Content-based Networking
References...

In a wireless sensor network, a node is often interested in some information, but...

- Indifference of the information source
  - Does not matter which node observes an event and reports
  - Does not matter which sink receives the information
  - Timing of data request(s) and/or data availability is irrelevant
  - Asynchronous activities of source(s) and sink(s)
- The interaction in the network is “data-centric”
All nodes are connected on a “software bus”, to an event manager.

Data is made available publicly via a **publish** action.

Interested parties **subscribe** to appropriate data.

The parties may later **unsubscribe**.

When data becomes available “subscribed” parties are **notified**.

---

**Event Manager**

- **Storage and management of subscriptions**
- **Subscribe()**
- **Unsubscribe()**
- **Notify()**
Has three properties:
- Decoupling in space
- Decoupling in time
- Decoupling in flow (synchronization)
When subscribing to or publishing data, the question is *how to “refer” to this data?*

**Variants:**
- **Topic-based**
  - A set of keywords exist (e.g., stock names traded at a stock exchange)
- **Content-based**
  - Arbitrary predicates (e.g., “Is temperature reported in the events greater than 25°C?”)
  - Most important variant to capture the notion of “data-centric” networking
Implementation options:
- Centralized repository; inadequate
- Grouping based on topics; multicasting
  - problems...

The solution is “content-based forwarding and routing”
- Each node stores a predicate describing the set of predicates that the nodes in a sub-network have subscribed to
- The challenge is to keep the forwarding table up-to-date with minimum overhead
Distribution or gathering of data?

- Since publish/subscribe paradigm decouples the parties in space, distribution or gathering of data becomes irrelevant or less important.
- A technique called “convergecast” can be used.
  - Problem? **Implosion**...
- Data dissemination algorithms can be matched to application requirements.
- Gathering data → additional optimization (not *all* the data for *all* nodes need to arrive at the sink → **data aggregation**).
- Data aggregation is only one type of **in-network processing**.
Data-centric routing...

- **One-shot interactions**
  - Disseminating big data sets via SPIN
  - Active query processing

- **Repeated interactions**
  - Directed diffusion
  - Geographic scoping
  - Push diffusion
  - ...
• **Target scenario:**
  - One, several, or all nodes have data to be disseminated to the entire network
  - The data per node is relatively large (a unique name—small size—can be assigned)
  - Simple flooding is bad → implosion and overlap

• **Protocol:**
  - Data names are used to negotiate which nodes should forward which data
  - Negotiation (replace sending data in a flooding protocol with)
    - “Advertise” new data’s name to neighbors
    - If unheard, the neighbor “requests” the data
    - Transmit data only to the requester

• **Savings are based on the fact that the actual data is much larger than its name**

• **SPIN is a data-centric version of flooding**
ACQUIRE...

- ACtive QUery forwarding In sensoR nEtworks
- ACQUIRE is comparable to gossiping or rumor routing
- Protocol:
  - A query sent into the network is partially resolved as far as possible at an intermediate node
  - Forwarded onward with accumulated intermediate results, as long as not fully answered
  - Once full, the query transforms into a response and routes back to the issuer
  - An intermediate node can draw upon data from its $d$ hop away neighbors
    - This information is updated, when needed
    - May be used to guide the “forwarding” process
Repeated interactions...

- Directed diffusion (or two-phase pull)
- Publish/subscribe assisted by geographic scoping
- Push diffusion
  - Supporting few senders many receivers
- On-phase pull
  - Supporting many senders few receivers
- Directed diffusion assisted by topology control
- Multiple forwarding paths
  - Increased robustness
- Other issues
  - Scheduling, hierarchical dissemination, ...
Directed diffusion...

- **One possible realization of subscribe/publish paradigm**
  - Data generated by the nodes is named by “attribute-value” pairs
  - A node requests data by sending “interests” for named data
  - A gradient (direction state) list is setup
  - Data matching the interests is then “drawn” toward the node
  - Data can be cached, transformed, or directed by the intermediate nodes

![Diagram](image)
In data-centric networking, all messages need to be delivered to all sinks; have the ability to “operate” on data.

Simplest form of in-network processing is aggregation:
- Computing smaller representation of a number of messages; a mean or the max of the measured values.
- The actual benefit depends on the location of the data sources:
  - When data sources are spread out, the paths to the sinks don’t intersect, there is not much opportunity to aggregate.
  - When data sources are all nearby, the benefit is large.
Data aggregation efficacy...

- Judged by several metrics:
  - **Accuracy**—the difference between the resulting value at the sink and the true value
    - Can be represented as differences, ratios, statistics, or other values
  - **Completeness**—the percentage of all readings that are included in the final aggregate
  - **Latency**—the duration that the aggregate finally arrives at the sink
  - **Message overhead**—the main advantage of aggregation, but there is always a trade off between the metrics
Open questions for data aggregation ...

- Which aggregation functions can be used? Any categories?
- How can the tree be formed? Where and how should the aggregate points be placed?
- How long should a node wait for data from the neighbors (or children?)
- Aggregate action interface; how should it be?

- Answers follow...
A database interface...

- The WSN as a whole is considered as a virtual SQL database called sensors

```sql
SELECT { agg(expression), attributes } FROM sensors
WHERE { selectionPredicates }
GROUP BY { attributes }
HAVING { havingPredicates }
EPOCH DURATION i
```

- Nodes periodically measure, transmit, and aggregate information and the epoch duration marks the period of repetitions
- Other database-inspired query models for WSNs exist; e.g., COUGAR
Aggregation operations...

Can be distinguished according to the following:

- Representation of intermediate results
  A tuple (partial state records) should be exchanged

- Properties of the actual aggregate function
  Given two partial state records \(<x>\) and \(<y>\), aggregate function \(f\) computes a new state record \(<z> = f(<x>,<y>)\)
  - Properties of \(f\)
    - Duplicate sensitive
    - Summary or examplary
    - Composable
    - Partial state record behavior: distributive, algebraic, content-sensitive, holistic, unique, monotonic, timing aspects
Placement of aggregation points...

- The aggregation points must be placed carefully for maximum benefit
  - Too close to the source, or on unique paths does not help
- Direct diffusion does not necessarily result in a tree, but it is well suited to aggregation. As such, aggregation should occur as early as possible
- The aggregation is in a sense opportunistic
- If the routing structure is grown without regard to later aggregation, the resulting structure is no longer optimistic
When to stop waiting for more data?...

- When aggregating, nodes, as well as sinks, has to decide how long to wait for the data from the children.
- A node, knowing which of its neighbors are its children can wait for all.
- Constant times for each hop yields a relatively simple scheme.
- Random time make it a challenging problem.
Dissemination of aggregated data...

- Aggregation can be useful if aggregated data is broadcast with the entire WSN (every node would then know the maximum temperature)

- Alternatives:
  - Gossip-based
    - Naively fully distributed → every node sends its measurement to every other node
  - Continuous aggregation with adaptive accuracy
    - Similar to gossiping, but useful only with exemplary (e.g. minimum) aggregate functions
Information-directed routing and aggregation...

- A query that is injected into the network travels around and collects information as it is forwarded.
- The question is:
  - Which node to forward next?
  - Ideally, the answer would be the node that can contribute the most information, but...
  - Possible criteria:
    - Best average case
    - Maximize the worst case
    - Maximize best case
Examples...

- Information-Driven Sensor Query (IDSQ)
- Constrained Anisotropic Diffusion Routing
- Tiny Aggregation (TAG)
- Data funneling and coding by ordering
- PEGASIS
The central question is: which entity needs to know which data?

- Nothing is known → distribute the data in the entire network, or node that produced the data stores it
  - Problem is querier has to find the way to flood the network, if it does not know where the data is or which node stores the data
- The idea of “Data-Centric Storage (DCS)” is to let data itself describe where it is stored
  - Name of data → key to its location can be used to lookup
- Geographic Hash Table (GHT)