay in Gossiping

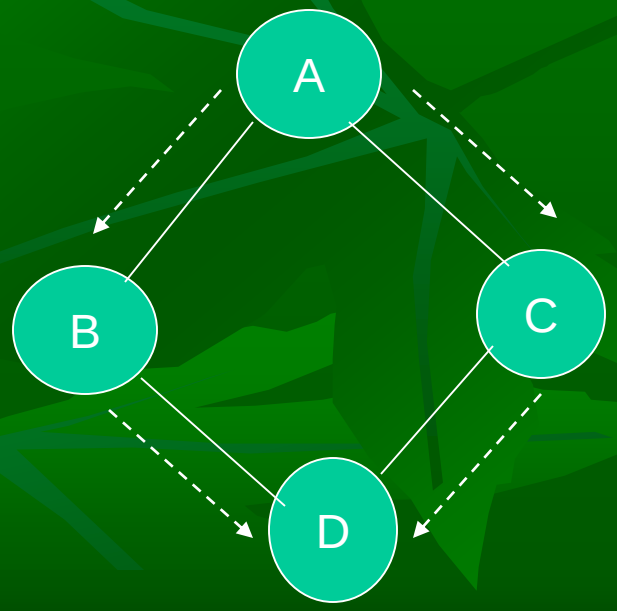
Classical Flooding

- Send data to all neighbors



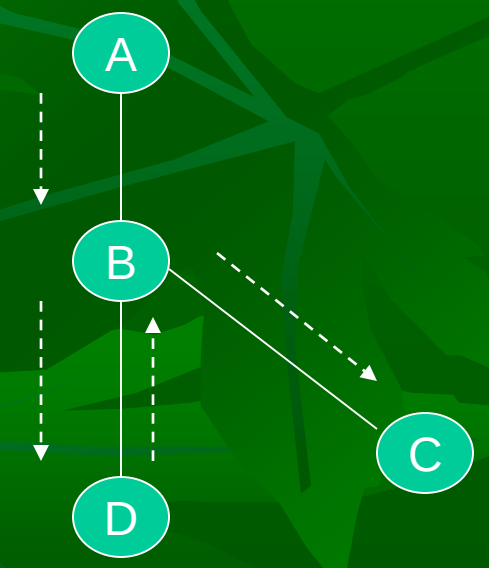
Classical Flooding Problems

- Implosion Problem: A starts by flooding its data to all of its neighbors. Two copies of the data eventually end at node D. The system wastes energy and bandwidth.
- Overlap Problem: Two sensors cover an overlapping graphic region. When the sensors flood their data to a node, the node receives two copies of the data.
- Resource Blindness: Resources do not modify their activities based on the amount of energy they have.



Gossiping

- Gossiping is an alternative to the classical flooding approach that uses randomization to conserve energy.
- At every step each node only forwards data to one neighbor, which it selects randomly.
- Problem: After node D receives the data, it must forward the data back to sender (B), otherwise the data would never reach node C.



Routing in Sensor Networks

- No router
- Hop-by-hop communication
- No globally unique identifier is needed – only the neighbours must be distinguishable
- Coordinated sensing, in-network processing

SPIN: Sensor Protocols for Information Negotiation

- One of the most dominant form of routing in the wireless sensor networks.
- Name data, using meta-data
  - Meta Data for each sensor data
  - Same senor data -> same meta-data
  - Different sensor data -> different meta-data
- Size of meta-data << Size of actual data
- Uses three types of messages:
  - ADV – advertise data
  - REQ – request for data
  - DATA – data message, contains actual sensor data

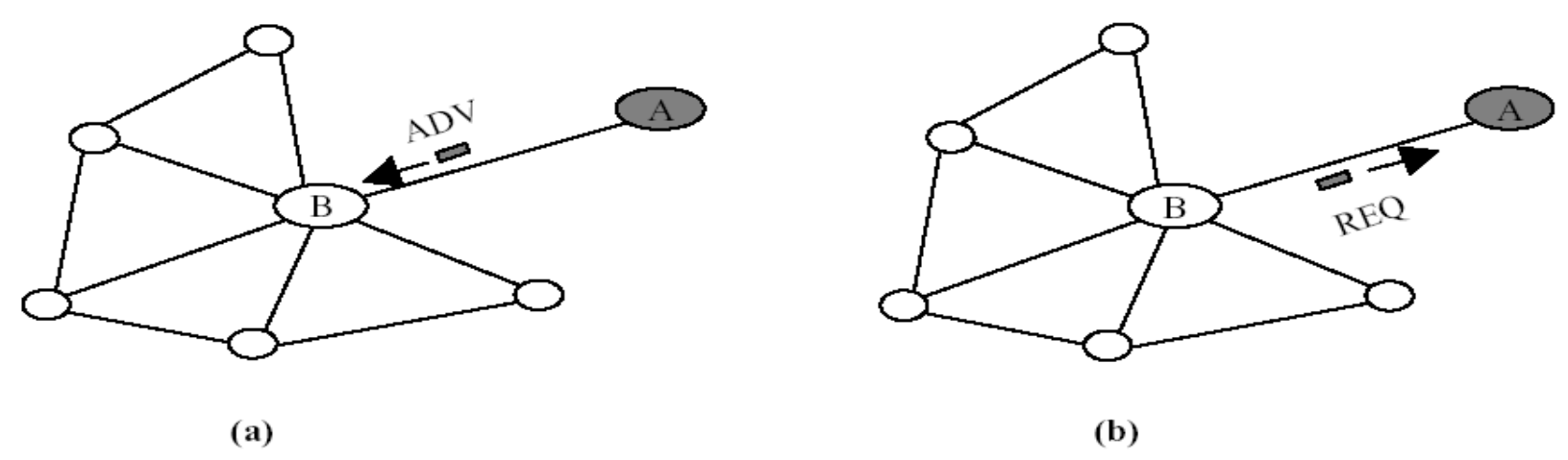
SPIN1

Three way handshaking protocol. ADV, REQ, DATA.

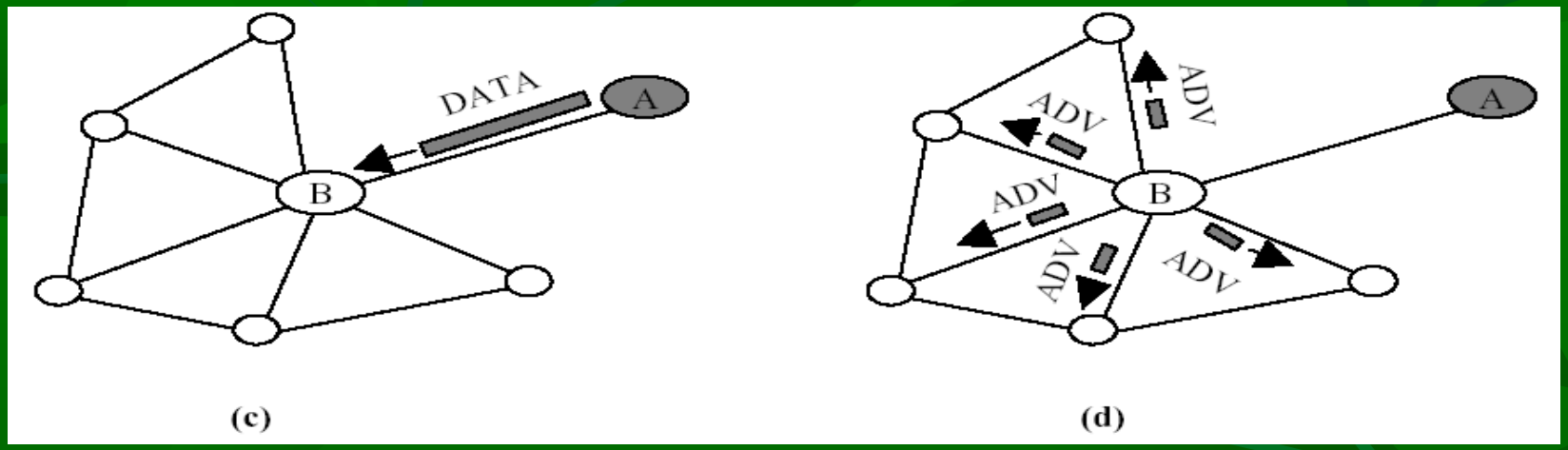
- Event-driven
- Each sensor node has resource manager
  - Keeps track of resource consumption
  - Applications probe the manager before any activity
  - Cut down activity to save energy

SPIN1

- Node A advertises saying that it possesses some data
- Node B sends a REQ listing all of the data it would like to acquire.

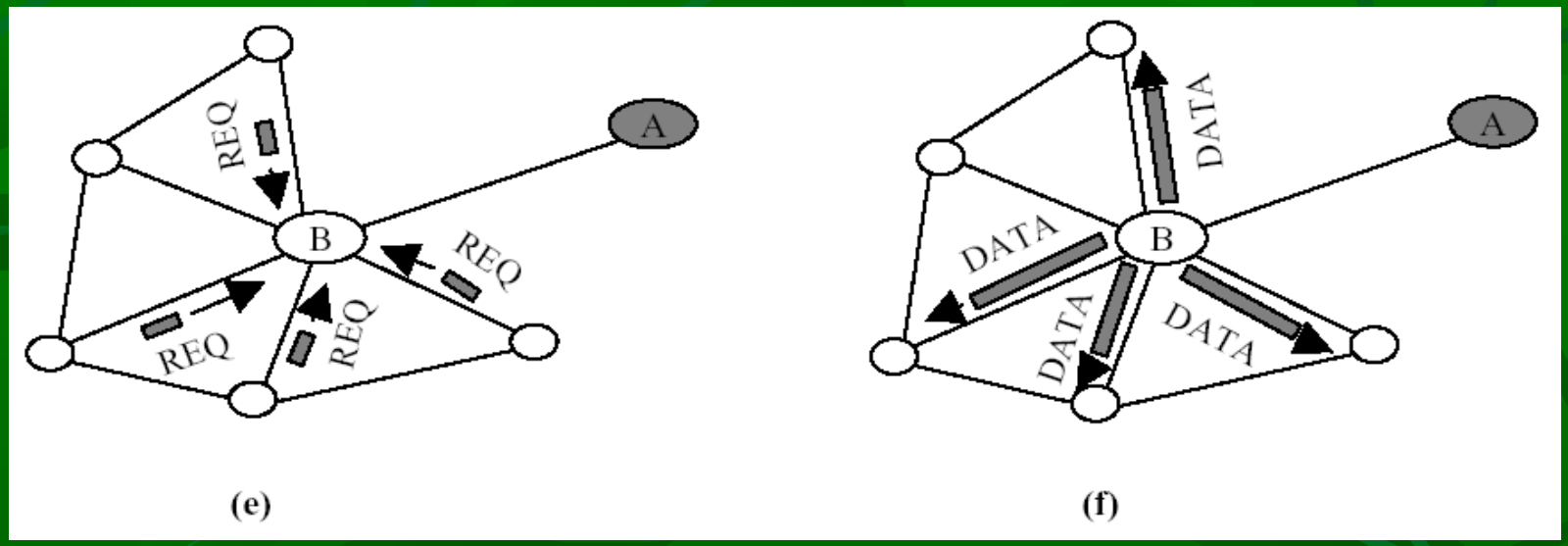


SPIN1



- If node B had its own data, it could aggregate this with the data of node A and advertise
- Nodes need not respond to every message

SPIN1



- Pros
  - Solve the classic problems
  - Topological changes are localized
- Cons
  - No guarantee on the delivery of data

Sensor Protocols for Information Negotiation

- Treats all sensors as potential sink node
- Uses two basic ideas
  - Negotiation
    - Ensures that only useful information will be transferred
    - Nodes describe or name data they observe using metadata
  - Resource adaptation
    - Each sensor node has a resource manager that keeps track of resource consumption
    - The resource manager is probed before transmitting or processing any data
    - Certain activities are postponed or stopped if energy is low

Sensor Protocols for Information Negotiation

- Overcomes the implosion problem
  - Negotiation preceding actual data transmission eliminates transmission of redundant data messages
- Overcomes the overlap problem
  - Using the metadata descriptors nodes can name the portion of the data in which they are interested
- Adapts to the current resource availability scenario

Sensor Protocols for Information Negotiation

- SPIN is an application-level approach
- A middleware library with well-defined API can implement the basic message types, message handling routines and resource management functions

Sensor Protocols for Information Negotiation

- Metadata
  - SPIN does not specify a format for metadata
  - The format can be application-specific
  - However, a metadata x must be able to distinctly identify a data X and distinguish it from other data
  - Size of x must be shorter than X
  - E.g. (x,y,φ) may be a metadata for a camera sensor at geographic coordinates (x,y) and having an orientation φ
  - Applications are responsible for interpreting or synthesize its metadata
  - Some overheads are associated with the storage, retrieval and general management of metadata
  - But as the size of this data is smaller than the actual data, there is always some saving in the communication costs

SPIN Messages

- Uses three types of messages to communicate
  - ADV – new data advertisement
    - When a node has some data to share, it can advertise the fact by transmitting ADV message containing meta-data
  - REQ – request for data
    - A node sends an REQ message when it wishes to receive some advertised data
  - DATA – data message
    - Contains actual sensor data with a meta-data header

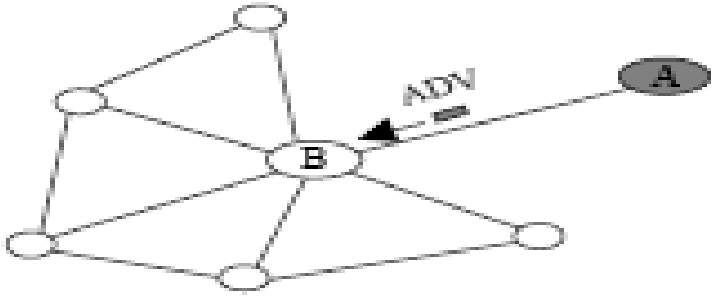
ADV and REQ messages contain only meta-data and therefore they are smaller and cheaper for send and receive

SPIN Protocols

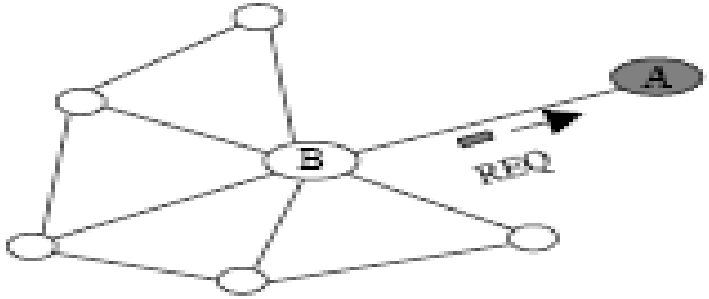
- SPIN-PP
  - Designed to perform optimally point-to-point communication
  - No consideration for energy
- SPIN-EC
  - Same as SPIN-PP with a energy-consideration heuristic added to it
- SPIN-BC
  - designed for broadcast networks in which the nodes use a single shared channel to communicate.
- SPIN-RL
  - A reliable version of SPIN-BC for lossy network

SPIN-PP

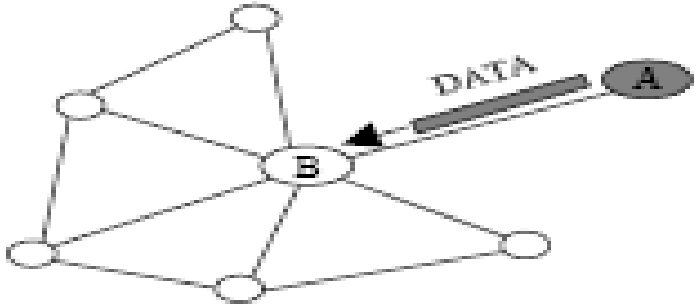
- Two nodes can have exclusive communication with each other without any interference from the other nodes
- The cost of communication with n nodes in terms of time and energy is n times more expensive than communicating with one node
- A simple 3-way handshake protocol in which energy is not considered to be a constraint
  - When a node has some new data, it advertises this new data using the ADV messages to its neighbors
  - When a neighboring node receives this advertisement, it checks the meta-data to see whether it already has the data item or not
  - In case it does not, it sends an REQ message back requesting for the data item
  - Upon receiving the REQ message, the originating node sends DATA messages containing the data to the requesting node.


ADV
A
B

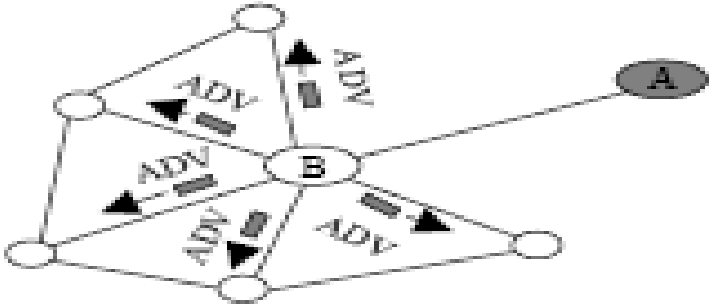
(1)


A
B
REQ

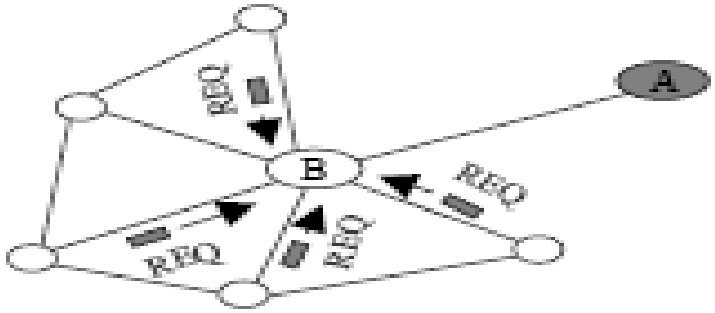
(2)


DATA
A
B

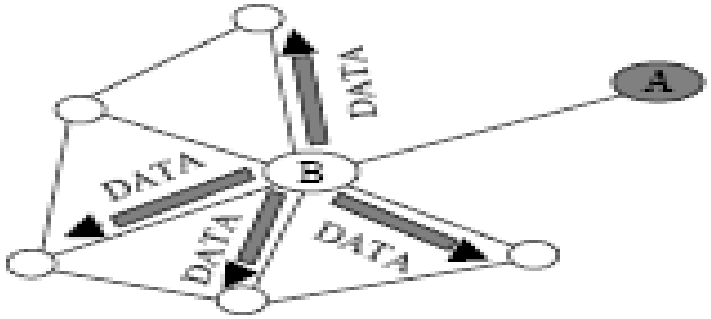
(3)


ADV
ADV
A
ADV
B
ADV
ADV

(4)


REQ
A
B
REQ
REQ
REQ

(5)


DATA
A
DATA
B
DATA
DATA

(6)

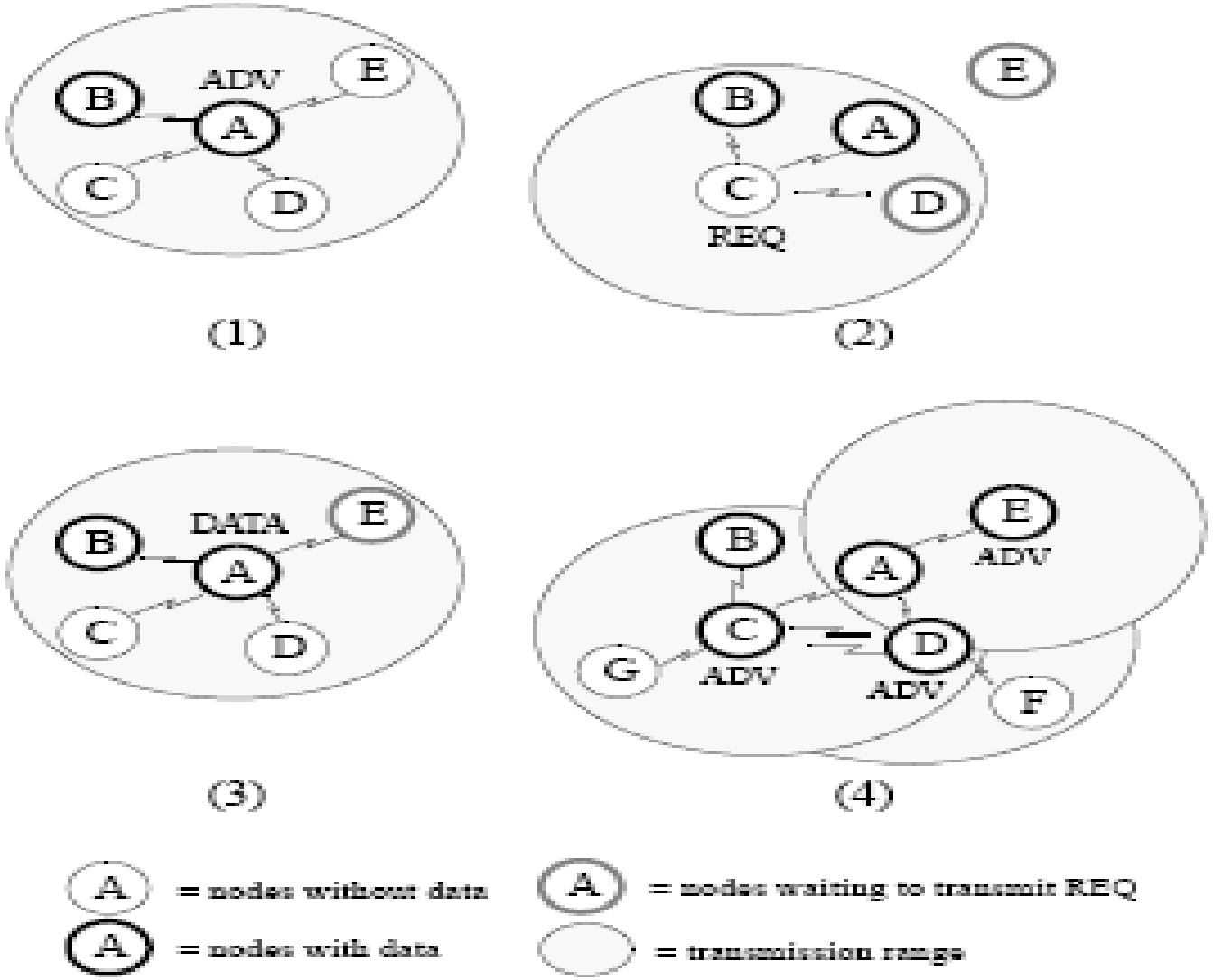
SPIN-PP

- Node B does not have to send ADV message to A
- Node B can aggregate its own data with the data from node A and send advertisement for the aggregated data
- All nodes need not have to respond
- Suitable for lossless network, but can be adapted to a lossy network as well
  - E.g. re-advertising or sending repeated requests or data
  - In case of mobile networks, when the topology changes a node's neighbours list is updated and the node re-advertises for its data

- Major advantage is its simplicity
- Each node requires to know only about its single-hop neighbors and does not require any other topology information

SPIN-BC

- When a node sends out a message, it is received by all the other nodes within a certain range of the sender
  - The node senses the channel to check whether it is free before transmitting messages
- A node which has received an ADV message does not immediately respond with an REQ message
  - It first checks whether it has already received or requested for the advertised data
  - If not, a randomly chosen timer (from a predetermined interval) is set
  - When the timer expires, the node sends (broadcasts) an REQ message with the original advertising node mentioned in the message header
  - When a node other than the advertising node receives the REQ message (before expiration of the timer, it cancels its own request
  - Thus there cannot be any redundant requests for the same message
- When the advertising node receives an REQ message, it sends the data message only once, even though it might have got multiple requests for the same message


B
ADV
E
A
C
D
(1)
E
B
A
C
D
REQ
(2)
B
DATA
E
A
C
D
(3)
E
B
ADV
A
C
D
G
ADV
ADV
F
(4)
A = nodes without data
A = nodes with data
A = nodes waiting to transmit REQ
= transmission range

SPIN-BC

- One-to-many communication is *1/n* times cheaper than point-to-point communication (*n* being the number of nodes in the neighbourhood)
  - Broadcast and unicast transmissions use same amount of network resources in a broadcast network
- Nodes can coordinate their resource conserving effort more effectively as each node overhears any transmission within the range
  - More than one requests can be responded with a single broadcast
  - Redundant requests are suppressed

SPIN-EC

- A node will participate actively in the protocol only if it is above a certain energy threshold and believes it can complete all the other stages of the protocol
- If a node receives an advertisement , it will not send out an REQ message if it does not have enough energy to transmit an REQ message and receive the corresponding DATA message
- A node can, however, receive ADV and REQ messages below this threshold
- Only handling DATA messages is prevented

SPIN-RL

- This protocol makes two changes to the SPIN-BC protocol
  - Each node keeps track of all the advertisements it hears and the corresponding nodes
  - If it does not receive a piece of data it has requested within a certain period of time, it sends out the request again
  - Secondly, the nodes put a limit on the frequency with which they resend the data messages
  - After sending out a data message, a node will wait for a certain period of time before it responds to other requests for the same data message

Directed Diffusion

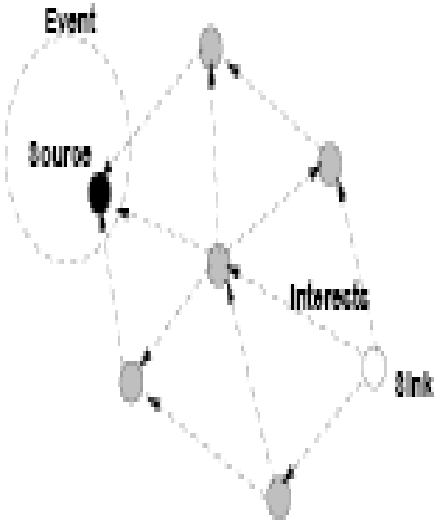
- A data-centric communication paradigm
- A destination-initiated reactive routing technique in which routes are established when required
- The querying node (sink) broadcasts its interest message periodically to all of its neighbors
  - For example – “I would like to track x animal”
- Interest is expressed as a list of attribute-value pairs that describe a task using some task-specific naming scheme
- The data which is sent as a response to such an interest is also named in a similar manner
- Every node is task-aware, i.e. each of them store and interpret interests.
- All nodes have an interest cache in which each item corresponds to a different interest (information about the sink is not required)

Directed Diffusion

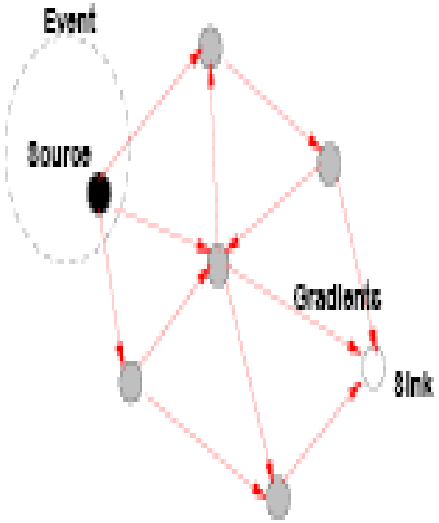
- The node remembers which neighbour (s) has sent the interest and sets up a gradient towards this neighbour
- A gradient represents
  - the direction towards which data matching an interest flows, and
  - the status of that demand
- The sensor node then redistributes the interest to its neighbors
  - It will simply broadcast the interest to all of its neighbors
  - When the node can infer where potential sources might be (for example, from geographic information or existing similar gradients), the interest can be forwarded to a subset of neighbors
- When a sensor node that matches the interest is found (source), local sensors are activated to collect data
- The sensor node then generates data messages matching the interest (using an attribute-based naming scheme)

Directed Diffusion

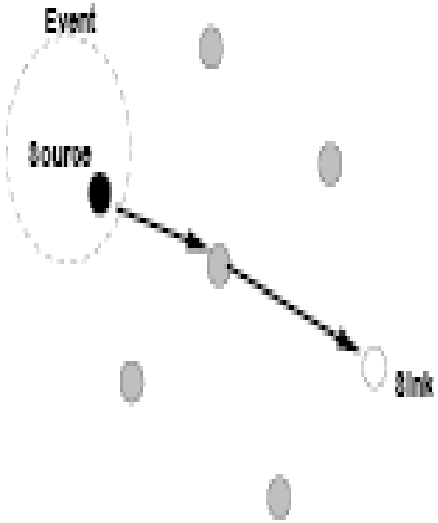
- The initial data message from the source is marked as exploratory and is sent to all neighbors for which it has matching gradients
- If the sink has multiple neighbors, it chooses to receive subsequent data messages for the same interest from a preferred neighbor (for example, the one which delivered the first copy of the data message)
- The sink reinforces the preferred neighbour, which, in turn reinforces its preferred upstream neighbour and so on
- After the initial exploratory data message, subsequent messages are sent only on reinforced paths
- Data is cached at intermediate nodes as it propagates toward sinks
- Cached data is used for several purposes at different levels of diffusion
  - E.g. to suppress duplicate messages and to prevent loops



(a) Interest propagation



(b) Initial gradients set up



(c) Data delivery along reinforced path

Directed Diffusion

- Communication Model
  - Interest (Broadcast)
  - Exploratory message(Broadcast)
  - Reinforcement(Unicast)
  - Data message (Unicast)

Directed Diffusion

- Pros
  - On demand route setup
  - Each node does aggregation and caching, thus good energy efficiency and low delay
- Cons
  - Query-driven, not a good choice for continuous data delivery
  - Extra overhead for data matching and queries

Directed Diffusion

- **Attribute Tuples and Matching Rules**
  - Diffusion messages and application interests are composed of attribute-value-operation tuples
  - Attributes are identified by unique *keys* and have a data format
  - The operation field defines how data messages and interests interact, e.g. EQ, NE, LE, IS etc.
  - A *one-way match* compares all formal (comparison) parameters of one attribute set against the actuals (IS operation) of the others
  - Two sets of attributes have a *complete match* if one-way matches succeed in both directions

Directed Diffusion

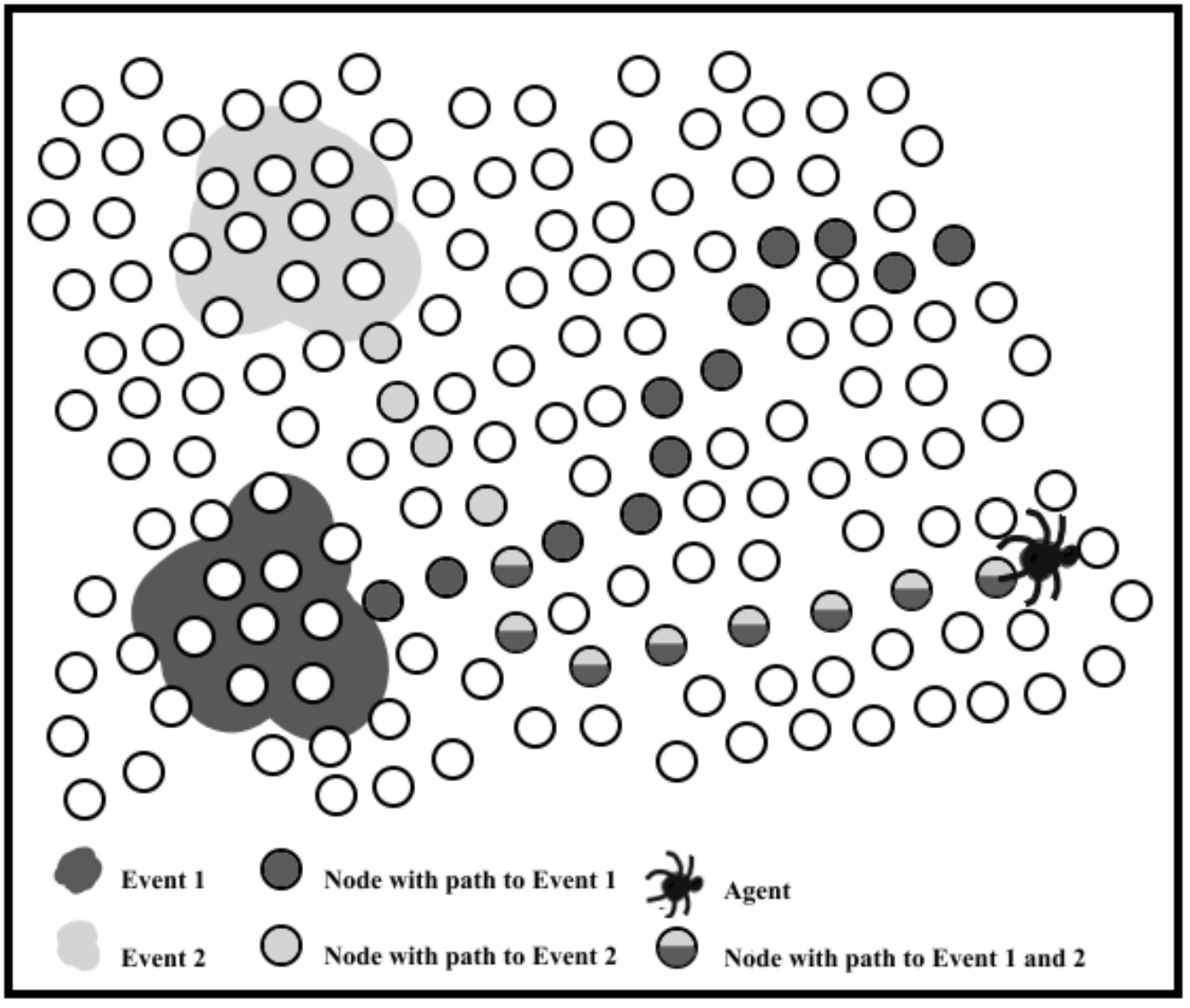
- Matching Rules
  - Actual
    - literal or bounds value
    - IS operation only
  - Formal
    - comparison or unbound value
    - Rest operations including EQ
- Matching Algorithm
  - One-way match (A with B)
    - Compares ONLY all FORMAL parameters of A's attribute set with ACTUAL of the B's set
  - Two-way match (A with B and B with A)
  - E.g. "Confidence GT 0.5" Matches "Confidence IS 0.7"; doesn't Match "Confidence IS 0.3", "Confidence LT 0.7", "Confidence GT 0.7"

Rumor Routing

- In many applications the quality of the path may not be important
- The application requests only a small amount of data back
- In such cases, flooding every query may not be as efficient as delivering it by a non-optimal route
- If there are few events and many queries, notification of events may be flooded and gradients may be set up towards it
- In rumor routing, a compromise is made between flooding query and flooding event notification
- Paths are created leading to each event, thereby creating a network-wide gradient field
- When a query is generated it can be sent on a random walk until it finds the event path

Rumor Routing - overview

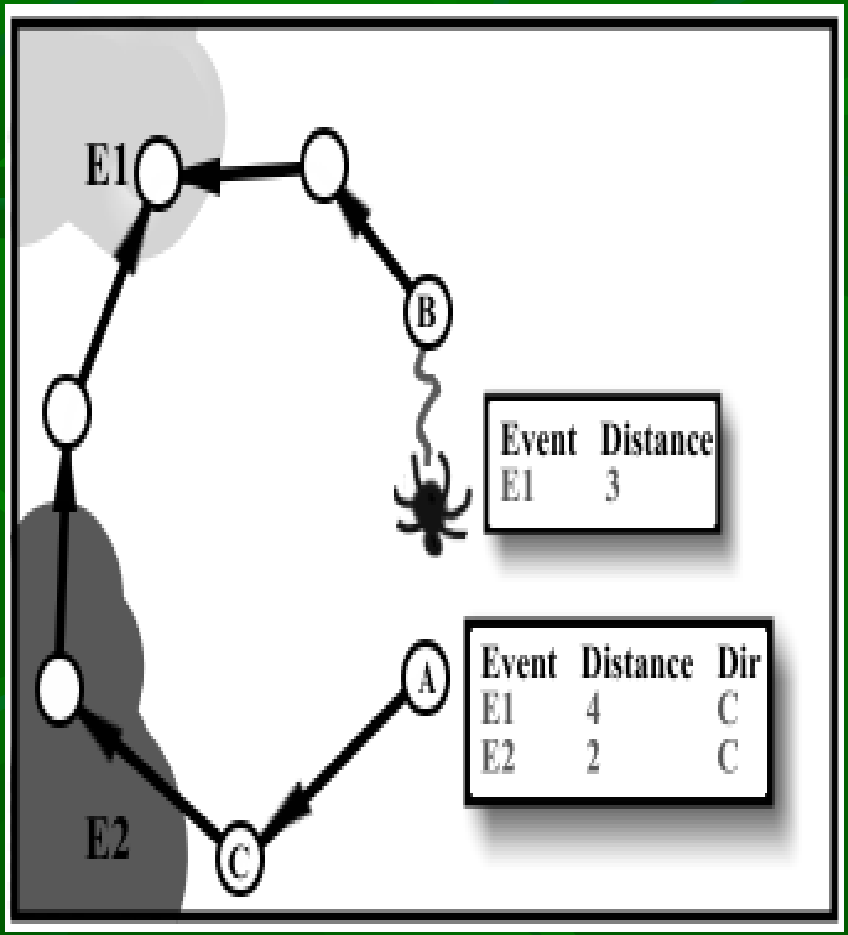
- Looks at routing queries to the nodes which have observed a particular event
- Uses a set of long-lived *agents* which create paths that are directed towards the events they encounter
- Whenever an agent crosses path with a path leading to an event that it has not encountered, it adapts its behaviour thus creating a path state which leads to both (or multiple) events
- When the agents come across shorter paths, they optimize the paths in the network by updating the routing tables to reflect the more efficient path
- The agent has a lifetime of a certain number of hops after which it dies

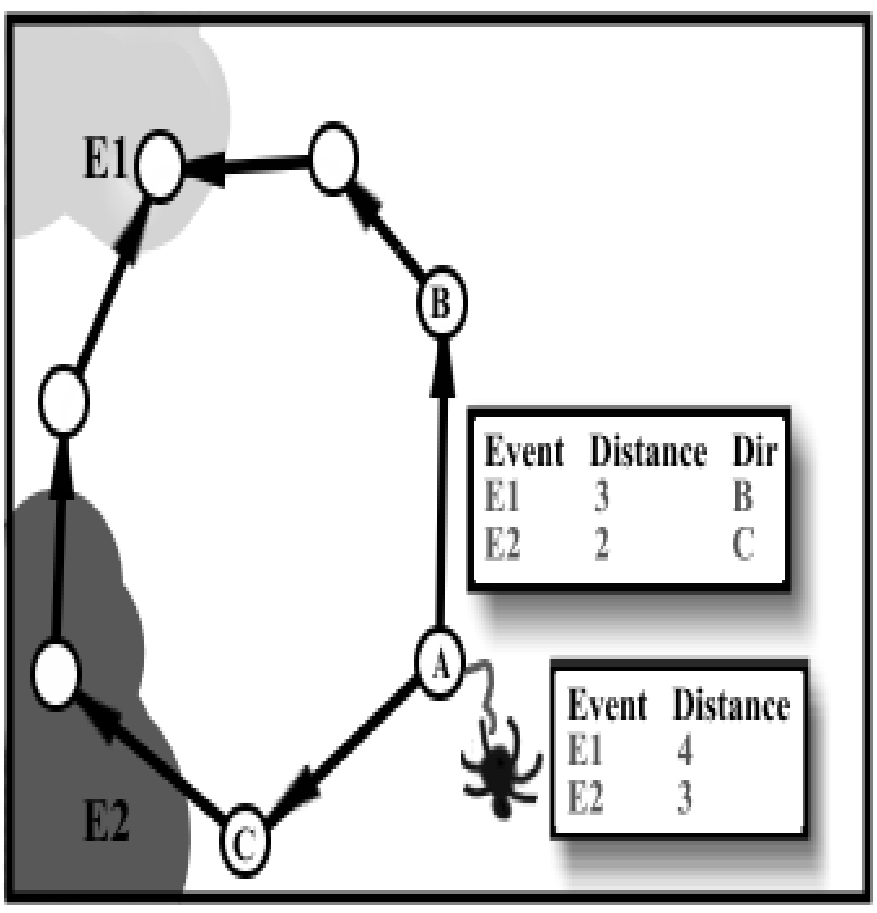

Event 1
Node with path to Event 1
Agent
Event 2
Node with path to Event 2
Node with path to Event 1 and 2

Rumor Routing – nodes and agents

- Each node maintains a list of its neighbours and an events table with forwarding information to all the events it knows
- The neighbor list can be actively created and maintained by actively broadcasting a request, or passively, through listening for other node broadcasts
- When a node witnesses an event, it adds it to its event table, with a distance of zero to the event
- It also probabilistically generates an agent
- An agent is a long-lived packet, which travels the network, propagating information about local events to distant nodes
- It contains an events table, similar to the nodes, which it synchronizes with every node it visits

Rumor Routing – nodes and agents



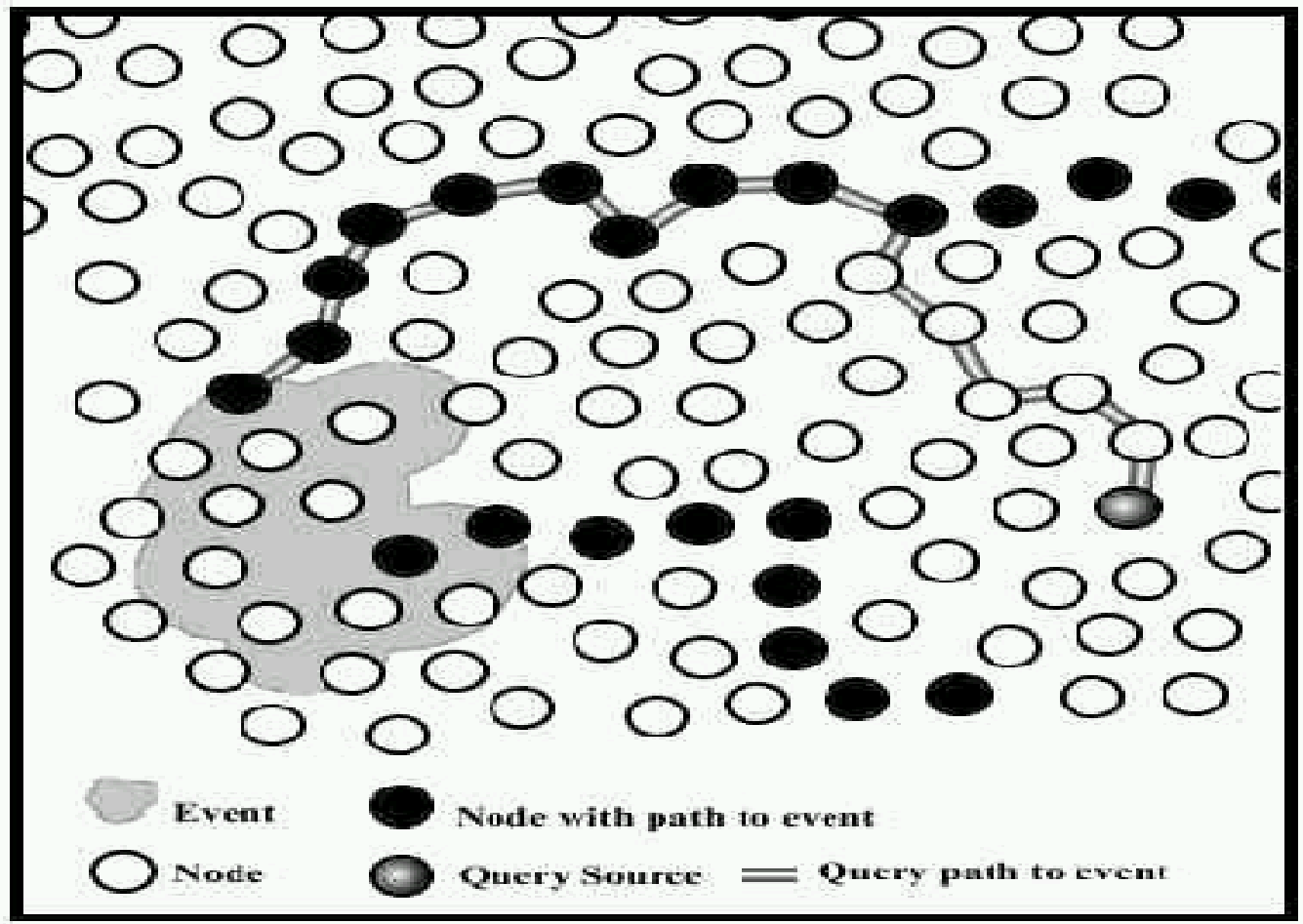


Rumor Routing – nodes and agents

- In order to determine the agent's next hop
  - the agent maintains a list of recently seen nodes
  - when it arrives at a node, it adds all of the node's neighbors to the list
  - when deciding its next hop, it checks the list to ensure that no visited node is selected (as far as possible)
  - this helps to eliminates most loops
- Policy to generate agents:
  - Although any node can generate an agent, it is logical to allow a node to generate an agent only when it observes an event
  - Braginsky et al suggest to maintain a fixed probability that a node that has witnessed an event would generate an agent (for simulation purpose)
  - The actual number of agents generated depends on the number of events, the event size, and the node density.
  - This may be taken up for future work.

Flooding query

- Any node generating a query will transmit the query if it has a route to the event else it will transmit it in a random direction
- Forwarding query is done until the query has not exceeded its TTL
- The query also keeps a list of recently seen nodes, and avoiding visiting them
- To avoid looping in a dynamic network
  - a random id is assigned to each query
  - a list of recently seen queries is maintained in the node

  .
- The application must have a timeout mechanism to determine that a query finally did not reach the destination
- The query generating node then floods the network
- The lesser the number of queries which flood, the lesser the energy consumed

Rumor Routing