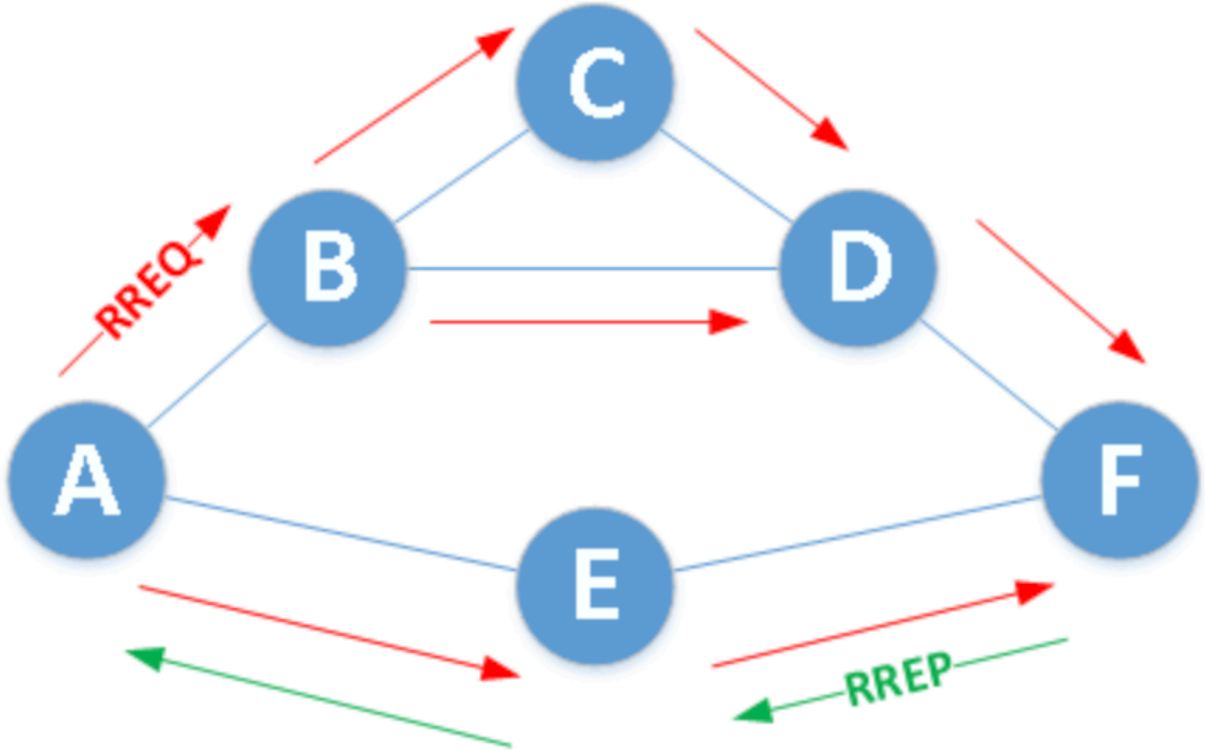
ROUTING PROTOCOLS



FLAT ROUTING

In flat network, each node typically plays the same role and sensor nodes collaborate together to perform the sensing task.

Types:

- Flooding
- Gossiping
- SPIN (Sensor Protocols for Information via Negotiation)
- DD (Directed Diffusion)
- Rumor routing

FLOODING

A classical mechanisms to relay data in sensor networks without the need for any routing algorithms and topology maintenance.

drawbacks:

- Implosion
- Overlap

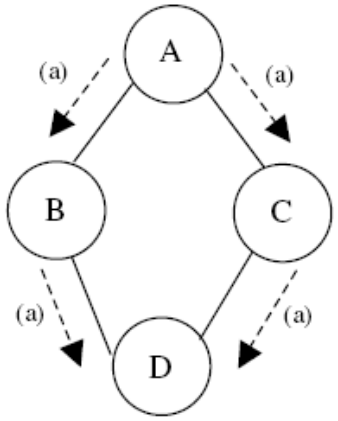


Fig. 1. The implosion problem. Node A starts by flooding its data to all of its neighbors. D gets two same copies of data eventually, which is not necessary.

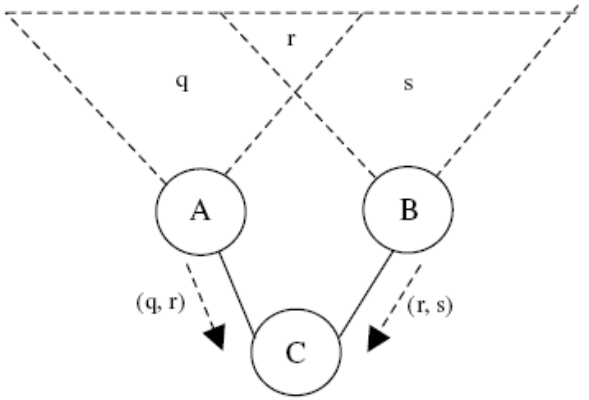
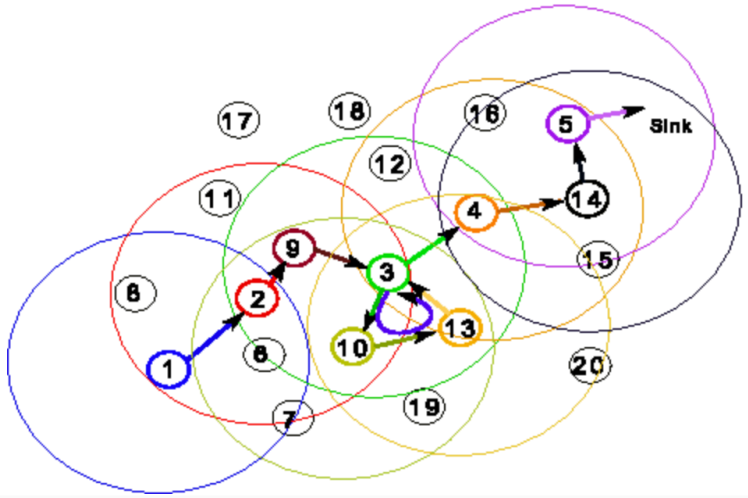


Fig. 2. The overlap problem. Two sensors cover an overlapping geographic region and C gets same copy of data form these sensors.

GOSSIPING

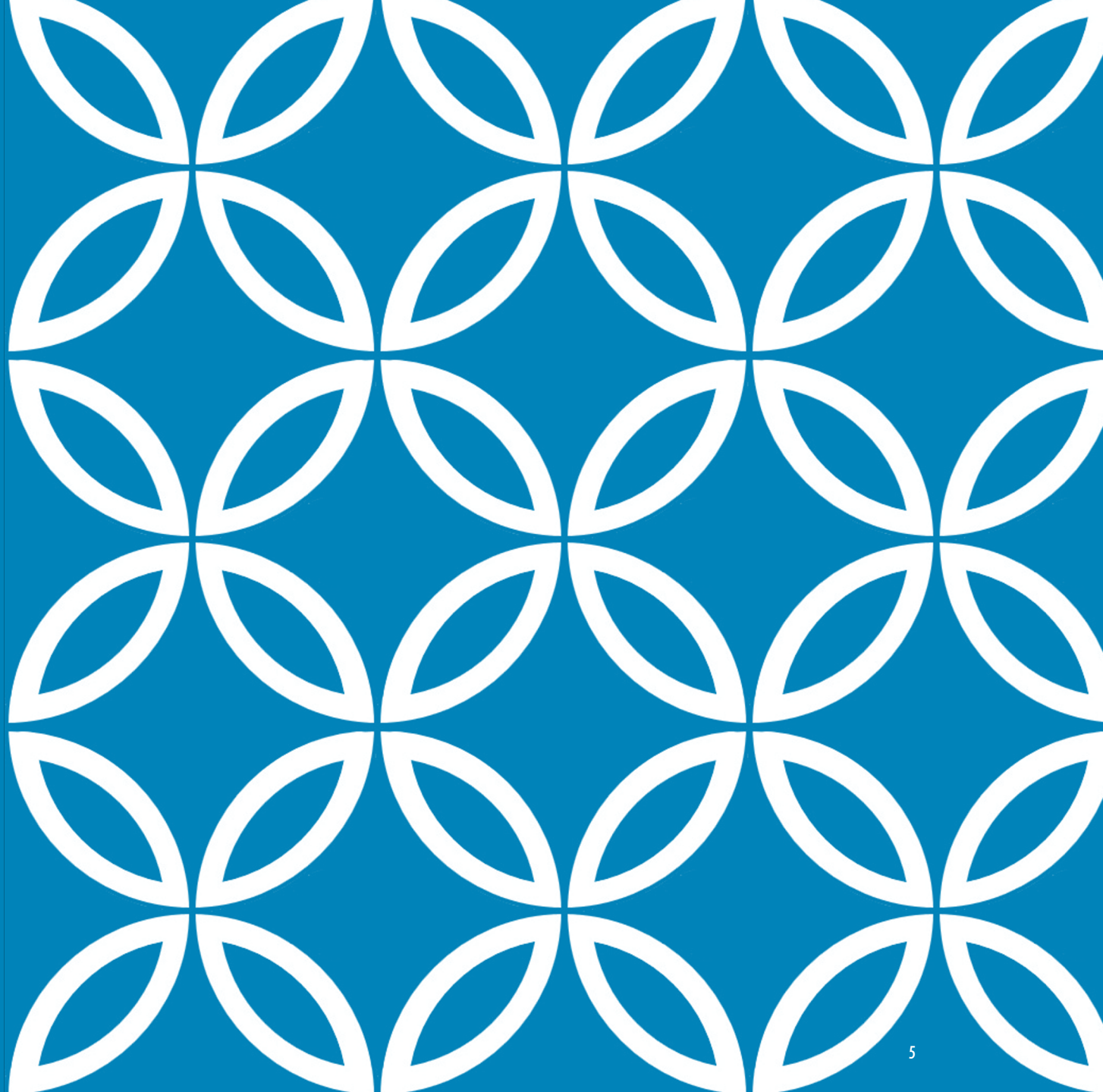


A slightly enhanced version of flooding where the receiving node sends the packet to a randomly selected neighbor which picks another neighbor to forward the packet to and so on.

Advantage: avoid the implosion

Drawback: Transmission delay

SPIN ROUTING



SPIN

A family of adaptive protocols called Sensor Protocols for Information via Negotiation (SPIN)

Nodes assign a high-level name to completely describe their collected data (called meta-data)

Use thee types of messages for routing mechanism

- ADV (advertisement), REQ (request) and DATA

SPIN (CONT.)

Negotiation

- Before transmitting data, nodes negotiate with each other to overcome implosion and overlap
- Only useful information will be transferred
- Observed data must be described by meta-data

Resource adaptation

- Each sensor node has resource manager
- Applications probe manager before transmitting or processing data
- Sensors may reduce certain activities when energy is low

THREE-STAGE HANDSHAKE PROTOCOL

ADV – data advertisement

- Node that has data to share can advertise this by transmitting an ADV with meta-data attached

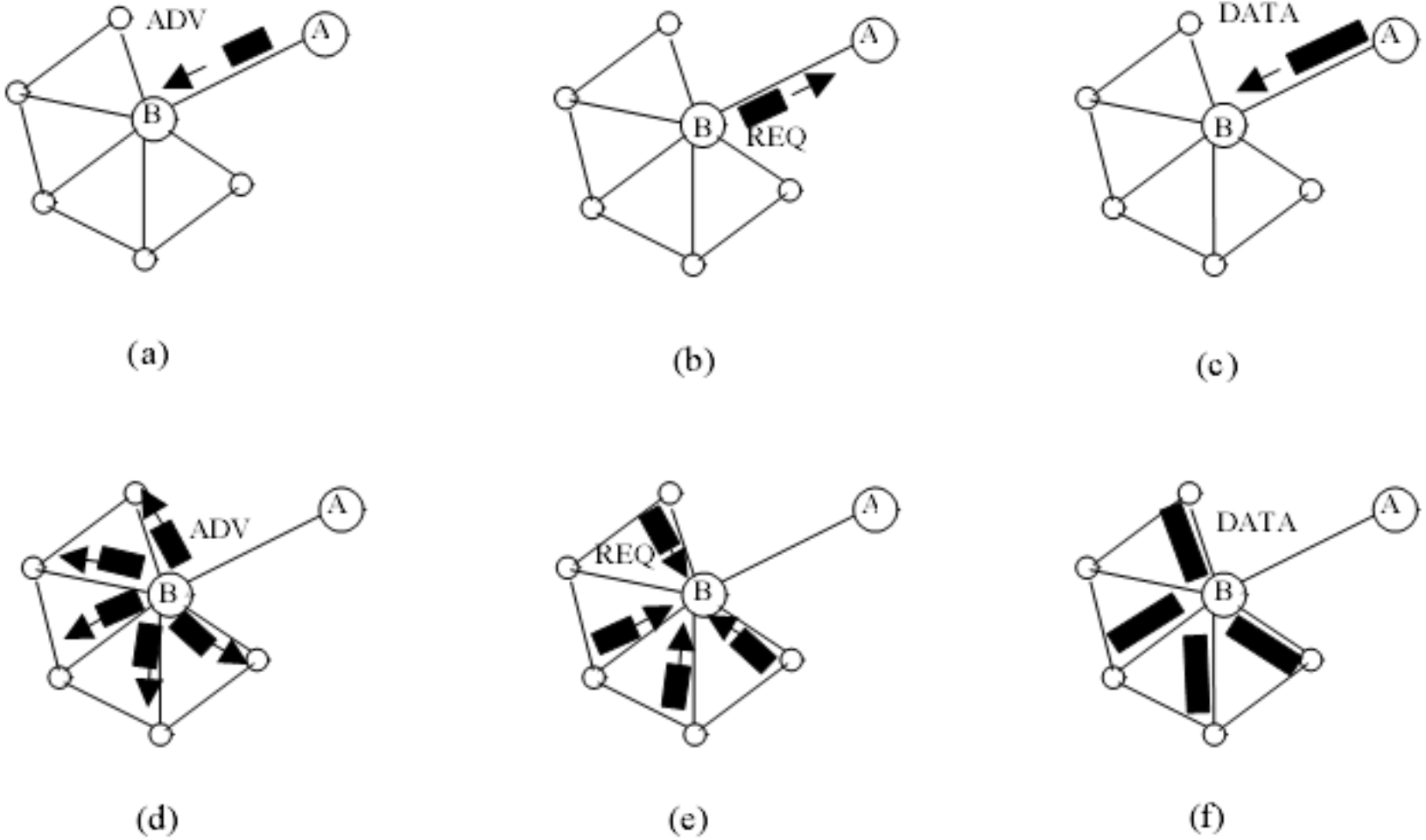
REQ – request for data

- Node sends a request when it wishes to receive some actual data

DATA – data message

- Contain actual sensor data with a meta-data header
- Usually much bigger than ADV or REQ messages

ROUTING MESSAGES EXCHANGE



SPIN META-DATA

Completely describe the data

- Must be smaller than the actual data for SPIN to be beneficial
- If you need to distinguish pieces of data, their meta-data should differ

Meta-Data is application specific

- Sensors may use their geographic location or unique node ID
- Camera sensor may use coordinate and orientation

SPIN FAMILY

Protocols of the SPIN family

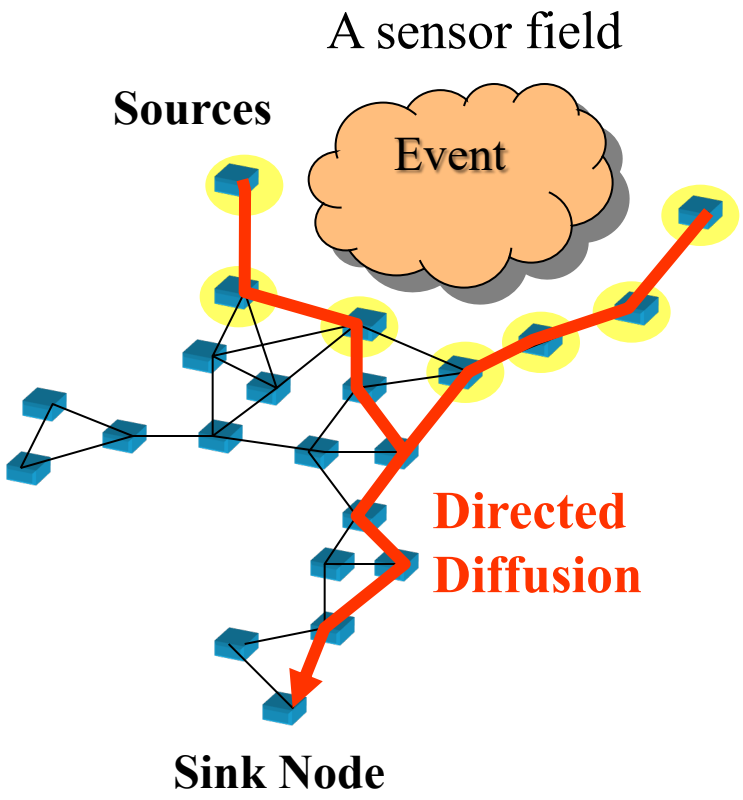
- SPIN-PP
  - It is designed for a point to point communication, i.e., hop-by-hop routing
- SPIN-EC (energy conserving)
  - It works similar to SPIN-PP, but, with an energy heuristic added to it
- SPIN-BC
  - It is designed for broadcast channels
- SPIN-RL
  - When a channel is lossy, a protocol called SPIN-RL is used where adjustments are added to the SPIN-PP protocol to account for the lossy channel.

DIRECTED DIFUSION

DIRECTED DIFFUSION DD

Data-centric communication

- Data is named by attribute-value pairs
- Different form IP-style communication
  - End-to-end delivery service
- e.g.
  - How many pedestrians do you observe in the geographical region X?



DD ROUTING

Propagate interest

Set up gradients

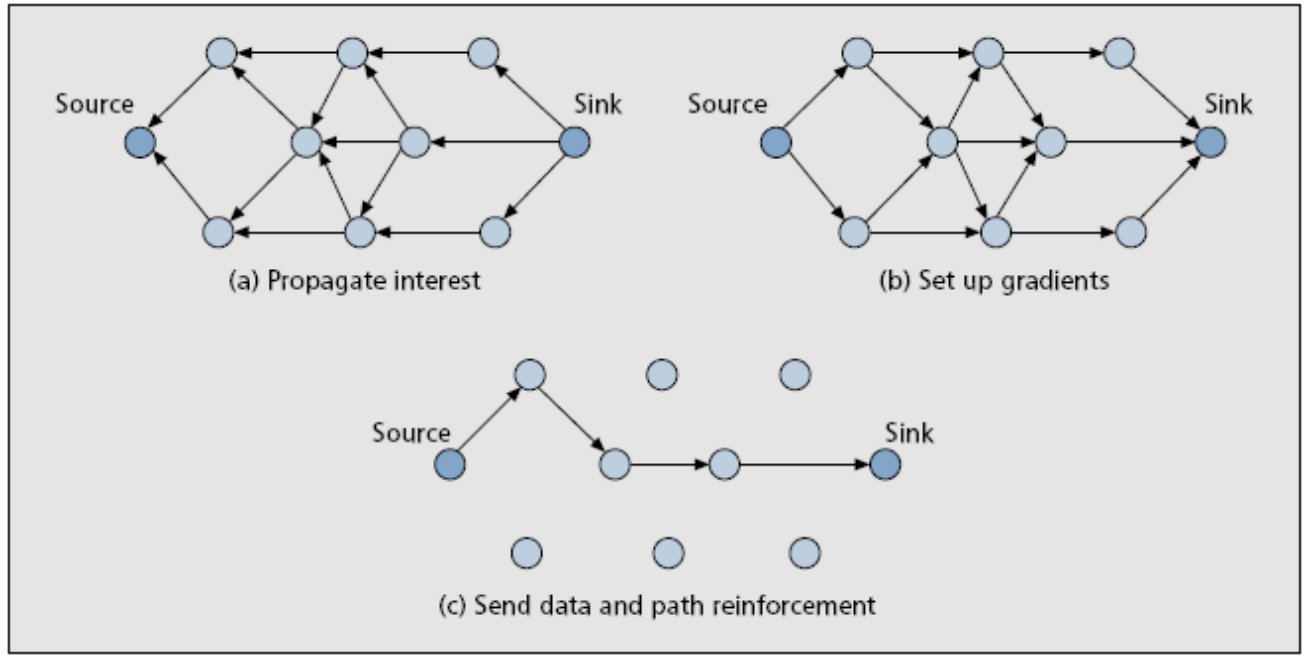
Send data and path reinforcement

Data-centric communication

- Human operator's query (task) is *diffused*
- Sensors begin collecting information about query
- Information returns along the reverse path
- Intermediate nodes *aggregate* the data

Directed diffusion – use publish/subscribe

- Inquirer expresses an interest, *I*, using attribute values
- Sensor sources that can service *I*, reply with data


Source
Sink
(a) Propagate interest
Source
Sink
(b) Set up gradients
Source
Sink
(c) Send data and path reinforcement

SPIN VS. DD

Directed diffusion differs from SPIN in two aspects.

- Query method
- Communication method

directed diffusion may not be applied to applications (e.g., environmental monitoring)

Matching data to queries might require some extra overhead

RUMOR ROUTING

RUMOR ROUTING

Designed for query/event ratios between query and event flooding

Motivation

- Sometimes a non-optimal route is satisfactory

Advantages

- Tunable best effort delivery
- Tunable for a range of query/event ratios

Disadvantages

- Optimal parameters depend heavily on topology (but can be adaptively tuned)
- Does not guarantee delivery

RUMOR ROUTING

Variation of directed diffusion

Don't flood interests (or queries)

Flood events when the number of events is small but the number of queries large

Route the query to the nodes that have observed a particular event

Long-lived packets, called agents, flood events through the network

When a node detects an event, it adds the event to its events table, and generates an agent

Agents travel the network to propagate info about local events

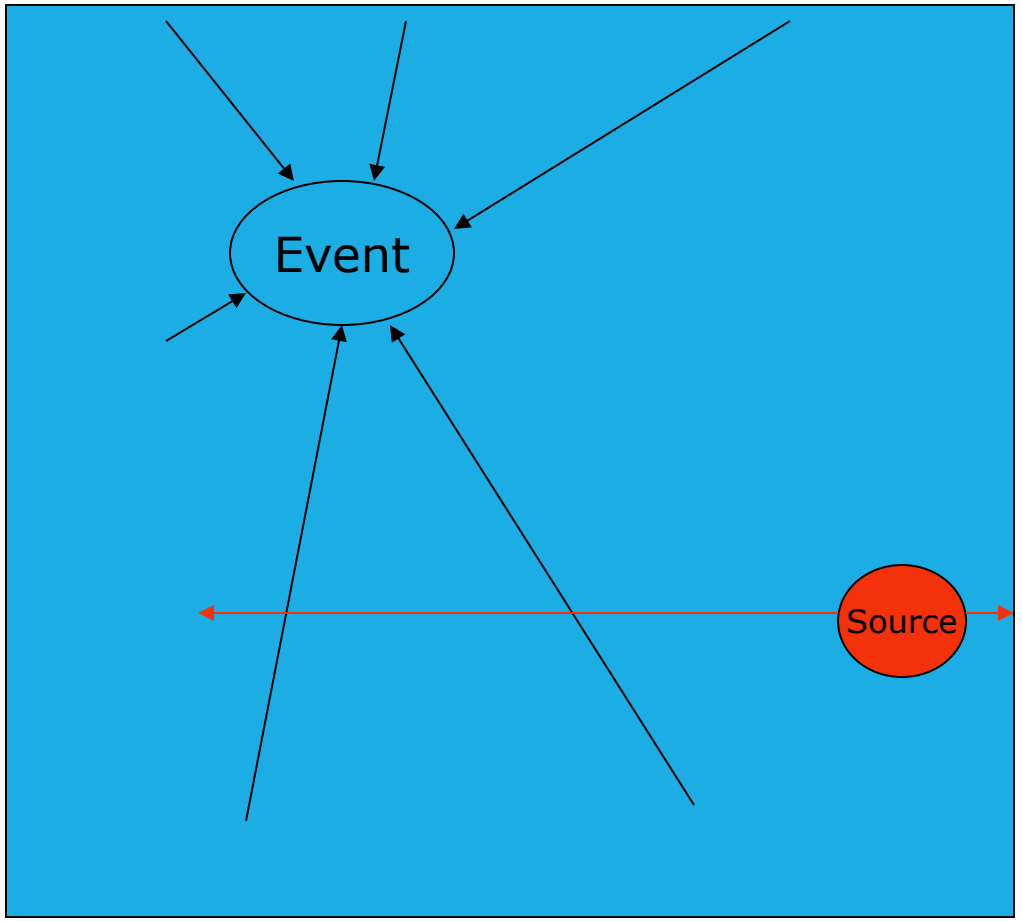
• An agent is associated withTTL (Time-To-Live)

BASIS FOR ALGORITHM

Observation: Two lines in a bounded rectangle have a 69% chance of intersecting

Create a set of straight line gradients from event, then send query along a random straight line from source

Thought: Can this bound be proved for a random walk . What is this bound if the line is not really straight?



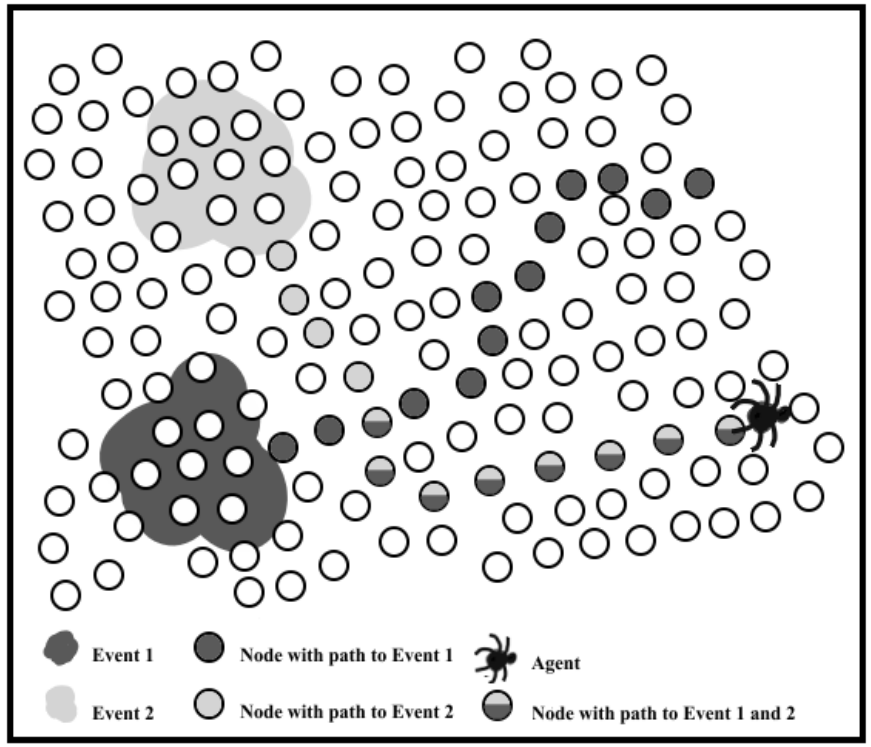
CREATING PATHS

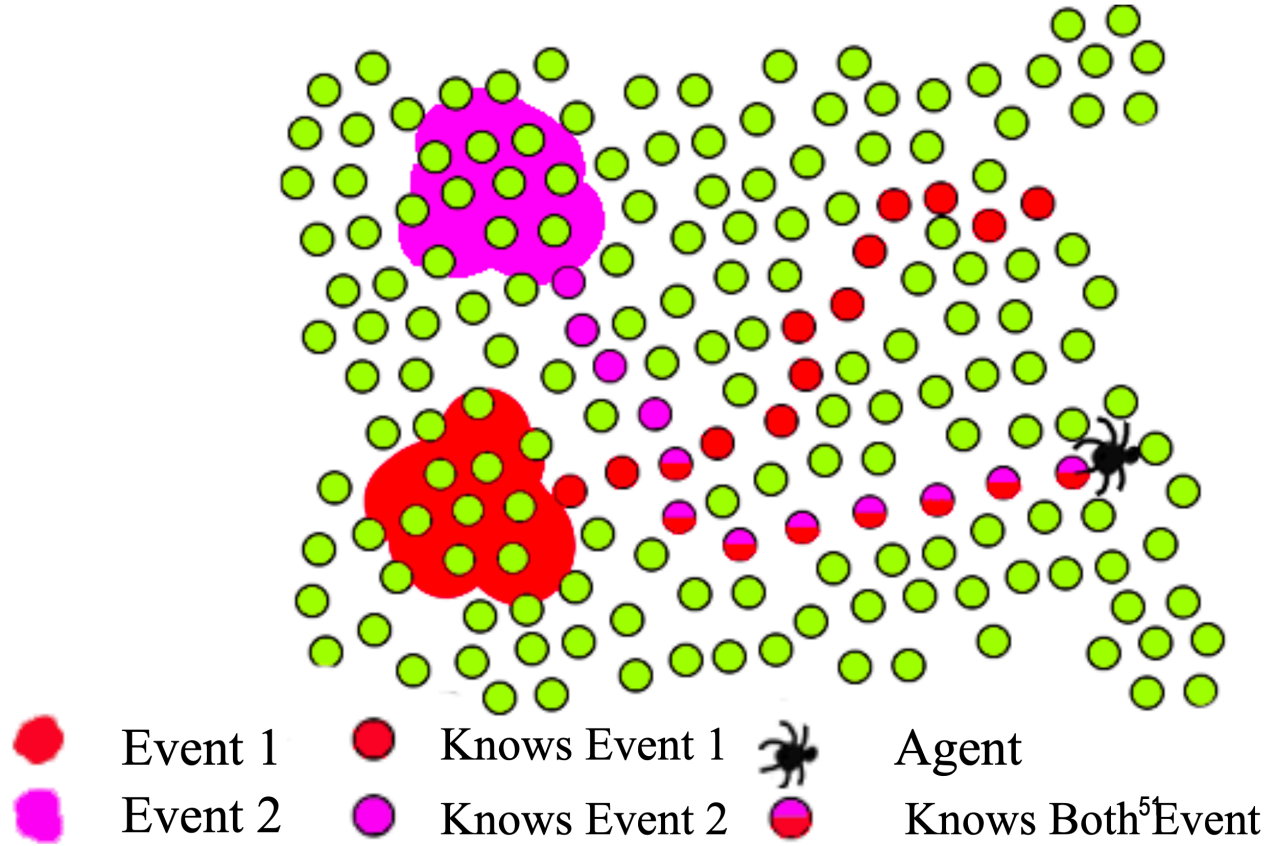
Nodes that observe an event send out agents which leave routing info to the event as state in nodes

Agents attempt to travel in a straight line

If an agent crosses a path to another event, it begins to build the path to both

Agent also optimizes paths if they find shorter ones




Event 1
Event 2
Knows Event 1
Knows Event 2
Agent
Knows Both Event

OBSERVATIONS

- When a node generates a query, a node knowing the route to a corresponding event can respond by looking up its events table
- No need for query flooding
- Only one path between the source and sink
- Rumor routing works well only when the number of events is small
- Cost of maintaining a large number of agents and large event tables will be prohibitive
- Heuristic for defining the route of an event agent highly affects the performance of next-hop selection

Rumuor routing is another variation of Directed Diffusion and is mainly intended for contexts in which geographic routing criteria are not applicable.

Rumor Routing is a logical compromise between flooding queries and flooding event notifications.

The idea is to create paths leading to each event; whereas event flooding creates a network-wide gradient field.

In this way, when a query is generated it can be sent on a random walk until it finds the event path; instead of flooding it throughout the network.

OBSERVATION

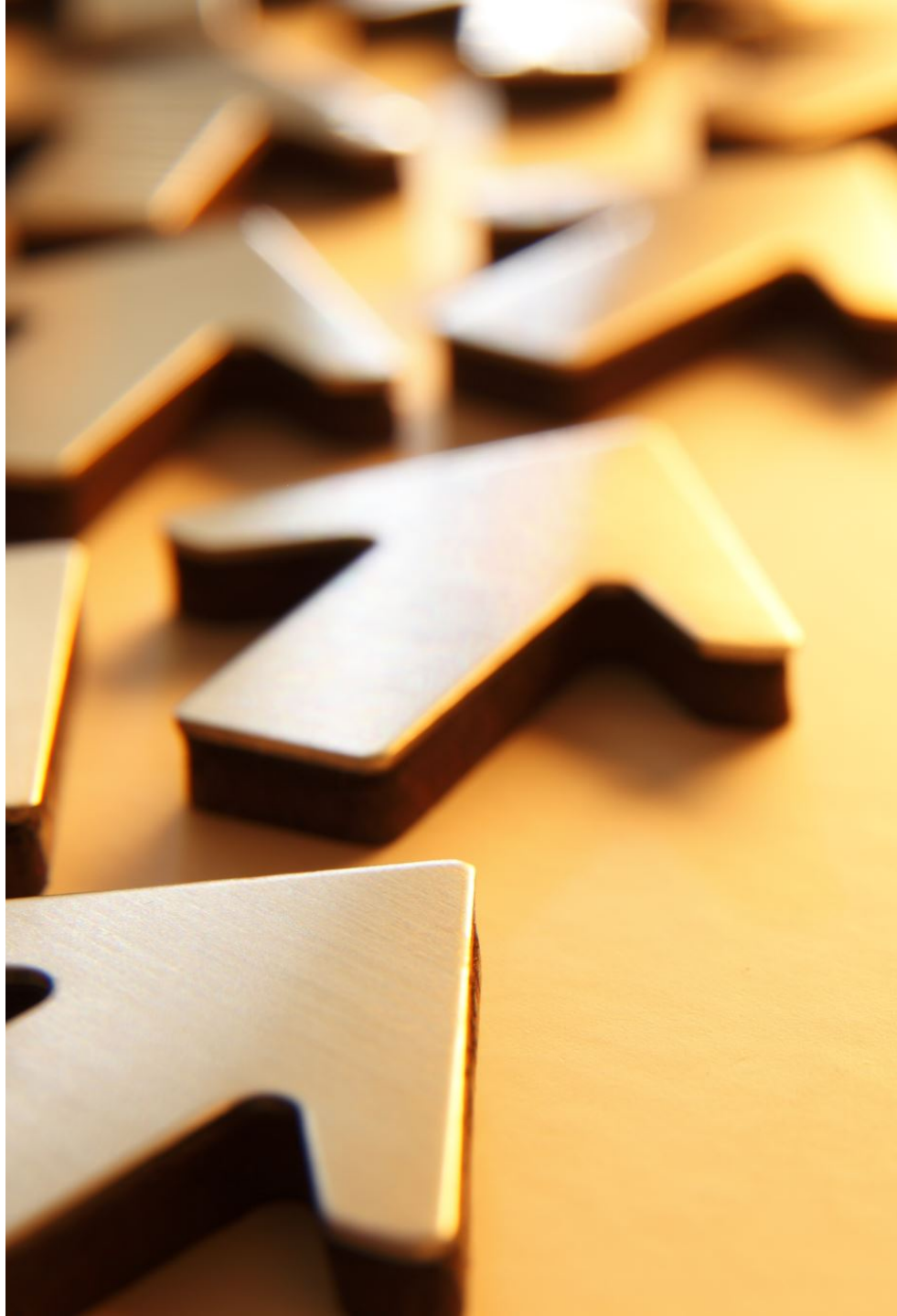
- for many application any arbitrary path will do – No Need for a Shortest Path
- Nodes are Densely Distributed
- Bidirectional Links
- Localized Communication
- Stationary Nodes
- Meet Trails of Queries and Events

- Attractive only when the ratio between events and queries is inside a threshold where it is not attractive to flood neither.
- Optimal parameters depend heavily on topology (but can be adaptively tuned)
- Does not guarantee delivery

AGENT PATH

Agent tries to travel in a “somewhat” straight path

- Maintains a list of recently seen nodes
- When it arrives at a node adds the node’s neighbors to the list
- For the next tries to find a node not in the recently seen list
- Avoids loops
- -important to find a path regardless of “quality”



FOLLOWING PATHS

A query originates from source, and is forwarded along until it reaches it's TTL

Forwarding Rules:

- If a node has seen the query before, it is sent to a random neighbor
- If a node has a route to the event, forward to neighbor along the route
- Otherwise, forward to a random neighbor using straightening algorithm

FAULT TOLERANCE

After agents propagated paths to events, some nodes were disabled

Delivery probability degraded linearly up to 20% node failure, then dropped sharply

Both random and clustered failure were simulated with similar results