ARUM: an Approach for the Resilience of Ubiquitous Mobile Systems

Marc-Olivier Killijian
David Powell
Nicolas Rivière
Matthieu Roy
Gaétan Séverac
Christophe Zanon

Workshop Internet of Things - october 21st 2008

Internet of Things

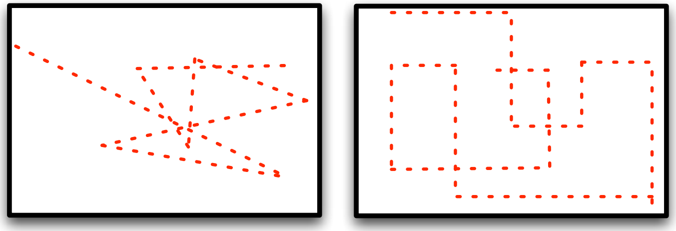
- Internet of Things
  - Ambiant Intelligence,
  - Ubiquitous Systems
- So what ?
  - Connecting zillions of nodes
  - Always evolving
    - Disconnections/new services, versioning, etc.
  - Open to new (untested/malicious/...) things
  - Mobile things/objects/devices
  - Tightly integrated into users environment/life
  - Critical as a whole

ARUM Motivation

- Need to increase resilience of AmI/IoT/UbiSys
- Evaluation of such systems is difficult
  - Usability, Acceptance, Performance, Resilience
  - Analytically: need for models, tools, etc.
  - Experimentally: need for platforms, prototypes, benchmarks, metrics, data, applications, etc.
- Why is it so difficult ?
  - What are ubiquitous systems ?
  - Scale (#/size devices/environment)
  - What to evaluate ? (user xp, network, etc.)

Evaluation vs. Simulation

- Evaluation usually based on simulation
  - Network simulators (ns/2, glomosim, etc.)
  - Simple mobility models
    - Random drunken
    - Manhattan



- Simulation is not satisfactory for critical systems
  - Cavin et al. 2002 : "*significant divergence exist between the [different] simulators*"
  - Very little work on simulation/fault-injection at node level [Goswami97]
- Mobility models meaningless for most applications

ARUM : emulation & realistic models/ parameters

- Emulation as a complement to simulation
  - Real prototype of a complex software stack
    - Runs on real hardware, i.e. uses real device drivers, communication stacks, etc.
    - Embeds real middleware, real FT mechanisms
    - Is a full prototype of the application
  - No details can be simplified
  - Fault-injection
- Use of real mobility/usage data for
  - Realistic mobility/failure/usage models
  - Realistic analytical evaluation parameters

ARUM : emulation & realistic models/ parameters

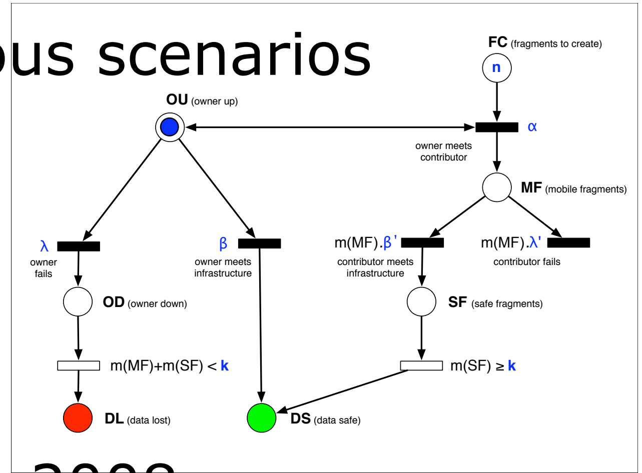
- Emulation as a complement to simulation
  - Real prototype of a complex software stack
    - Runs on real hardware, i.e. uses real device drivers, communication stacks, etc.
    - Embeds real middleware, real FT mechanisms
    - Is a full prototype of the application
  - No details can be simplified
  - Fault-injection
- Use of real mobility/usage data for
  - Realistic mobility/failure/usage models
  - Realistic analytical evaluation parameters

Emulation platform

Mobility models

ARUM : Mobility/Fault Models in UbiSys

- For resilience evaluation in ubiquitous scenarios
  - Mobility/Connectivity models
  - Fault models
  - Energy consumption models
- The ARUM LAAS Project
  - R. Diaz's PhD started september 2008
  - Collect real data on different pops
    - Humans using GPS Smartphones
    - Animals using GPS collars
  - Build realistic models
    - for simulation, emulation, prototyping, etc



ARUM Emulation Platform - Scale issue

- Ubiquitous systems can be very diverse
  - VANETs
  - Urban social networking
  - Nano-robots
- Can we have a unified approach
  - Scale, scale, scale

Experimental platform

- Typically composed of fixed and mobile devices
  - Programmable mobile platform
  - Light processing unit
  - Wireless comm. interfaces (adhoc+infra)
  - Positioning device
- Dedicated laboratory
  - Infrastructure
    - Computing, communication, positioning
  - Reproducible experiments
  - Fault injection

Scalable lab !

- Different prototypes ➔ different scales
  - Communicating vehicles (VANETs)
    - Device: 3m
    - Environment: 1km road
    - Communication range: 100s m
  - Cooperating nano-robots
    - Device: ∅1μm
    - Environment: ∅20μm vessel
    - Communication range: 10s μm
- Scale increase or decrease
  - Factor from 10^{-6} to 10^{2}

} 50

} 10^{-6}

Distributed BlackBox Example

- Exemplifies the emulation platform
  - critical application
  - mobile devices (C2CC)
  - distributed protocol (P2P)
- Collects and saves car/env data
  - Status of dr.wheel/pedals
  - Speed, orientation, position
  - Surrounding vehicles
  - etc.
- Not hardened hardware but distributed algorithm

HIDENETS Highly DEpendable IP-based NETworks and Services

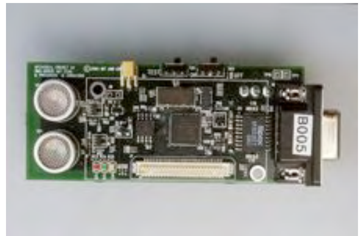
Information Society Technologies

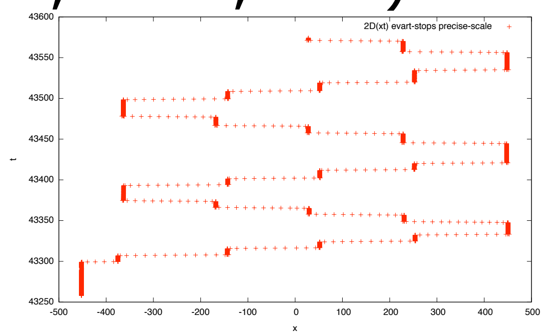
What are the issues to solve

- Technical issues
  - Precise indoor positioning
  - Programmable mobility
  - Range-controlled communication

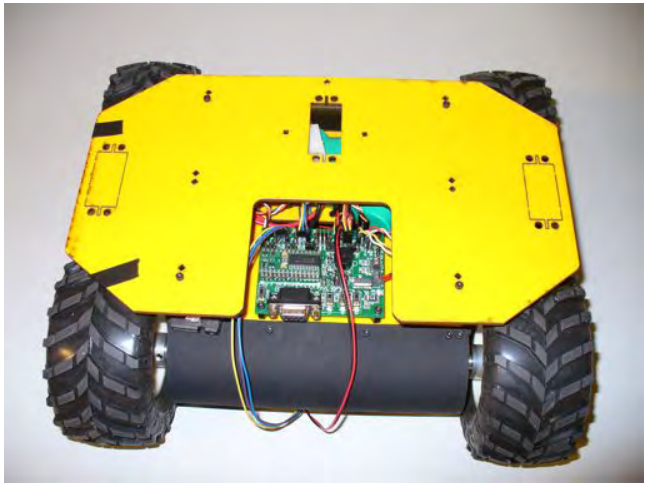
Precise Indoor Positioning

- Desired precision
  - In-vehicle GPS ≈ 5m
  - Scale reduction factor 50
    - Indoor precision ≈ 10cm
- Several technologies
  - Scene analysis (motion capture)
  - Triangularization (RF, ultra-sound, UWB, etc.)
- Evart positionning system
  - sub-mm precision @ 100 Hz





Programmable Mobility



- Reproducible Mobile experiments
  - Small robot platforms
    - Carry PDA or laptop
    - LynxMotion 4WD
- Different designs
  - Tape tracks
    - 20cm/s for a few hours
  - "*Autonomous*" version under development
    - Increased programmability/precision/speed
    - Use of LAAS robotic architecture

Programmable Mobility

- Reproducible Mobile experiments
  - Small robot platforms
    - Carry PDA or laptop
    - LynxMotion 4WD
- Different designs
  - Tape tracks
    - 20cm/s for a few hours
  - "*Autonomous*" version under development
    - Increased programmability/precision/speed
    - Use of LAAS robotic architecture

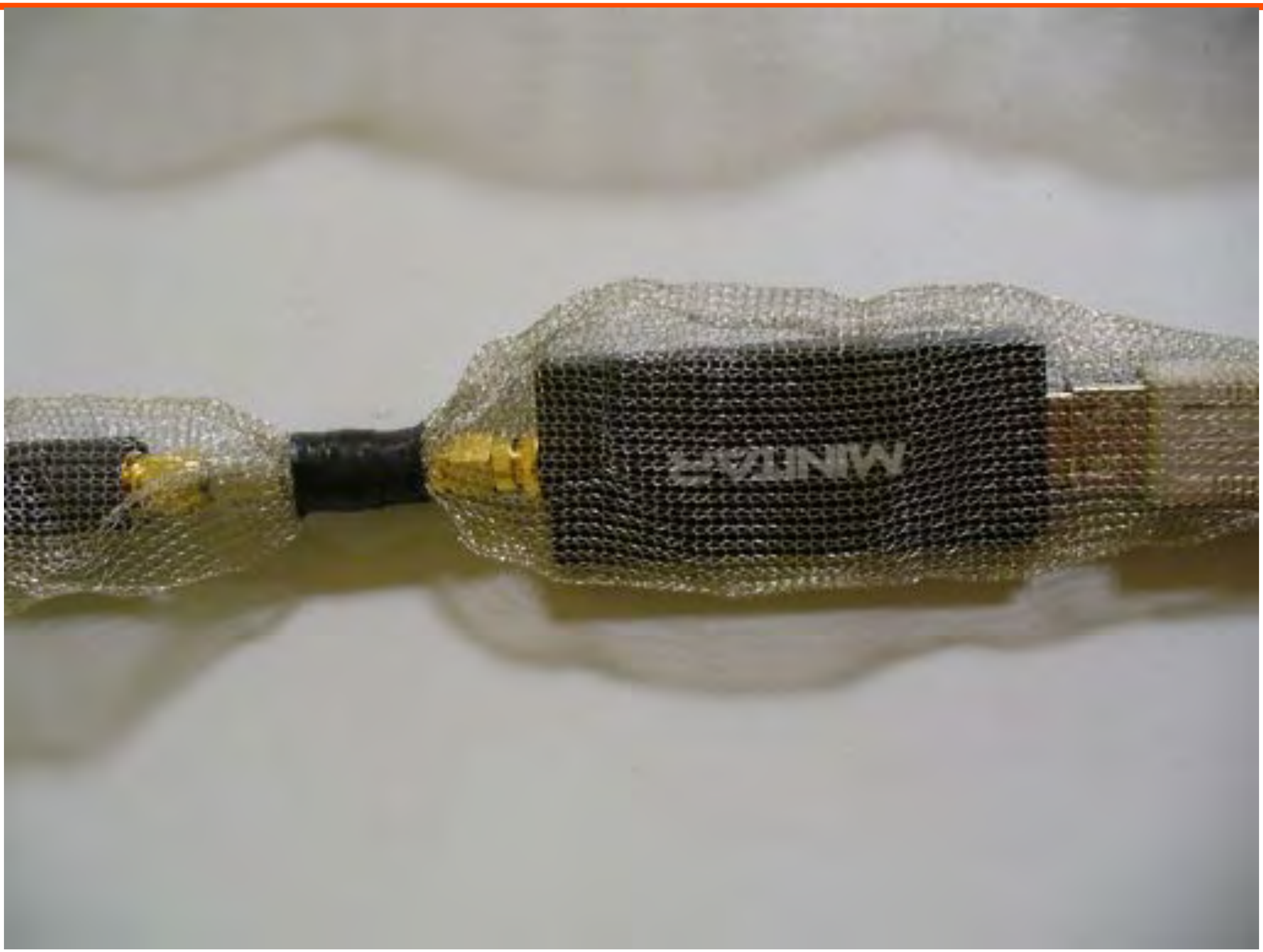
The BIG issue: wireless communication

- How to scale down WIFI ?
  - Hundreds meters ➔ few (2-3) meters
  - Precisely and controlled
- Potential solutions
  - Via emulation
  - Reducing Tx power of adapters
    - Access driver
    - Faraday cages, tents, etc.
    - Signal attenuators

The BIG issue: wireless communication

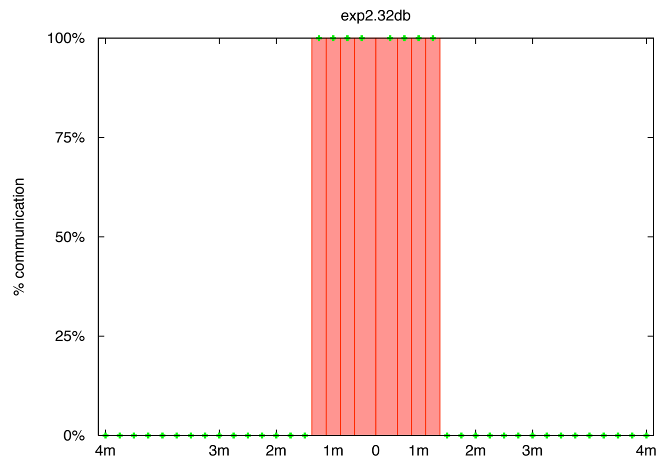
The BIG issue: wireless communication

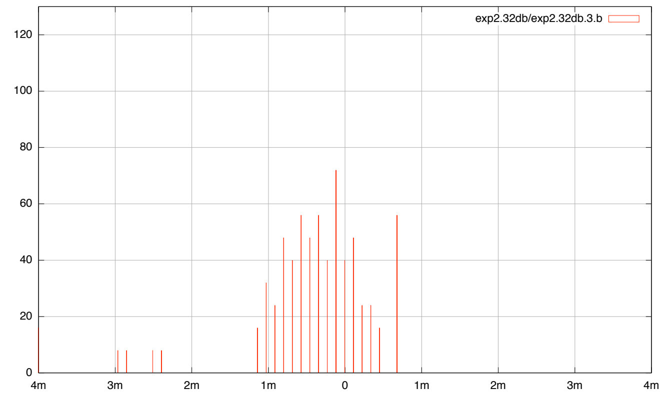
-
-



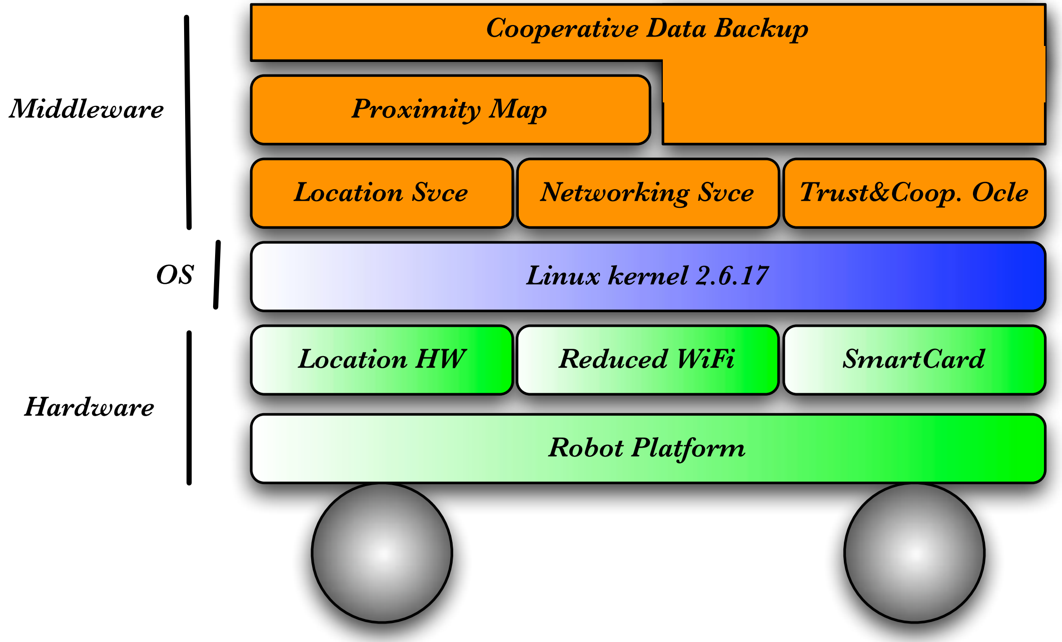
Scaled WiFi : Results

- Final experiments
- Static and mobile objects
- C2C: 1.5m -> 32db

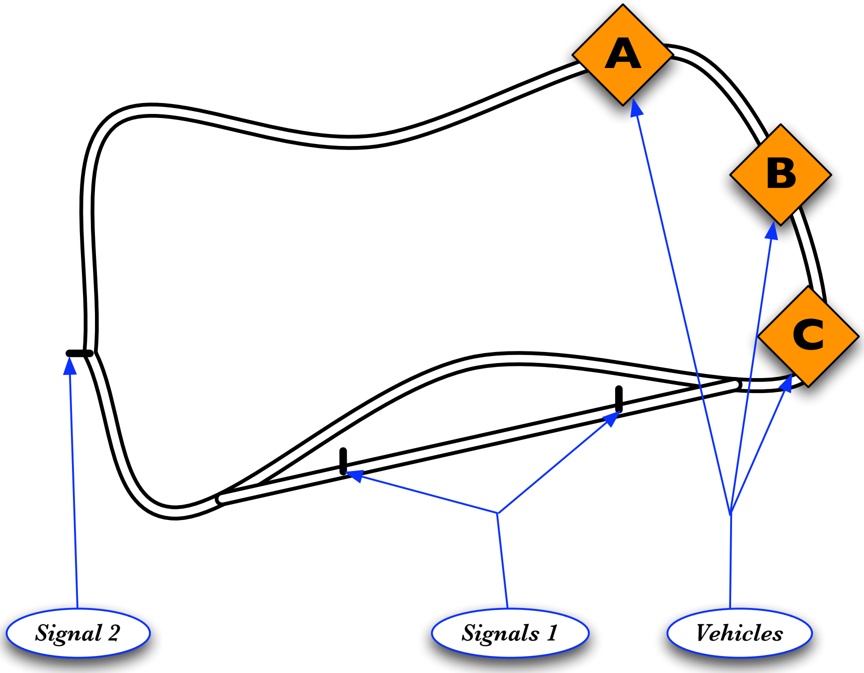




LAAS Hidenets Platform



Scenario



A B signal 1 = accelerate/decelerate
signal 2 = change mode: a-next turn left
b-next turn right

C go a little faster, when obstacle = go straight forward and crash

Future Work

- Experimental evaluation
  - Complete the evaluation prototype
  - Extend the mobility aspect
  - Develop fault-injection techniques
  - Develop benchmarks
- Analytical evaluation and mobility models
  - Develop a data collection platform
  - Extract and anonymize data
  - Synthesize models from data
  - Evaluate models

Conclusion

- Ambiant Intelligence/Ubiquitous/Internet of Things
  - Tremendous need for user trust/confidence
  - Resilience (Dependability + Evolution)
- Evaluation of Resilient UbiMob Systems
  - Large open field
    - Evolvability
    - Privacy
  - ARUM Contribution
    - Analytical evaluation
    - Experimental evaluation