
On Developing Dependable Data Gathering in Wireless Sensor Networks

Françoise Sailhan, CNAM

Joint Work with : Thierry Delot (Univ. Valenciennes),
Animesh Pathak (INRIA), Aymeric Puech (SmartGrains),
Matthieu Roy (LAAS)

Outline

☞ Background – Wireless Sensors

■ Murphy project:

- Developing dependable wireless sensor networks
- Use cases:
 - Parking lot monitoring
 - Emergency situation

■ Conclusion

Wireless Sensors

- Networked sensor are intended to :
 - Monitor a phenomenon (sensing)
 - Perform an action (actuation)
 - Focus on wireless-enabled gathering, macro-programming platforms
 - Key constraint : resource/energy
 - Focus on energy-, computationally-, memory-aware algorithms, hardware and protocols
-

Wireless Sensors - Deployments

- Few long term deployment campaigns
 - Indoor: health care, smart home
 - Fairly well domesticated
 - Outdoor
 - Agricultural control : potatoes & tomatoes fields
 - Structural health monitoring : high mountains
 - Subject to a variety of non-envisioned **natural faults**
-

Wireless Sensors - Natural Faults

- A major class of sensor-specific faults whose origin:
 - External (water infiltration, sunlight)
 - Internal (power transient)
 - Phase of creation: operation
 - They are discovered on the field
 - The latter is the discovery, the higher is the impact
 - Un-envisioned faults (accidental & human nature)
 - Lack of adequate packaging
 - Use of off-the-shelf components
-

Wireless Sensors – Other faults

- Interaction faults
 - Partial coverage (leafs)
 - Channel sharing (overhead)
 - Resource exhaustion
 - The higher is the transmission range, the better is the coverage =>less sensor are needed to the expense of 3 hours lifetime
 - Verbose communication
 - Development faults
-

Wireless Sensors - Fault Manifestation

(1/2)

- Manifestation of faults are data centric
 - Faults are revealed at the fusion center
 - Fault are circumvented by applying data-mining approaches
 - Sensors constitute black boxes
 - No diagnosis is provided
 - They are seen as external (out of control) components rather than internal/integrated components
-

Wireless Sensors - Fault Manifestation

(2/2)

- Distributed macro-programming, gathering, data base middleware are designed so as to hide errors
 - Fault tolerance is mostly provided through distribution, replication, multi-paths, redundancy
 - No exception-oriented language/platform
 - Absence of distributed alarm system & handling
 - Developing dependable sensor networks is challenging
-

On developing **Dependable** Sensor Networks

- trust
 - Fault prevention: prevents the occurrence of faults
 - Rigorous design: formal specification, modeling
 - Fault-tolerance: avoids failure despite the presence of fault
 - Resource-aware gathering and micro-programming
 - confidence
 - **Fault removal & forecast**: reduce the number of faults and their severity while estimating and forecasting fault impact
 - **Murphy project**
-

Murphy Project

- Why Murphy ?
 - “anything that can go wrong will go wrong”
 - Objective: facilitate the development of dependable sensor networks based on rigorous testing
 - Difficulties:
 - (Dormant) faults are not easily detectable unless they cause a failure
 - Faults are not easily reproducible
-

Murphy Project

- Approach: a fault injector
 - Accelerates the occurrence of faults
 - Eases the testing
 - Actually: the project is in its design phase
 - Faults, requirements are identified
 - Evaluation: two practical case-studies
 - Parking lot monitoring
 - Emergency & rescue scenario
-

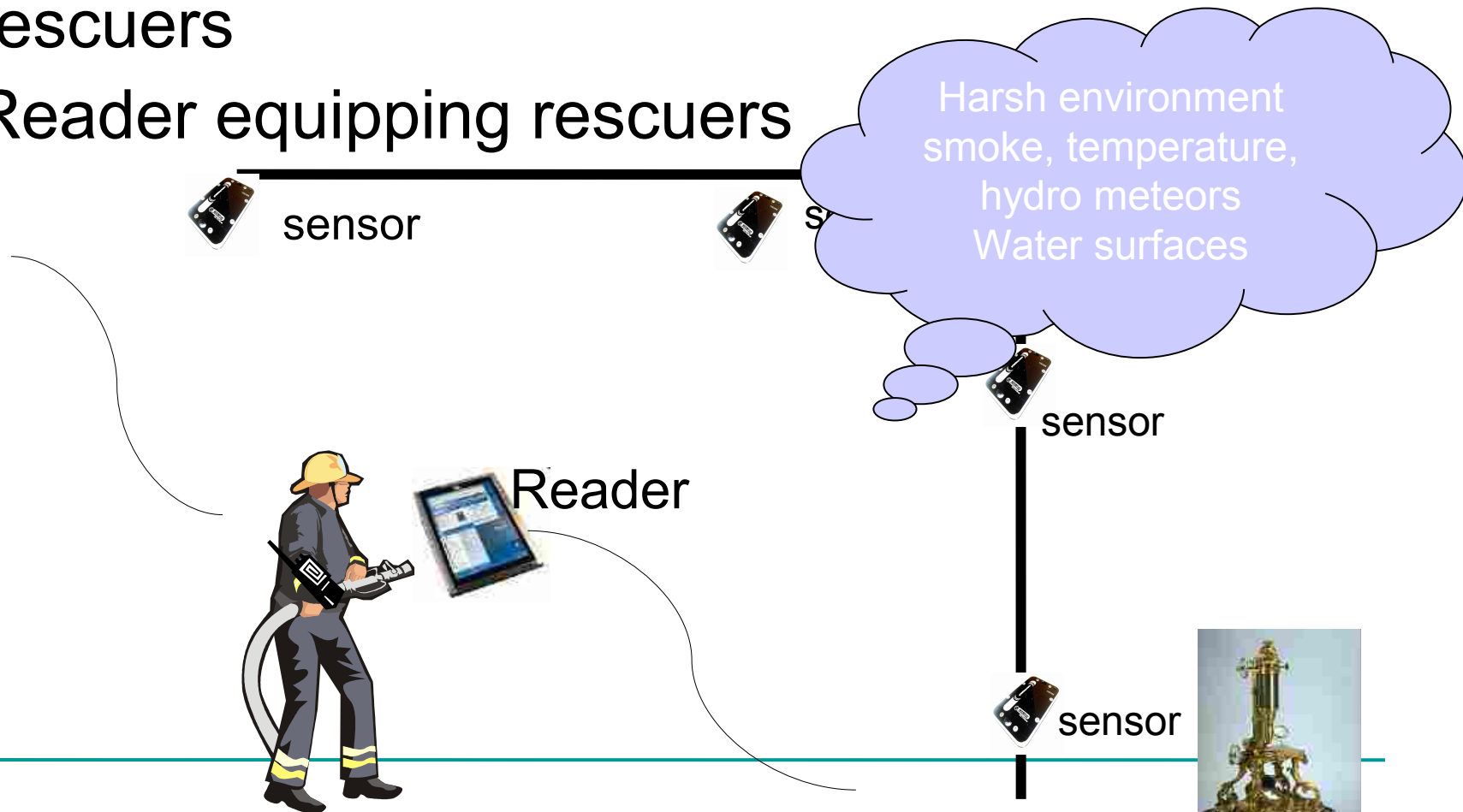
Parking Lot Monitoring System

Monitoring the occupancy of parking lots

- ❑ *A sensor monitors the presence of a car*
 - ❑ *A monitoring station gathers (remote) sensed data*
 - *Property that should be provided:*
 - ❑ (Long term) availability:
 - ❑ Prevent communication & low energy-consuming protocols
 - ❑ Wireless mesh backbone composed of battery-powered nodes
 - ❑ Provide flushing, booting mechanisms
 - ❑ Integrity
 - ❑ Not only a matter of networking or storage integrity
 - ❑ Can be traced back to the calibration of the sensor, harsh conditions, up to wireless transfer/retrieval
-

Emergency situation in the CNAM Museum

- Sensors embedded on art-crafts, walls, rescuers
- Reader equipping rescuers



Fire Fighter & Art-Craft Localisation

Triangulate the position of a firefighter & estimate the distance between firefighters and art-crafts

- Analyze the signal strength for each sensor
 - Deduct the distance separating the tag and the reader
 - Triangulate the position of the firefighter
 - Guide the firefighter towards available art-crafts
 - Properties that should no be violated :
 - Safety
 - Survivability
-

Geo-localisation – Safety

- Safety: ability to protect the firefighters against the effect of localization errors
 - Geo-localization precision (granularity of the estimation)
 - Provide context-aware geo-localization
 - Estimate of a couple of meters is sufficient
 - Geo-localization accuracy:
 - False positive
 - False negative
 - Degree of confidence (accuracy, precision) provided
-

Geo-Localization – survivability

- Survivability: ability to survive to failure during the sinister time-scale
 - Sensors disappear as the sinister evolves
 - Fail-verbose (in opposition of fail-silent or fail babbling)
 - Still, **testing** remains the only truly effective means for ensuring that properties are not violated
-

Which Faults?

- Define which faults are relevant focusing on the use cases
 - Networking faults: signaling, communication, network covering issues, noise
 - High impact on the geo-localization precision
 - Data-oriented faults:
 - Corruption of packets, messages, data
 - Generation of workload
 - Provide to user the ability to select a fault, or a combination of these faults
-

How ?

- Faults should be injected at
 - **Software level** : code and data are corrupted, additional package are implemented to inject faults
 - Hardware level (e.g. electric, electromagnetic interferences)
 - Faults are inserted within
 - A simulated platform (cheaper, easier)
 - A **physical platform** (less flexible but more realistic for networking faults)
-

Where ?

- Inserting faults on a single sensor is not sufficient
 - Geo-localisation is based on triangulation
 - Remote parking lot monitoring
- Insert faults in a distributed manner
 - Convenient abstractions of the networks, e.g., cluster head, gateways, % of nodes to reach
 - Macro-programming to remotely insert faults
 - 👍 The user is not required to modify the implementation of the system under study
 - 👍 High decoupling: the code relating to fault injection is separated from the tested protocol/application
 - 👍 It prevents from reusing part of the investigated system

And then?

- Provide in-network distributed monitoring
 - ❑ Should not be based on the global state of the targeted system
 - ❑ Keep a local view of the overall system
 - ❑ Post analysis is then applied to
 - Ensure that the proper fault insertion has been provided
 - Provide low intrusive **fault validation**
 - Diagnose and validate the predefined properties
-

Conclusion

- Wireless sensor networks suffer from dependability issues
 - Testing still remains a truly enabler of dependability
 - There is a need for a software fault injector
 - Permits to verify fault tolerance
 - Reproduces easily faults
 - Validates software operation over a full input range variance, noise and parameterized drifts
-