Wireless Sensor Networks Routing Protocols

MO809 - Tópicos em Computação Distribuída

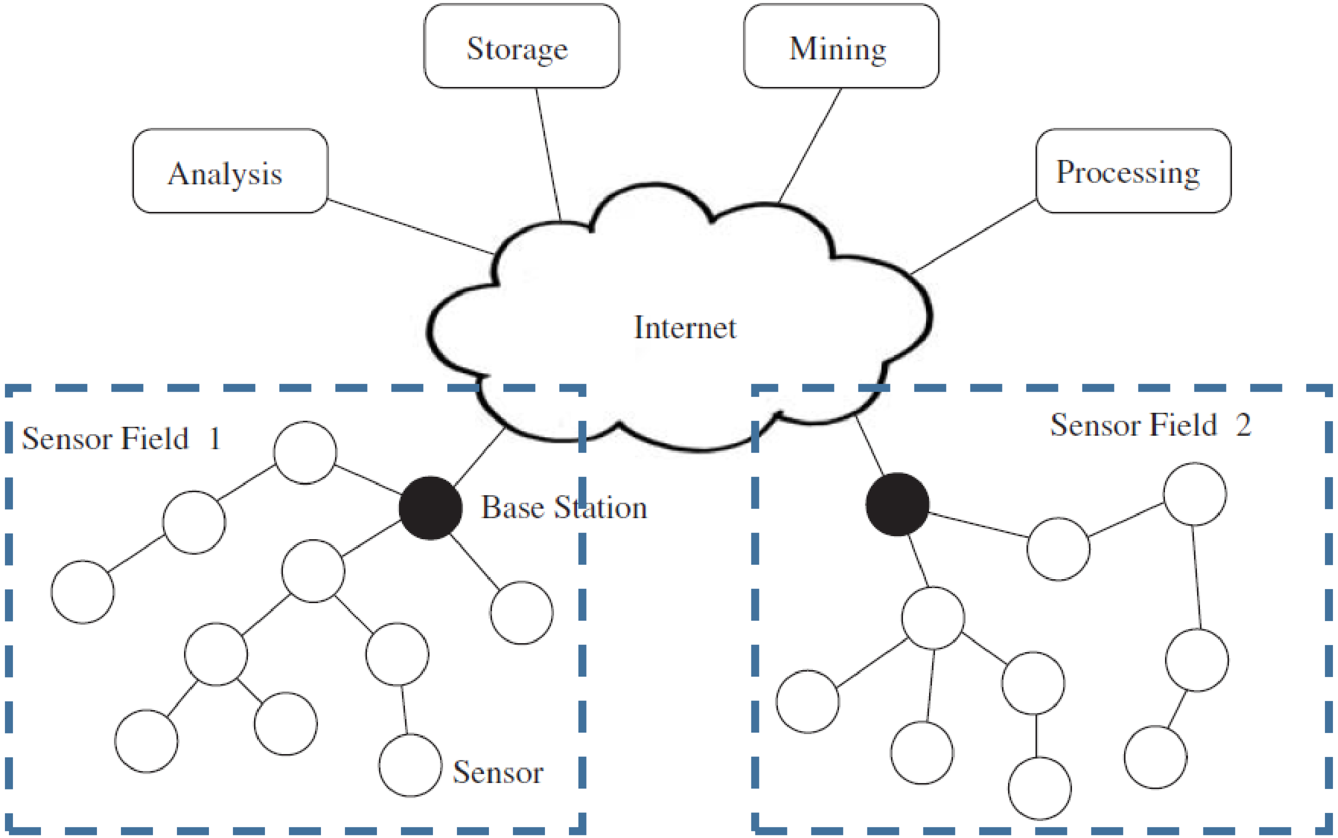
Luísa Madeira Cardoso

Topics

- Introduction to Wireless Sensor Networks
- Network Layer
- Characteristics: routing protocols
- Protocols
  - Data-Centric
  - Proactive
  - On-demand
  - Hierarchical
  - Location-based
  - QoS-based

Wireless Sensor Networks (WSN)

Sensors that cooperatively monitor large physical environments using their wireless radios to communicate

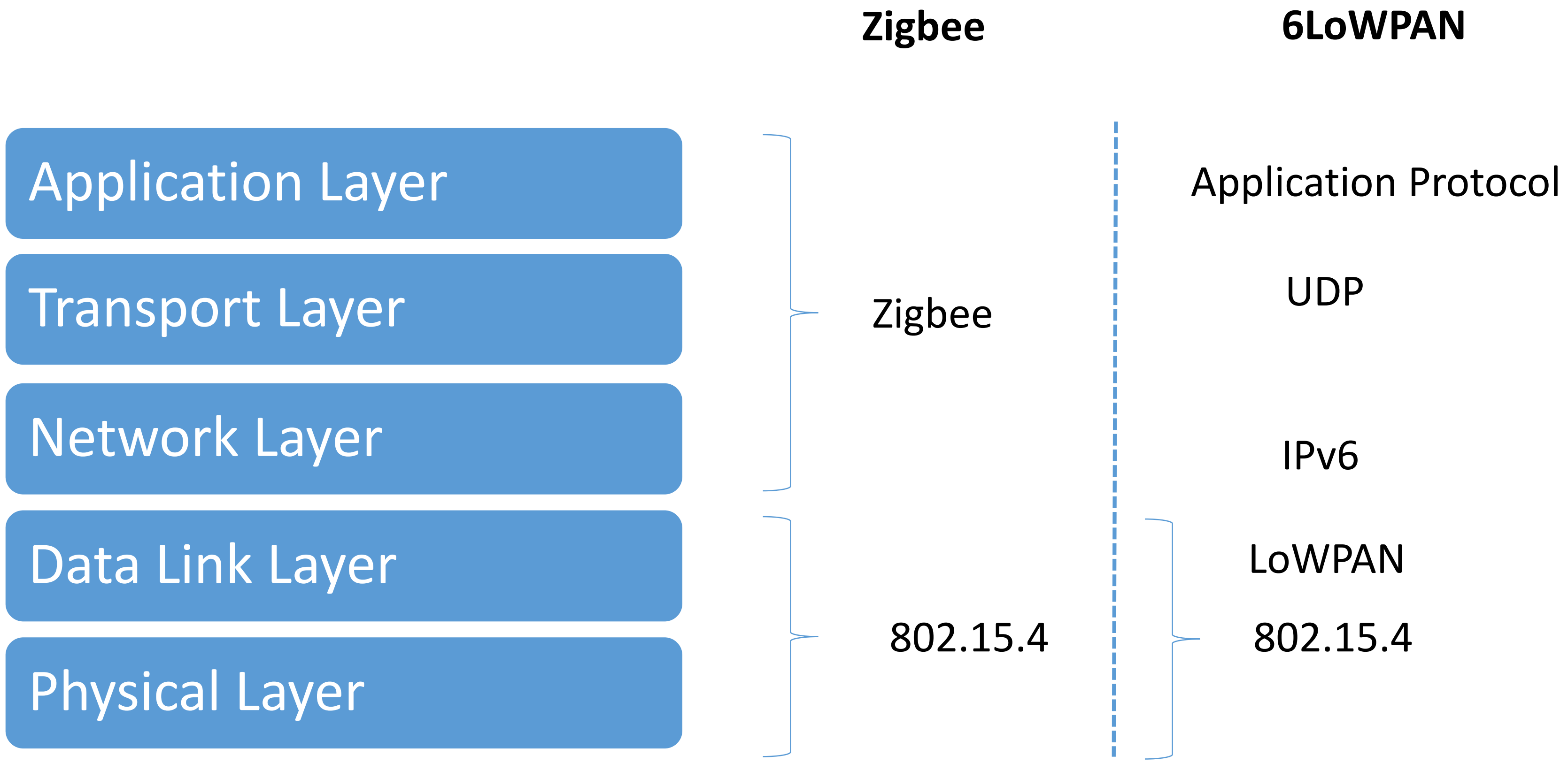


Traditional networks	Wireless sensor networks
General-purpose design; serving many applications	Single-purpose design; serving one specific application
Typical primary design concerns are network performance and latencies; energy is not a primary concern	Energy is the main constraint in the design of all node and network components
Networks are designed and engineered according to plans	Deployment, network structure, and resource use are often ad hoc (without planning)
Devices and networks operate in controlled and mild environments	Sensor networks often operate in environments with harsh conditions
Maintenance and repair are common and networks are typically easy to access	Physical access to sensor nodes is often difficult or even impossible
Component failure is addressed through maintenance and repair	Component failure is expected and addressed in the design of the network
Obtaining global network knowledge is typically feasible and centralized management is possible	Most decisions are made localized without the support of a central manager

Sensor Data Gathering Approaches

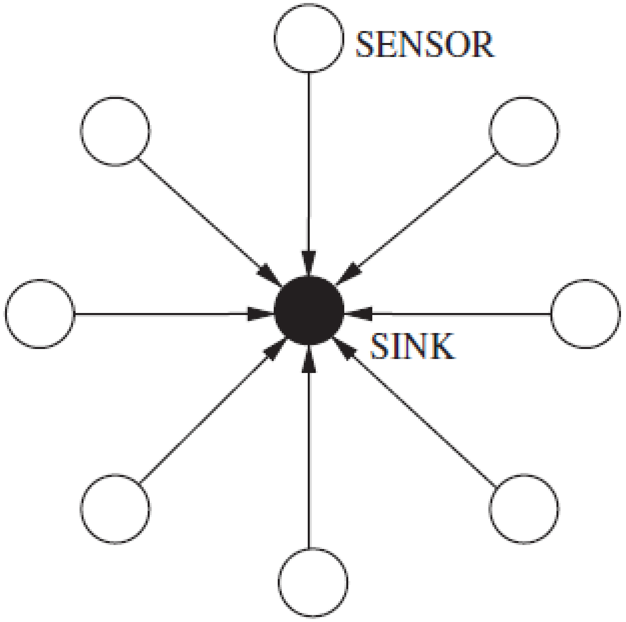
- Time-driven
  - e.g. Environmental monitoring
  - Periodically propagate their collected sensor data to a sink or gateway device
- Event-driven
  - e.g. Wildfire detection
  - Nodes only report their collected information when events of interest occur
- Query-driven
  - Is the responsibility of the sink to request data from sensors when needed

Basic Architectural Network Framework

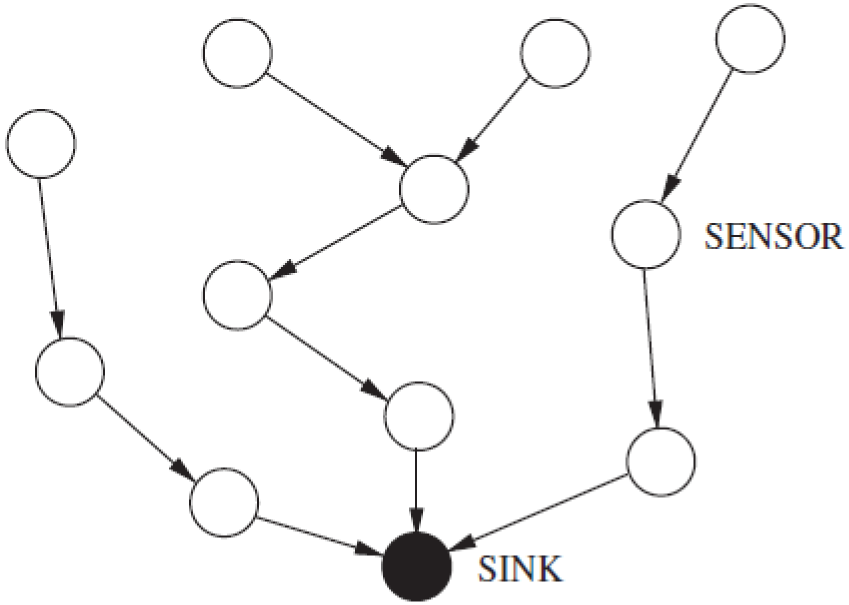


Network Layer

The key responsibility of the network layer is to find **paths** from data sources to sink (gateway) devices - routing



- Small sensor networks
- Sensor nodes and a gateway are in close proximity

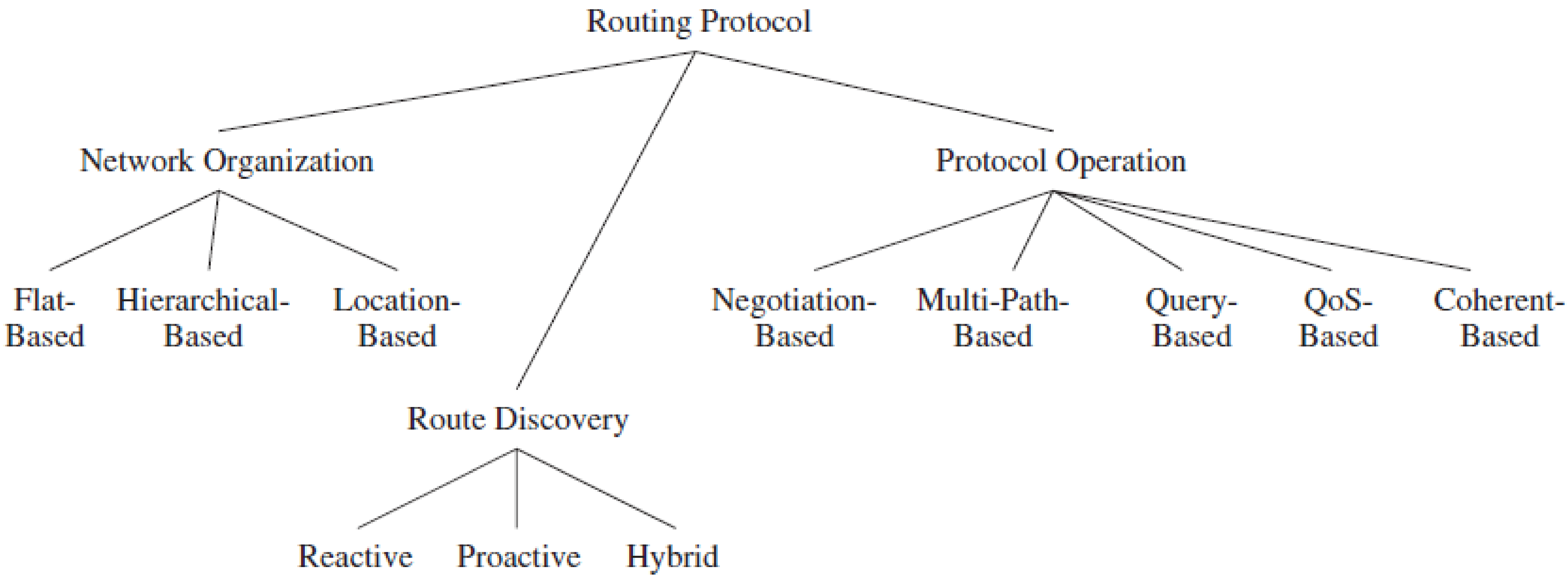


- Large numbers of sensor
- Cover large areas
- Multi-hop

When the nodes of a WSN are deployed in a deterministic manner (i.e., they are placed at certain predetermined locations), communication between them and the gateway can occur using predetermined routes.

However, when the nodes are deployed in a randomized fashion (i.e., they are scattered into an environment randomly), the resulting topologies are nonuniform and unpredictable. In this case, it is essential for these nodes to self-organize, that is, they must cooperate to determine their positions, identify their neighbors, and discover paths to the gateway device.

Routing protocols



Network Organization

Routing protocols

- **Flat-based** routing protocols consider all nodes of equal functionality or role.
- **Hierarchical-based** routing protocols, different nodes may assume different roles in the routing process, that is, some nodes may forward data on behalf of others, while other nodes only generate and propagate their own sensor data.
- **Location-based** routing protocols rely on the location information from nodes to make routing decisions.

Route Discovery

Routing protocols

- **Reactive** protocols discover routes *on-demand*, that is, whenever a source wants to send data to a receiver and does not already have a route established.
- **Proactive** routing protocols establish routes before they are actually needed. *table-driven*, because local forwarding decisions are based on the contents of a *routing table* that contains a list of destinations, combined with one or more next-hop neighbors that lead toward these destinations and costs associated with each next hop option.
- **Hybrid** exhibit characteristics of both reactive and proactive protocols.

Operation

Routing protocols

- **Negotiation-based** protocols aim to reduce redundant data transmissions by relying on the exchange of negotiation messages between neighboring sensor nodes before actual data transfers occur
- **Multipath-based** protocols use multiple routes simultaneously to achieve higher performance or fault tolerance
- **Query-based** routing protocols are receiver-initiated, that is, sensor nodes send data in response to queries issued by the destination node
- **QoS-based routing** protocols is to satisfy a certain Qualityof-Service (QoS) metric (or a combination of multiple metrics), such as low latency, lowenergy consumption, or low packet loss
- **Coherent-based** protocols perform only a minimum amount of processing (e.g., eliminating duplicates, time-stamping) before sensor data is sent to receivers and data aggregators. However, in noncoherent-based protocols, nodes may perform significant local processing of the raw data before it is sent to other nodes for further processing.

Routing Metrics

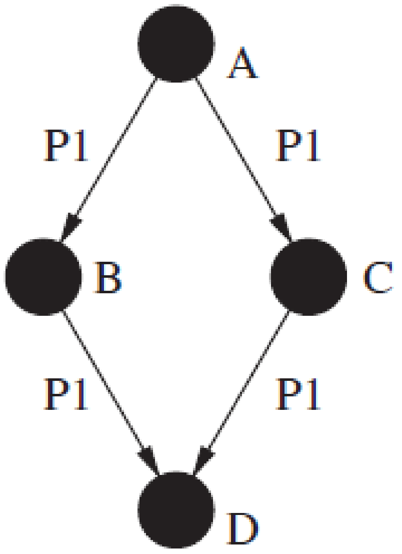
- **Minimum Hop:** smallest number of relay nodes (hops)
- **Energy**
  - *Minimum energy consumed per packet*
  - *Maximum time to network partition*
  - *Minimum variance in node power levels*
  - *Maximum (average) energy capacity*
- **Quality-of-Service:** performance in networks, including end-to-end latency (or delay) and throughput, but also jitter (variation in latency) and packet loss (or error rate).
- **Robustness:** stability and reliability for long periods of time

Protocol	Characteristics
SPIN	Flat topology, data-centric, negotiation-based
Directed diffusion	Flat topology, data-centric, query-based, negotiation-based
Rumor routing	Flat topology, data-centric, query-based
GBR	Flat topology, data-centric, query-based
DSDV	Flat topology with proactive route discovery
OLSR	Flat topology with proactive route discovery
AODV	Flat topology with reactive route discovery
DSR	Flat topology with reactive route discovery
LANMAR	Hierarchical with proactive route discovery
LEACH	Hierarchical, support of MAC layer
PEGASIS	Hierarchical
Safari	Hierarchical, hybrid route discovery (reactive near, proactive remote)
GPSR	Location-based, unicast
GAF	Location-based, unicast
SPBM	Location-based, multicast
GEAR	Location-based, geocast
GFPG	Location-based, geocast
SAR	Flat topology with QoS (real-time, reliability), multipath
SPEED	Location-based with QoS (real-time)
MMSPEED	Location-based with QoS (real-time, reliability)

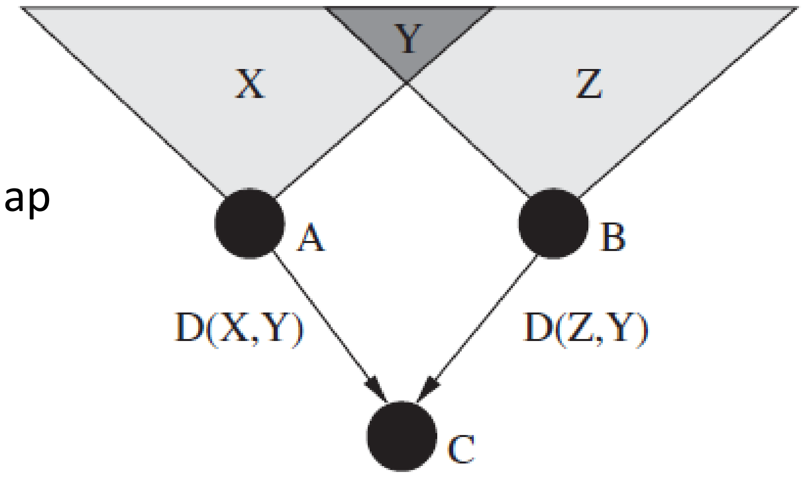
Flooding

- Sender node broadcasts packets to its immediate neighbors, which will repeat this process by rebroadcasting the packets to their own neighbors until all nodes have received the packets or the packets have traveled
- Very simple, but causes heavy traffic
- Problems:

Implosion



Overlap



Resource blindness

1. Data-Centric Protocols

- The sensor nodes themselves are less important than the information they generate
- Focus on the retrieval and dissemination of information of a particular type or described by certain attributes, as opposed to the data collection from particular sensors

Flat-based networks (no topology management is necessary)

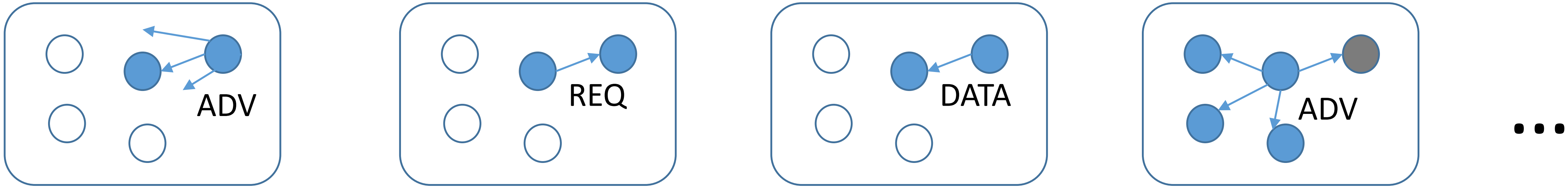
Protocols

1. SPIN
2. Direct Diffusion
3. Rumor Based
4. GBR

1.1 Sensor Protocols for Information via Negotiation (SPIN)

- Negotiation-based, data-centric, time-driven flooding
- To address the problems of implosion and overlap, SPIN nodes negotiate with their neighbors before they transmit data, allowing them to avoid unnecessary communications.
- To address the problem of resource blindness, each SPIN node uses a resource manager to keep track of actual resource consumption, allowing them to adapt routing and communication behavior based on resource availability
- Uses meta-data to succinctly and completely describe the data collected
- Meta-data < data-size; Two identical pieces of sensor data should have the same meta-data representation. The actual translation of sensor data to meta-data is application-specific and SPIN relies on each application to interpret and synthesize its own meta-data.

4 variants: SPIN-PP, SPIN-EC, SPIN-BC, SPIN-RL



SPIN-BC

1. Broadcast ADV (all nodes within a sender's transmission range will receive a copy of the message)
2. Upon receiving an ADV message, a node checks whether it wishes to receive a copy of the advertised data, and if so, sets a random timer, uniformly chosen from a predetermined interval. Only after the timer expires, the node issues the REQ message, again to the broadcast address (specifying the identity of the sender of the ADV message in the message header). When a node overhears he REQ message before its own timer expires, the node cancels its timer and does not send its own REQ message.
3. The advertiser transmits the advertised data to the broadcast address **only once**, that is, it will ignore duplicate REQ messages for the same data.

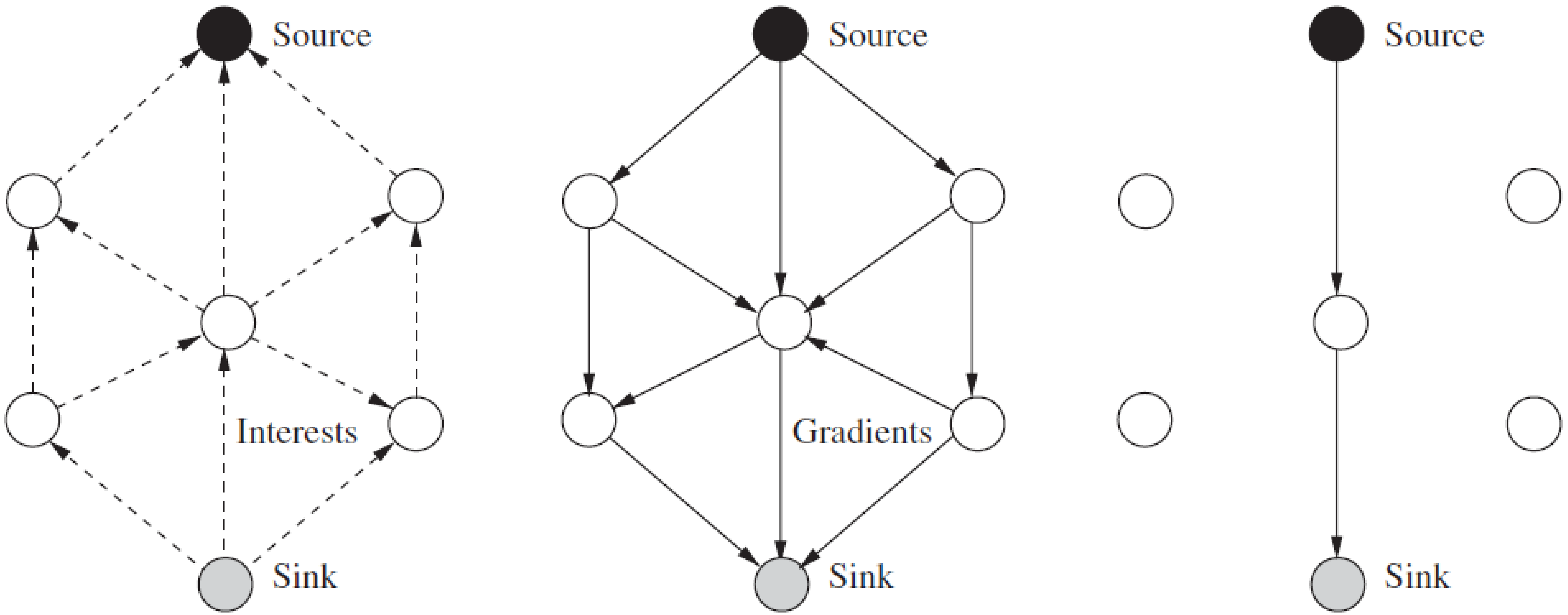
1.2 Directed Diffusion

- Data-centric, query-based, application-aware
- Data generated by sensor nodes is named by attribute-value pairs
- Nodes request data by sending *interests* for named data
- This interest dissemination sets up *gradients* within the network that are used to direct sensor data toward the recipient, and intermediate nodes along the data paths can combine data from different sources to eliminate redundancy and reduce the number of transmissions

Description for a simple vehicle-tracking application

```
type = vehicle                // detect vehicle location
rect = [-100,-100,200,200] // from sensors within rectangle
```

Task description expresses a node’s desire (or interest) to receive data matching the provided attributes



1. Sink node periodically broadcasts an interest message to its neighbors, which continue to broadcast the message throughout the network
2. Each node establishes a gradient toward the sink node, where a gradient is a reply link toward the neighbor from which the interest was received. As a consequence, using interests and gradients, paths between event sources and sinks can be established
3. Source begins to transmit data, it can use multiple paths for transmission toward the sink

1.3 Rumor Routing

Variation of directed diffusion that attempts to combine *event flooding and query flooding*

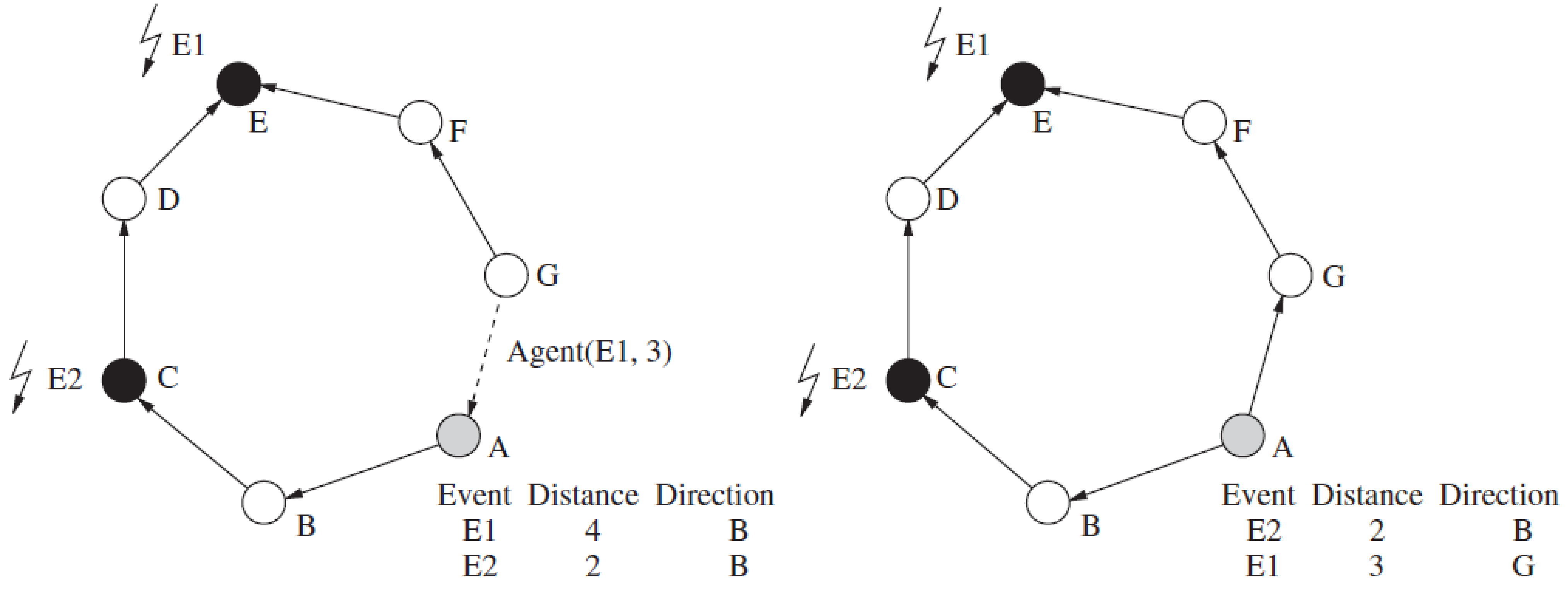
Each node maintains:

- A list of its neighbors
- An event table that contains forwarding information to all known events.

Query

Node checks whether it has a route toward the target event.

- Positive case: forwards the query to the neighbor indicated by the event's table entry
- Negative: a random neighbor is selected and the query is passed to this neighbor.



Event	Distance	Direction
E1	4	B
E2	2	B

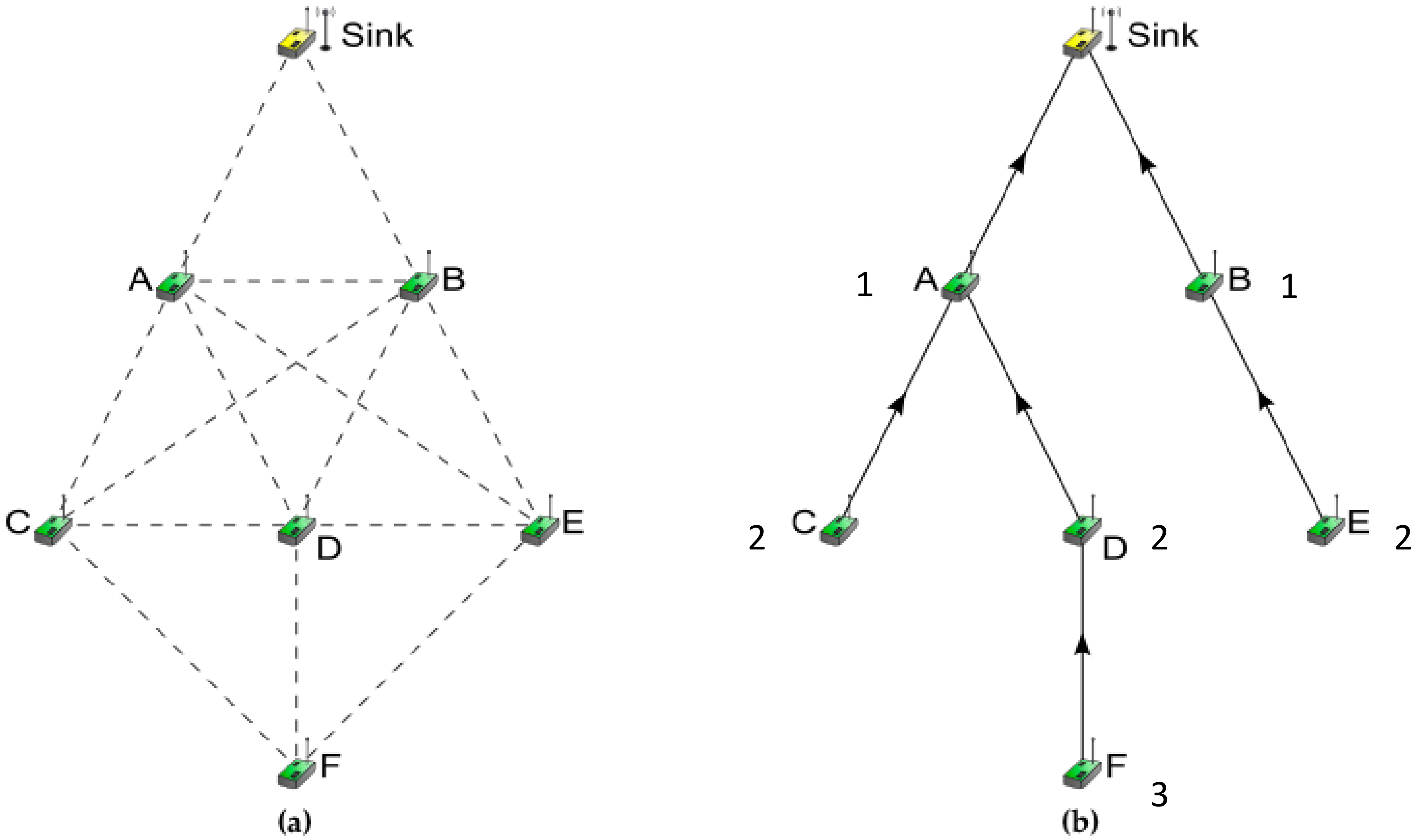
Event	Distance	Direction
E2	2	B
E1	3	G

Once a node witnesses an event, the event is added to this table (including a distance of zero) and an agent is generated with a certain probability. This agent is a long-lived packet that travels the network to propagate information about this event and other events encountered along its route to remote node.

1. Once an agent arrives at a node, the node can use the agent's content to update its own table. The table for node A pointing toward events E1 and E2 is shown, before the arrival of an agent originating at node E.
2. When the agent arrives at node A (via node G), A sees that E1 can be reached via neighbor G using a shorter route than currently stored in its table.
3. It therefore updates its table with the newly obtained information from E's agent.

1.4 Gradient Based Routing (GBR)

- Variant of the concept of directed diffusion
- Memorize the number of hops when interest is diffused
- Height: distance (in hops) to the sink
- Gradient: the difference between a node's height and the height of its neighbor.
- A data packet is then forwarded on the link with the **largest gradient**
- When multiple routes pass through a node, their data may be combined: Data Combining Entity (DCE)
- Traffic balancing
  - Stochastic-based – randomly selects the next hop (same gradient)
  - Energy-based – Node increases its height when detects energy has dropped below a threshold
  - Stream-based – New streams are diverted away from nodes that already serve other streams



Sink
A
B
C
D
E
F
(a)
Sink
1
A
B
1
2
C
D
2
E
2
F
3
(b)

2 Proactive Routing

Proactive (or table-driven) routing protocols establish paths before they are actually needed.

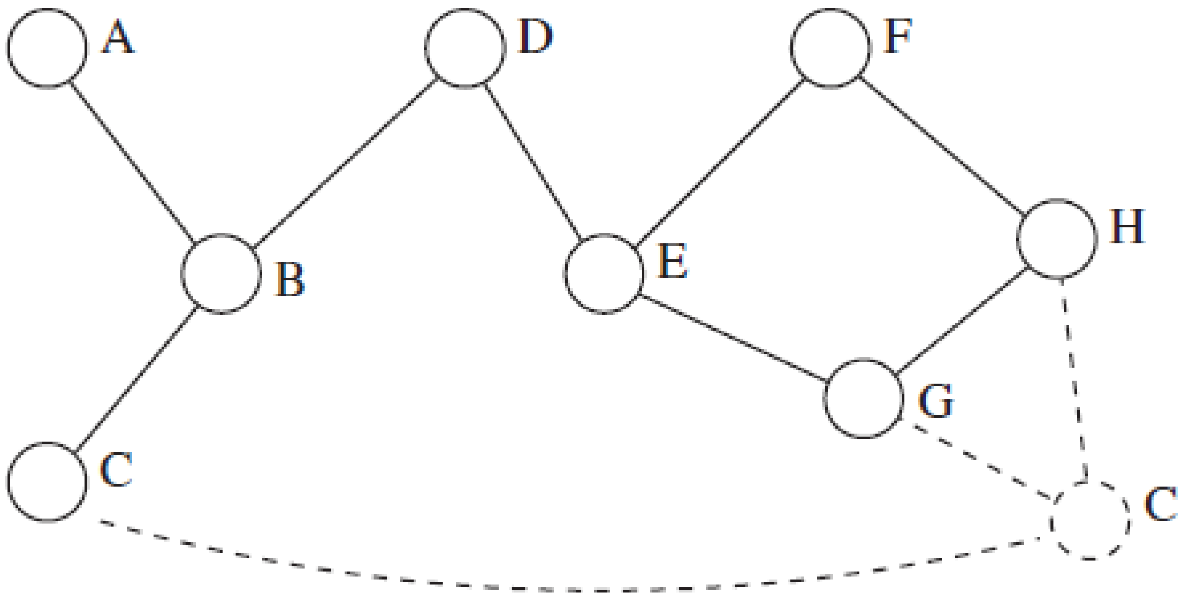
- Advantage: routes are available whenever they are needed
- Disadvantage: overheads involved in building and maintaining potentially very large routing tables and that stale information in these tables may lead to routing errors.

Protocols

1. Destination-Sequenced Distance Vector (DSDV)
2. Optimized Link State Routing (OLSR)

2.1 Destination-Sequenced Distance Vector (DSDV)

- Modified version of the classic Distributed Bellman-Ford algorithm.
- In distance-vector algorithms, every node *i* maintains a list of distances d_{ij}^{x} for each destination *x* via each neighbor *j* .
- Node *i* selects node *k* as the next hop for packet forwarding if $d_{ik}^{x} = \min\{d_{ij}^{x}\}$
- Stored in a routing table, along with a sequence number for each entry, where this number is assigned by the destination node. The purpose of the sequence numbers is to allow nodes to distinguish stale routes from new ones in order to prevent routing loops.
- Each node broadcasts updates for the routing table periodically, but also immediately whenever significant new information becomes available.



Destination	NextHop	Distance
A	B	2
B	B	1
C	B	2
D	D	0
E	E	1
F	E	2
G	E	2
H	E	3

Node D's Table Before C's Move

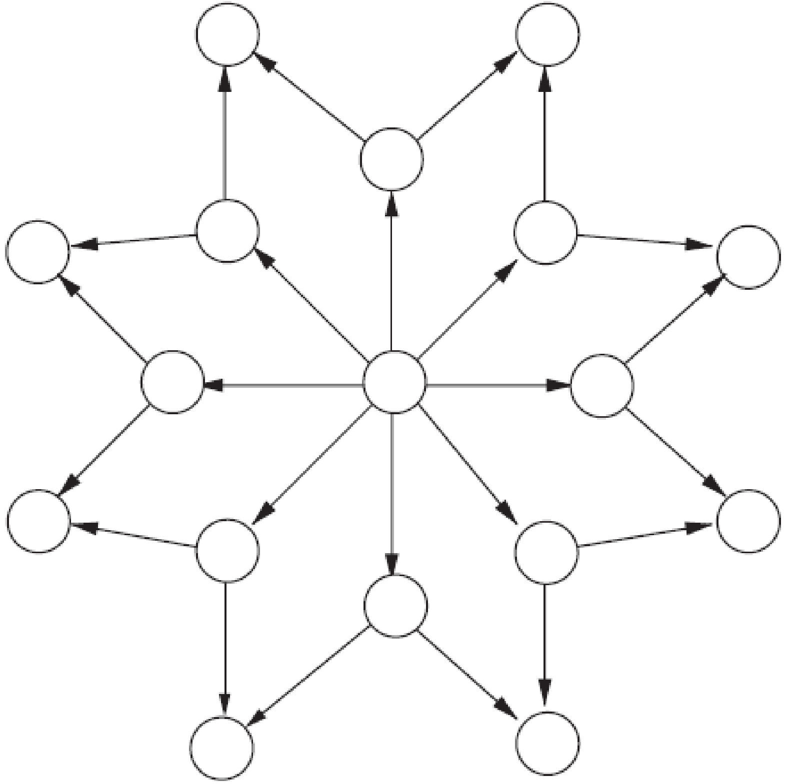
Destination	NextHop	Distance
A	B	2
B	B	1
C	E	3
D	D	0
E	E	1
F	E	2
G	E	2
H	E	3

Node D's Table After C's Move

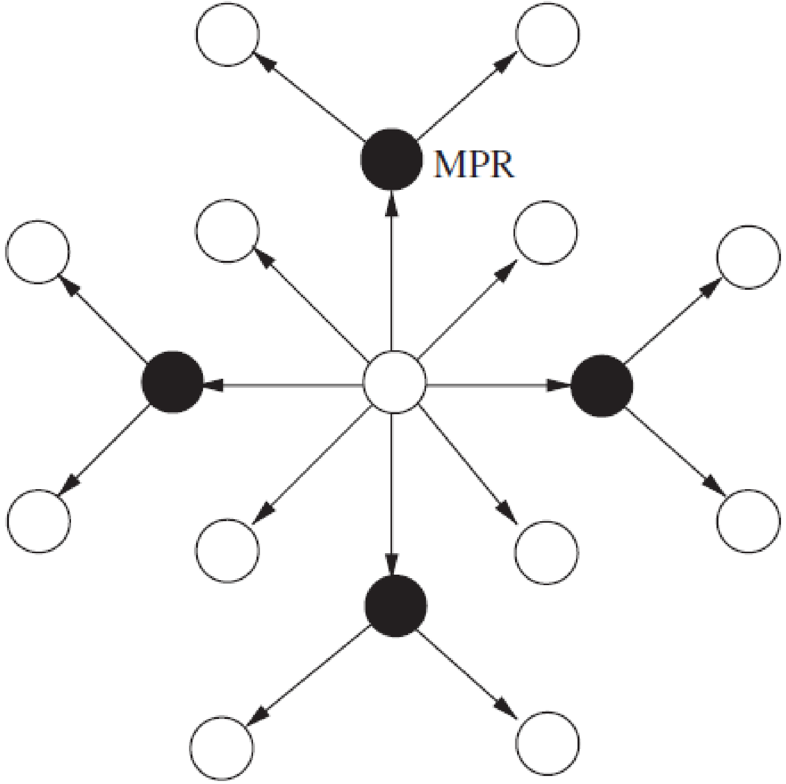
2.2 Optimized Link State Routing (OLSR)

- Based on the link state algorithm
- Nodes periodically broadcast topological information updates to all other nodes in the network allowing them to obtain a complete topological map of the network and to immediately determine paths to any destination in the network
- HELLO contains the node’s identity (address) and a list of all known neighbors of this node

Topological information must be flooded throughout the network. OLSR relies on *multipoint relays* (MPRs) to provide a more efficient way of disseminating such control information. **Only MPRs forward** messages to other nodes, which may significantly reduce duplicate transmissions



Classical flooding



OLSR - MPR

3 On-demand Routing

- Reactive protocols do **not discover and maintain routes** until they are explicitly requested.
- Route discovery process: completes when at least one route is found or when all possible routs have been examined.
- A route is then maintained until it either breaks or is no longer needed by the source

Protocols

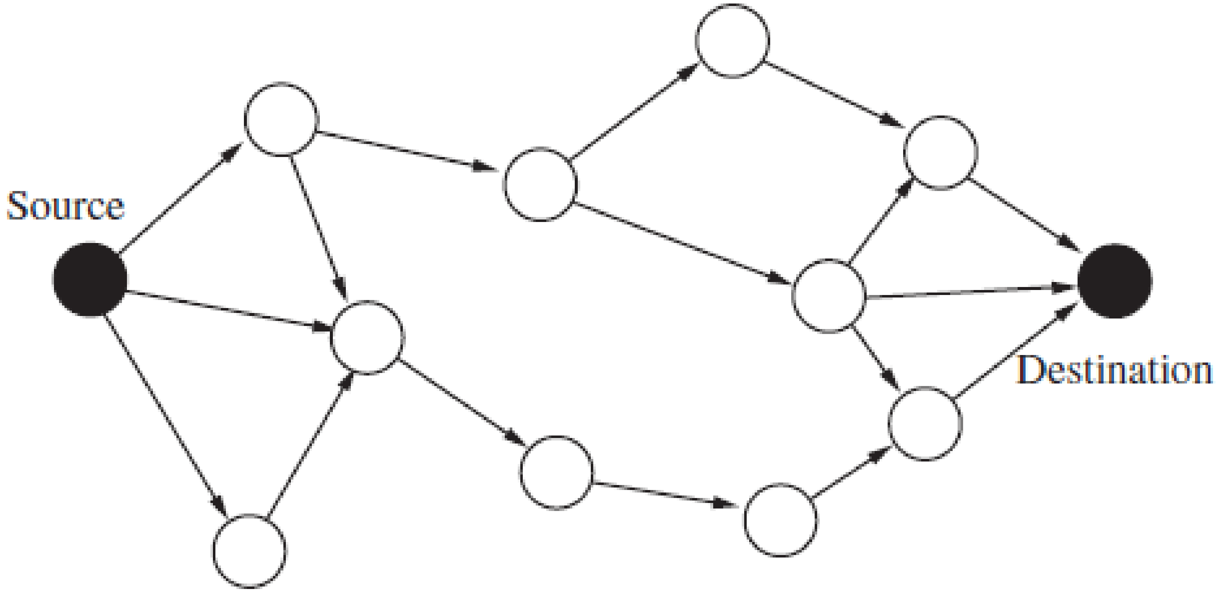
1. Ad Hoc On-Demand Distance Vector (AODV)
2. Dynamic Source Routing

3.1 Ad Hoc On-Demand Distance Vector (AODV)

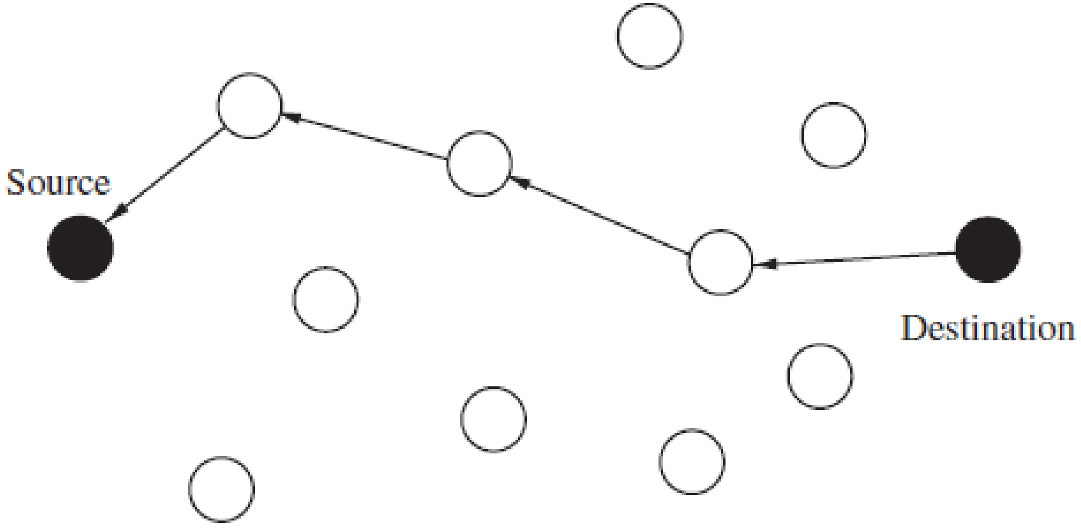
- Routes are only established when needed
- Has a route cache
- A timer for each entry in each node’s routing table limits the lifetime of unused routes
- Neighboring nodes also exchange periodic HELLO messages to monitor the state of their links
- When a link along the route breaks (e.g., due to a moving node), the intermediate node closer to the source noticing the broken link issues a *route error* (RERR) packet upstream

Path discovery process

1. Source node broadcasts a *route request* (RREQ) packet to its neighbors (addresses of the source and the destination, a hop count value, a broadcast ID, and two sequence numbers)
2. Upon receiving an RREQ packet:
    1. If the node has a route to the current destination: responds by sending a unicast *route reply* (RREP) message directly back to the neighbor from which the RREQ was received
    2. Else: the RREQ is rebroadcast to the intermediate node's neighbor



RREQ Packet Propagation



Path Taken By RREP Packet

3.2 Dynamic Source Routing (DSR)

- Route discovery and route maintenance procedures similar to AODV.
- Route cache
- Path discovery:
  - Broadcasting a route request packet, which contains the address of the destination, the address of the source, and a unique request ID
  - As this request propagates through the network, each node inserts its own address into the request packet before rebroadcasting it
  - As a consequence, a request packet records a route consisting of all nodes it has visited

References

- Waltenegus Dargie and Christian Poellabauer. 2010. *Fundamentals of Wireless Sensor Networks: Theory and Practice*. Wiley Publishing.
- Yick, Jennifer, Biswanath Mukherjee, and Dipak Ghosal. "Wireless sensor network survey." *Computer networks* 52.12 (2008): 2292-2330.