Software Challenges and Solutions for Ad Hoc Networks

Thomas Gross

Departement Informatik
Laboratory for Software Technology
ETH Zürich
CH 8092 Zürich
Outline

• Ad hoc networks
• Software challenges
• How to address (some) challenges
  – Structures (research community)
  – Flexibility (software system design)
• Concluding remarks
Ad hoc networks

• Basic idea: network without “managed” infrastructure
  – An idea as old as networks
  – Infrastructure can break
Artist’s rendition
Ad hoc networks (2)

• Managed infrastructure has its advantages
  – A Jazz ensemble may get by without a conductor
  – But the audience still has to agree to be quiet.

• Ad hoc networks on the “fringe” of the wire-based (managed) infrastructure
  – Co-exist
Artist’s view (2)
Some problems

- Design of antennas and air interfaces
- Coordination of devices
- Protocol services
- Charging and fairness
- Security
- Privacy and public safety
- Application properties
Where is there software?

• Design of antennas and air interfaces
  – Software radios
• Coordination of devices
  – Distributed coordination
• Protocol services
  – Routing
• Charging and fairness
• Security
• Privacy and public safety
• Application properties
Ad hoc networks (3)

• Some issues/questions:
  – What are the “nodes”?
    • Capabilities
    • Latencies
    • Programming model
    • Physical constraints
    • Usage patterns
    • Government regulations and user perceptions
  – Launch into established, high-barrier-to-entry markets not a good idea
    • Even a sick 800 pound gorilla not easily moved
    • Plus: if the gorilla got sick, why do you want its cell?
• What applications are enabled by ad hoc networks?
Two endpoints

• Sensor nets
  – A hot topic
  – Environmental monitoring more important

• Application in self-organizing environments
  – Information systems as a model for applications
  – Large body of practical experience with heterogeneous environments and networks
Fighting isolationism

- The limitations of the “individual investigator” model make progress on cross-area issues difficult.
  - Combine constraints of multiple layers/areas
- Need to form teams
  - Done on many US institutions
  - Old mode of operation for mission-oriented funding agencies
  - Not usually done in universities in the Humboldt tradition
    - A “chair” covers an area/topic
  - (Virtual) research networks offer an alternative
    - Want to include all players, regardless of location
MICS: Mobile Information and Communication Systems

• A project funded by the Swiss National Science Foundation (SNF)
  – NCCR: National Center of Competence in Research
• NCCR: Swiss incarnation of a US NSF “Engineering Research Center“ (ERC)
What is an NCCR?

• Goals (of the Swiss National Science Foundation):
  – "Promotion of scientific excellence in areas of major strategic importance for Switzerland"
  – Re-shaping of the Swiss academic landscape, by getting the institutions to define priorities and to network with one another

• Currently 14 NCCRs
  – life sciences (5)
  – physics (3)
  – information technologies (3)
  – sustainable development and environment (2), social and human sciences (1)
What is an NCCR?

- Project horizon 10 years, funding allocated for 4 years (Nov 2005)
  - Organized as “network of individual projects (IP)“
- MICS budget: CHF 32M (US$ 20M) over 4 years
  - 50% SNF, 50% matching by home institution
- MICS size: about 30 faculty members and 60-70 PhD students
- MICS operation: officially started on Nov 1st 2001
  - For most groups: now in operation for one year
What is an NCCR?

• Goals (of the researchers):
  – Have fun.
  – Engage in good research. Write papers.
  – Pay doctoral students.
  – ..... (some are probably not disclosed)
  – Work with other experts.
MICS

- MICS organized into
  - Leading house (EPF Lausanne)
    - Director M. Vetterli
    - Administration and accounting
  - Management team (committee)
    - Representatives of IPs
    - Director/Deputy Director
  - 11 individual projects (“IP”)
    - Researchers from various Swiss institutions
NCCR MICS (self) organization

- Mathematics of self-organized communications, P. Thiran (EPFL)
- Information theoretic issues, E. Telatar (EPFL)
- Physical layer and software radio testbed, B. Rimoldi (EPFL)
- Self-organizing networking mechanisms, J.-P. Hubaux (EPFL)
- Self-organized distributed applications in a mobile environment, K. Aberer (EPFL)
- Security and cryptographic issues, S. Vaudenay (EPFL)
- Distributed signal processing and communication, M. Vetterli (EPFL)
- System and software architecture, T. Gross (ETHZ)
- Communicating embedded systems, L. Thiele (ETHZ)
- Terminodes, wireless e-business models and scenario planning, Y. Pigneur (UNIL)
- Wireless sensor networks, C. Enz (CSEM)
Projects are distributed

ETH Zürich: Departments of Computer Science and Information Technology and Electrical Engineering

CSEM, Swiss Center for Electronics and Microtechnology

University Zurich: Institute of Computer Science

University Berne: Institute of Computer Science and Applied Mathematics

EPFL: Schools of Computer and Communication Sciences (Leading House), Basic Sciences and Engineering

University Lausanne
Ecole des Hautes Etudes Commerciales

University St Gallen
mcm Institute
# Working across layers

<table>
<thead>
<tr>
<th>Theoretical foundations</th>
<th>Communication layers</th>
<th>Systems &amp; Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mathematical foundation (IP1)</td>
<td>Economics (IP10)</td>
<td>System architecture (IP8)</td>
</tr>
<tr>
<td>Information theory (IP2)</td>
<td>Real-time services (IP7)</td>
<td>Communicating embedded systems (IP9)</td>
</tr>
<tr>
<td>Security (IP6)</td>
<td>Information systems (IP5)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Network layer (IP4)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Physical layer and MAC (IP3)</td>
<td></td>
</tr>
</tbody>
</table>

## Explanation

The diagram illustrates the interconnectivity of various domains in communication systems and information technology, aiming to show how different layers and systems interact with each other. Here are some key aspects:

- **Mathematical foundation (IP1)**: Provides the theoretical basis for understanding and developing complex systems.
- **Information theory (IP2)**: Deals with the quantification of information and its transmission.
- **Security (IP6)**: Ensures the confidentiality, integrity, and availability of data.
- **Network layer (IP4)**: Facilitates the transmission of data across networks.
- **Physical layer and MAC (IP3)**: Handles the physical aspect of data transmission and the Medium Access Control layer.
- **Real-time services (IP7)**: Focuses on services that require immediate or near-immediate response times.
- **Information systems (IP5)**: Involves the design, development, and application of computer systems.
- **Economics (IP10)**: Examines the economic aspects of communication systems.
- **System architecture (IP8)**: Describes the overall structure of a system, including its components and their interactions.
- **Communicating embedded systems (IP9)**: Concerns systems that incorporate communication capabilities.
- **Sensor networks (IP11)**: Relates to networks where devices are deployed to gather data and communicate it to a central location.

This chart emphasizes the holistic approach to designing and managing communication systems, highlighting the interdependencies between different layers and disciplines.
Other activities

• Prototype systems
  – Sensor networks -- what sensors?
    • Environmental monitoring
    • Intelligent buildings

• Industrial liaison program

• Summer intern program for undergraduates

• Community building for women researchers

• Doctoral summer school
  – If you [or your students] are interested ...

www.nccr-mics.ch
Cross-layer and cross-cultural collaboration

- Connectivity in large-scale networks
- Mobility and routing
- Adaptive software systems
Connectivity of large-scale networks

- **P. Thiran, O. Dousse, EPFL, IP1**
- **Phase transition phenomenon**
  - Below a critical power $r_c \rightarrow$ disconnected network
  - Above critical power $r_c \rightarrow$ rapidly increasing probability $p$ of connected network
Connectivity of large-scale networks

Why would you care?

Zurich area

Surselva Alpine valley

Source: Geographical Information Service (Geostat), Swiss Federal Statistical Office
Self-organized routing based on mobility diffusion

Example due to M. Grossglauser, EPFL

Node mobility both challenge and opportunity:

• Past work has exploited mobility through the channel fluctuation it generates -> diversity

• Exploit “information carrying capacity” of mobile nodes

EASE: Exponential Age Search
Mobility

- Complicates matters
- Helps in other cases
- Problematic for software world -- changing environment/network properties
Aspect-oriented system design

• Key issue in design of a system: Separation of concerns [Dijkstra]
  – Concern --> aspect

• Why?
  – Evolution of system
  – Coherence of design

• How can it be realized?
  – Modularity
  – Components
  – Object-orientation
  – Aspect-oriented programming
Aspect-oriented programming

• “Untangle your code into cross-cutting, loosely coupled aspects” [Xerox AOP motto]
• Important early systems by Xerox PARC and IBM Research Lab
  – Aspect/J
  – Hyper/J
• Based on ideas developed by
  – Karl Lieberherr [NorthEastern]
  – Oscar Nierstrasz [Geneva, now Berne]

  – and many others ...
Aspect oriented programming

- System = application logic + advice(s)
- Advice
  - Code to deal with one issue (aspect)
- Combine advices with application logic
  - Join point: place where an advice can be invoked
  - This process is called weaving
Example

FTP printer spooler
- CancelJob
- PrintPS

Serial printer
- CancelJob
- PrintPS

- access control
- appl. logic
- accounting
- appl. logic
Example

1 class ExampleAspect extends Aspect {
2   Crosscut doActl = new FunctionalCrosscut() {
3       public void ANYMETHOD(ANY anyThis, REST rst) {
4           // access control code
5           { setSpecializer(
6               (MethodS.named(".*cancel.*")).AND
7               (MethodS.BEFORE)); }           }
8   };
9   Crosscut doAccnt = new FunctionalCrosscut() {
10      public void print(ANY anyThis, byte[] b) {
11           // accounting code
12           { setSpecializer(
13               (MethodS.AFTER).AND
14               (ClasseS.extending(Printer.class))); }           }
15   };
16 }
Aspect-oriented programming

• A number of systems have been designed
• Many variations
  – Use of “base” language
    • Application logic
  – Use of “advice” language
    • Special-purpose language
    • Base language
  – Specification of “join points”
  – Time of weaving
  – ...
Mobility requires adaptation

Ad hoc networks create a highly varying computing context:
- neighbors appear and disappear, e.g., in peer 2 peer settings
- policies (e.g., security rules) change over time or depending on location

- Increased mobility → new locations → new contexts
Adaptation in mobile setting

• Setting: robots that work in an intelligent factory
• Robots move (goods) through the assembly halls H₁, H₂, H₃, ....
  – Local (resident) robots modify goods
Adaptation in mobile setting

- Quality problems detected in production hall $H_2$
- $M$ enters $H_2 \Rightarrow$ it receives a run-time extension $e$ from $H_2$
- $e$ adds *at run-time*, on-the fly, functionality for
  - Monitoring all motor moves, sensor reads, incoming and outgoing messages
  - Records this information in an $H_2$–specific database
Aspect-oriented programming

• AOP provides an approach to modify software
  – Apply this idea to implement adaptation
• “Dynamic AOP”
  – AOP “on the fly”
• Many issues
  – Security
  – Performance
  – Programming model
PROSE - dynamic AOP

- PROSE - PROgrammable extenSions for sErvices
- Aspect-oriented programming for mobile systems
  - Base language: Java
- Joint work with A. Popovici and G. Alonso
  - Special thanks to A. Popovici for allowing me to use material from his (upcoming) thesis
Managed networks

- Long-running service application
- Costly to shut down or re-deploy
- Issue: How to apply fixes, monitoring
  - site-specific policies at *unexpected locations* in the code?
Problem and solution space

• Adapt:
  – multiple computing nodes
  – at multiple points in their execution
  – at various points in time (hot fixes) and space (mobility)

• Responsibility of adaptation: in the computing context

• Divide problem in sub-problems
  – infrastructure for adapting individual nodes
  – infrastructure for adapting entire node communities
AOP System architecture

Aspects

Join-point generator

Weaver

JVM

AOP Engine

Execution Monitor

Join point

Configuration API

Callback API

Join Point

Execute Advice

Filter Join Point Event

Find Advice

Generate Join Point hooks

forall hooks

activate Join Point
Event-based model

- Simple and general model
- Can be used by multiple AOP engines
- Issue: implementation cost
Execution monitor

• Implementation
  – Jikes RVM, v. 2.0.2
  – Baseline JIT compiler
  – 1600 lines of code added to the VM core
  – Changes affect several JVM modules
    (Garbage Collector)
Performance -- normal execution (R2)

- Performance penalty: < 10%
- Benchmark: SpecJVM 98
Security

Weaving can be secure
AOP operations can be easily made transactional
Other AOP Engines can be built on top of the same Execution Monitor
Ad hoc endpoints

• Individual nodes
  – AOP with PROSE
• “Information systems” applications
  – ... next ...
“Spontaneous containers”

- The networked environment acts like a container
- The applications interact dynamically
- Extensions use dynamic AOP to express adaptations
Spontaneous containers

- Allow new kinds of applications
- Ad hoc and spontaneous networking
WiFi, 4G, and the telcos’ debt

• Software solutions offer portability
  – Only a few [large] systems are used exclusively for the design tasks
    • Profits and key benefits sometimes elsewhere
    • Internet: NSW (National Software Works) -- email
  – Some systems good for nothing

• High-bandwidth WLANs attractive
  – People willing to pay

• G4 spectrum looking for a use, telcos looking for a way to recoup outlays
  – Mobile users create “ad hoc” networks
Challenges

• How to design, implement, test, and evolve adaptive systems

• Nobody (very few ...) an expert on all sub-areas of ad hoc networks
  – Layers a necessity and a problem
  – How to get (academic) researchers to self-organize

• How do we educate the next generation of researchers
  – If possible, can we maybe also create a few jobs? (or at least preserve a few ones?)
Thanks

- G. Alonso
- J. Bovay
- O. Dousse
- M. Grossglauser
- A. Popovici
- P. Thiran
- M. Vetterli

Swiss National Science Foundation
EPFL
ETH Zürich
Geographical Information Service (Geostat)
Swiss Federal Statistical Office

all participants in NCCR MICS
Concluding remarks

• Software crucial if we want to support applications on ad hoc networks
  – ... is that different from other networks?
• New constraints and requirements
  • Software systems are difficult to design and implement
  – Must deal with need to adapt
  – Aspect-oriented system design an interesting model
  • Resource demands and prediction still an issue
• Many good, hard questions remain.
Thank you for your attention.