Analysis of issues for the Wireless Sensor Networks' middleware of the future

By: Michele Albano

Supervisor: Stefano Chessa
Wireless Sensor Networks issues

- Sensors are micro-systems characterized by:
  - Small CPU (8 MHz) and memory (4KB RAM, < 1MB total)
  - Sensing units (light, pressure, temperature, etc)
  - Wireless interface (802.15.4, up to 250 kbps)
  - Battery powered, low cost, prone to failures

- Applications based on multi-hop ad-hoc networks:
  - Goals: collect data unattended, send them outside
  - Problems: variable unknown topology, limited energy budget, single sensors can die
Middleware for WSNs

Goals: cope with these common issues, abstract WSN applications from the underlying complexity, let developers focus on business logic.

Based on:
- Broadcast & Geographic routing (GPSR with GPSes or virtual coordinates)
- In-network storage of data for future access (DCS, data d is indexed by their meta-datum k, store(k,d) & retrieve(k))
  - Set of sensors for meta-datum k is function of k
  - In-network storage also saves energy
- Data replication to enforce availability
Publish/subscribe models, and QoS

• Broadcast, unicast & DCS are too low-level => publish/subscribe models:
  • Advertise\( (k) \), Subscribe\( (k) \), Publish\( (k,d) \)
  ○ Not all the three primitives are always needed

• QoS problem: DCS-GHT stores on a number of nodes that depends on topology

Q-NiGHT: \( \text{store}(k,d,q) \) stores data on the \( q \) nodes that are closest to \( \text{Hash}(k) = (x,y) \)
Data redundancy via erasure codes

- Pure replication is inefficient
- n out of m erasure codes:
  - data d is encoded into m=n+r fragments $f_1, \ldots, f_m$ using m keys
  - reduced memory overhead in the sensors: each fragment is $1/n$ of original data
  - d can be reconstructed from any subset of n different fragments
- Key distribution:
  - each sensor is pre-assigned a random key from a pool
  - correct reconstruction depends on size of the pool, and on q in store(k,d,q)

Case study on Q-NIGHT:
Probability of losing data, given a storage load on sensors
- Replication (worse)
  - n=3, pool size=15
  - n=5, pool size=15 (best)