Wireless Sensor Networks
25th Lecture
07.02.2007

Christian Schindelhauer
schindel@informatik.uni-freiburg.de

University of Freiburg
Computer Networks and Telematics
Prof. Christian Schindelhauer
Final Meeting (before the exams)

- Meeting Point: Waldkirch, main station
- Date: Tuesday 27.02.2006 14:01
  (Train departs Freiburg main station at 13:40)

- Plan
  - Hike the Kastelburg
  - Picknick

- BYOF
  - Order drinks on-line

- Don’t forget:
  - Food
  - Umbrella
  - Matches
Data-centric and content-based networking

- Interaction patterns and programming model
- *Data-centric routing*
- Data aggregation
- Data storage
One-shot interactions with big data sets

- **Scenario**
  - Large amount of data are to be communicated – e.g., video picture
  - Can be succinctly summarized/described

- **Idea: Only exchange characterization with neighbor, ask whether it is interested in data**
  - Only transmit data when explicitly requested
  - Nodes should know about interests of further away nodes

→ *Sensor Protocol for Information via Negotiation (SPIN)*
SPIN example

1. ADV
2. REQ
3. DATA
4. ADV
5. REQ
6. DATA

Wireless Sensor Networks

07.02.2007 Lecture No. 25 - 5
Repeated interactions

- More interesting: Subscribe once, events happen multiple times
  - Exploring the network topology might actually pay off
  - But: unknown which node can provide data, multiple nodes might ask for data
  → How to map this onto a “routing” problem?

- Idea: Put enough information into the network so that publications and subscriptions can be mapped onto each other
  - But try to avoid using unique identifiers: might not be available, might require too big a state size in intermediate nodes

→ Directed diffusion as one option for implementation
  - Try to rely only on local interactions for implementation
Directed diffusion – Two-phase pull

- Phase 1: nodes distribute interests in certain kinds of named data
  - Specified as attribute-value pairs (cp. Chapter 7)
- Interests are flooded in the network
  - Apparently obvious solution: remember from where interests came, set up a convergecast tree
  - Problem:
    - Node X cannot distinguish, in absence of unique identifiers, between the two situations on the right
    - set up only one or three convergecast trees?
Direction diffusion – Gradients in two-phase pull

- Option 1: Node X forwarding received data to all “parents” in a “convergecast tree”
  - Not attractive, many needless packet repetitions over multiple routes
- Option 2: node X only forwards to one parent
  - Not acceptable, data sinks might miss events

- Option 3: Only provisionally send data to all parents, but ask data sinks to help in selecting which paths are redundant, which are needed
  - Information from where an interest came is called gradient
  - Forward all published data along all existing gradients
Gradient reinforcement

- Gradients express not only a link in a tree, but a quantified “strength” of relationship
  - Initialized to low values
  - Strength represents also rate with which data is to be sent
- Intermediate nodes forward on all gradients
  - Can use a data cache to suppress needless duplicates
- Second phase: **Nodes that contribute new data (not found in cache) should be encouraged to send more data**
  - Sending rate is increased, the gradient is **reinforced**
  - Gradient reinforcement can start from the sink
  - If requested rate is higher than available rate, gradient reinforcement propagates towards original data sources
- Adapts to changes in data sources, topology, sinks
Directed diffusion – extensions

- Two-phase pull suffers from interest flooding problems
  - Can be ameliorated by combining with topology control, in particular, passive clustering

- Geographic scoping & directed diffusion

- Push diffusion – few senders, many receivers
  - Same interface/naming concept, but different routing protocol
  - Here: do not flood interests, but flood the (relatively few) data
  - Interested nodes will start reinforcing the gradients

- Pull diffusion – many senders, few receivers
  - Still flood interest messages, but directly set up a real tree
Data-centric and content-based networking

- Interaction patterns and programming model
- Data-centric routing
- Data aggregation
- Data storage
Data aggregation

- Any packet not transmitted does not need energy
- To still transmit data, packets need to combine their data into fewer packets → aggregation is needed
- Depending on network, aggregation can be useful or pointless
Metrics for data aggregation

- **Accuracy**: Difference between value(s) the sink obtains from aggregated packets and from the actual value (obtained in case no aggregation/no faults occur)

- **Completeness**: Percentage of all readings included in computing the final aggregate at the sink

- **Latency**

- **Message overhead**
How to express aggregation request?

- One option: Use database abstraction of WSN
- Aggregation is requested by appropriate SQL clauses

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

- WHERE: filter on value before entering aggregation process
  - Usually evaluated locally on an observing node
- GROUP BY: partition into subsets, filtered by HAVING
  - GROUP BY floor HAVING floor > 5
Partial state records

- Partial state records to represent intermediate results
  - E.g., to compute average, sum and number of previously aggregated values is required – expressed as <sum,count>
  - Update rule: \(<s, c> \rightarrow <s_1 + s_2, c_1 + c_2>\)
  - Final result is simply s/c
Aggregation operations – categories

- Duplicate sensitive, e.g., median, sum, histograms; insensitive: maximum or minimum
- Summary or exemplary
- Composable:
  - for $f$ aggregation function, there exist $g$ such that
    $f(W) = g(f(W_1), f(W_2))$ for $W = W_1 \cup W_2$
- Behavior of partial state records
  - Distributive – end results directly as partial state record, e.g., MIN
  - Algebraic – p.s.r. has constant size; end result easily derived
  - Content-sensitive – size and structure depend on measured values (e.g., histogram)
  - Holistic – all data need to be included, e.g., median
- Monotonic
Placement of aggregation points

- Convergecast trees provide natural aggregation points
- But: what are good aggregation points?
  - Ideally: choose tree structure such that the size of the aggregated data to be communicated is minimized
  - Figuratively: long trunks, bushy at the leaves
  - In fact: again a Steiner tree problem in disguise
- Good aggregation tree structure can be obtained by slightly modifying Takahashi-Matsuyama heuristic
- Alternative: look at parent selection rule in a simple flooding-based tree construction
  - E.g., first inviter as parent, random inviter, nearest inviter, …
  - Result: no simple rule guarantees an optimal aggregation structure
- Can be regarded as optimization problem as well
Alternative: broadcasting an aggregated value

- Goal is to distribute an aggregate of all nodes’ measurements to all nodes in turn
  - Setting up $|V|$ convergecast trees not appropriate

- Idea: Use gossiping combined with aggregation
  - When new information is obtained, locally or from neighbor, compute new estimate by aggregation
  - Decide whether to gossip this new estimate, detect whether a change is “significant”
Thank you

and thanks to Holger Karl for the slides