DATA 61

Modelling and Verification of Protocols for Wireless Networks

(Lecture 1)

Peter Höfner

(Lecture at University of Twente, Jan/Feb 2017)

www.data61.csiro.au

UNSW AUSTRALIA

CSIRO

last update: Jan 22, 2017

Lecturers

Dr. Peter Höfner
Data61, CSIRO and UNSW, Australia
email: peter.hoefner@data61.csiro.au

Dr. Ansgar Fehnker
University of Twente
email: ansgar.fehnker@utwente.nl

Timetable and Plan (1)

Time	*Location*	*Type*	*Topics*
17/01, 10:45-12:30	*CR2L*	*lecture*	*introduction, modelling: process algebra AWN (syntax)*
17/01, 15:45-17:30	*CR2N*	*lab*	*modelling*
19/01, 10:45-12:30	*HB2D*	*lecture*	*modelling: process algebra AWN (semantics)*
19/01, 13:45-14:30	*CR2N*	*tutorial*	
23/01, 13:45-15:30	*HB2D*	*lecture*	*modelling: timed automata*
23/01, 15:45-17:30	*HB2D*	*lab*	*modelling*
26/01, 13:45-15:30	*HB2D*	*lecture*	*comparison: AWN vs TAs*
26/01, 15:45-16:30	*HB2D*	*tutorial*	*Uppaal*

Consultation Time: Wednesdays 13:30-14:30 (or via appointment); room: ZI-3063

Timetable and Plan (2)

Time	*Location*	*Type*	*Topics*
01/02, 13:45-15:30	*CR2N*	*lecture*	*verification(1): query language and local properties*
01/02, 13:45-17:30	*CR2N*	*lab*	*invariants & verification*
02/02, 13:45-15:30	*CR2N*	*lecture*	*verification(2): global properties, case study AODV*
02/02, 15:45-16:30	*CR2N*	*lab*	*verification (use of Uppaal, Isabelle or pen and paper)*
06/02, 2 hours		*lecture*	*open problems, Q&A*
07/02, 4 hours		*oral exam*	*individual exams (~30 minutes)*
*09/02, 2 hours**		*lab*	*verification/setting up individual projects*
Weeks 5-7			*individual projects*

* taught by Ansgar Fehnker

Administration

- Do I have to come to each and every lecture?
- Are the dates suitable for you?
- Passing the course
  - submit lab exercises
  - participate actively in tutorials
  - oral exam in week 4
  - (individual project)

- Other administration issues?

Preliminaries

- (predicate) logics
- finite automata / finite state machine

- have you heard about the following
  - process algebra
  - timed automata
  - model checking
  - interactive theorem proving (Isabelle/HOL)

Modelling and Verification of Protocols for Wireless Networks

www.data61.csiro.au

UNSW AUSTRALIA

CSIRO

Contents of this Lecture

What should you have learnt

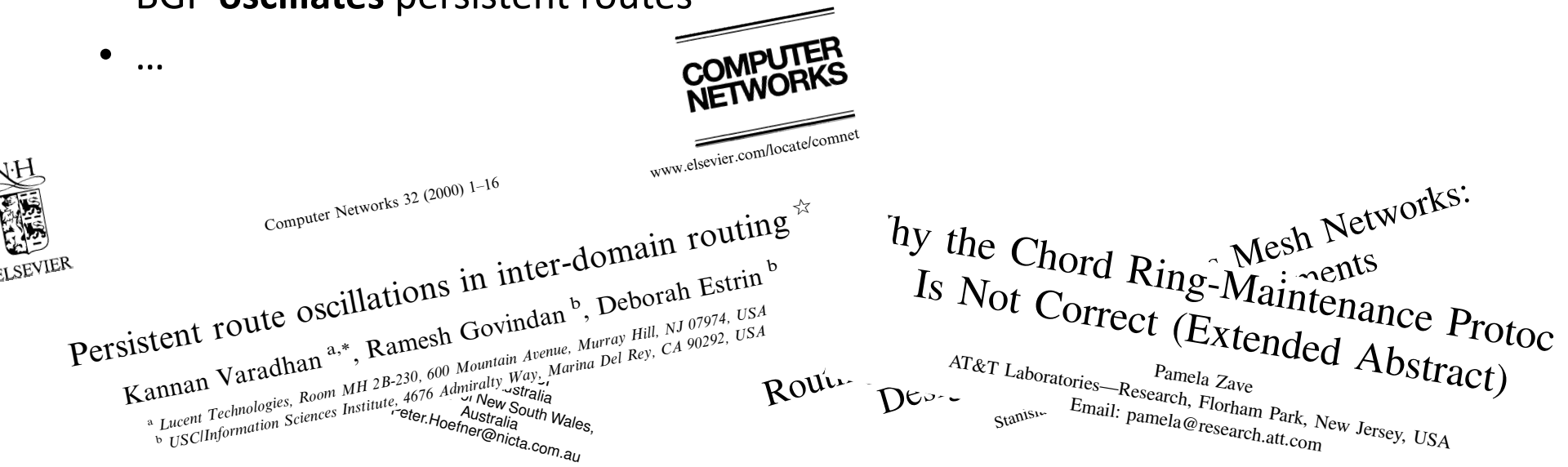
- Introduction
  - why formal modelling
  - why formal reasoning
  - problems of state of the art

- Process Algebra AWN
  - intuition
  - syntax
  - examples

Introduction

Why Formal Modeling and Analysis

- Routing Protocols are Broken
  - Routing Protocols establish **non-optimal routes**
  - AODV Routing Protocol sends packets in **loops**
  - Chord Protocol is **not correct**
  - BGP **oscillates** persistent routes
  - …



Today's Protocol Development

- IETF: "Rough Consensus and Running Code" (Trial and Error)
  - start with a good idea
  - build a protocol out of it (implementation)
    - run tests (over several years)
    - find limitations, flaws, etc…
    - fix problems
  - build a new version of the protocol
  - at some point people agree on an RFC (request for comments)

Beauvais Cathedral
(~300 years to build, at least 2 collapses)

Better Protocols are Needed Now!

- We cannot afford this approach
  - to expensive w.r.t. time
  - to expensive w.r.t. money
  - we are not working in a lab, i.e., sometimes we have one try only (e.g. BGP)
- Is there a method which is more reliable and cost efficient

The original design was so boldly conceived that it was found structurally impossible to build.

What's the Problem? (1)

- Specifications are (excessively) long
  - the Session Initiation Protocol is 268 pages long (and not even self contained - by 2009 142 additional documents were required)
  - IEEE 802.11 is 2.793 pages long

What's the Problem? (2)

- Specifications are
  - underspecified
  - contradictory
  - erroneous, and
  - ambiguous

What's the Problem? (3)

- Specifications are written in English Prose
  - in case of AODV there are 5 different implementations, all compliant to the standard

What's the Problem? (3)

- Specifications are written in English Prose
  - in case of AODV there are 5 different implementations, all compliant to the standard

Aims

- Provide complete and practical formal methods
  - expressive
    (mobility, dynamic topology, types of communication,…)
  - usable and intuitive
  - description language + proof methodology + automation

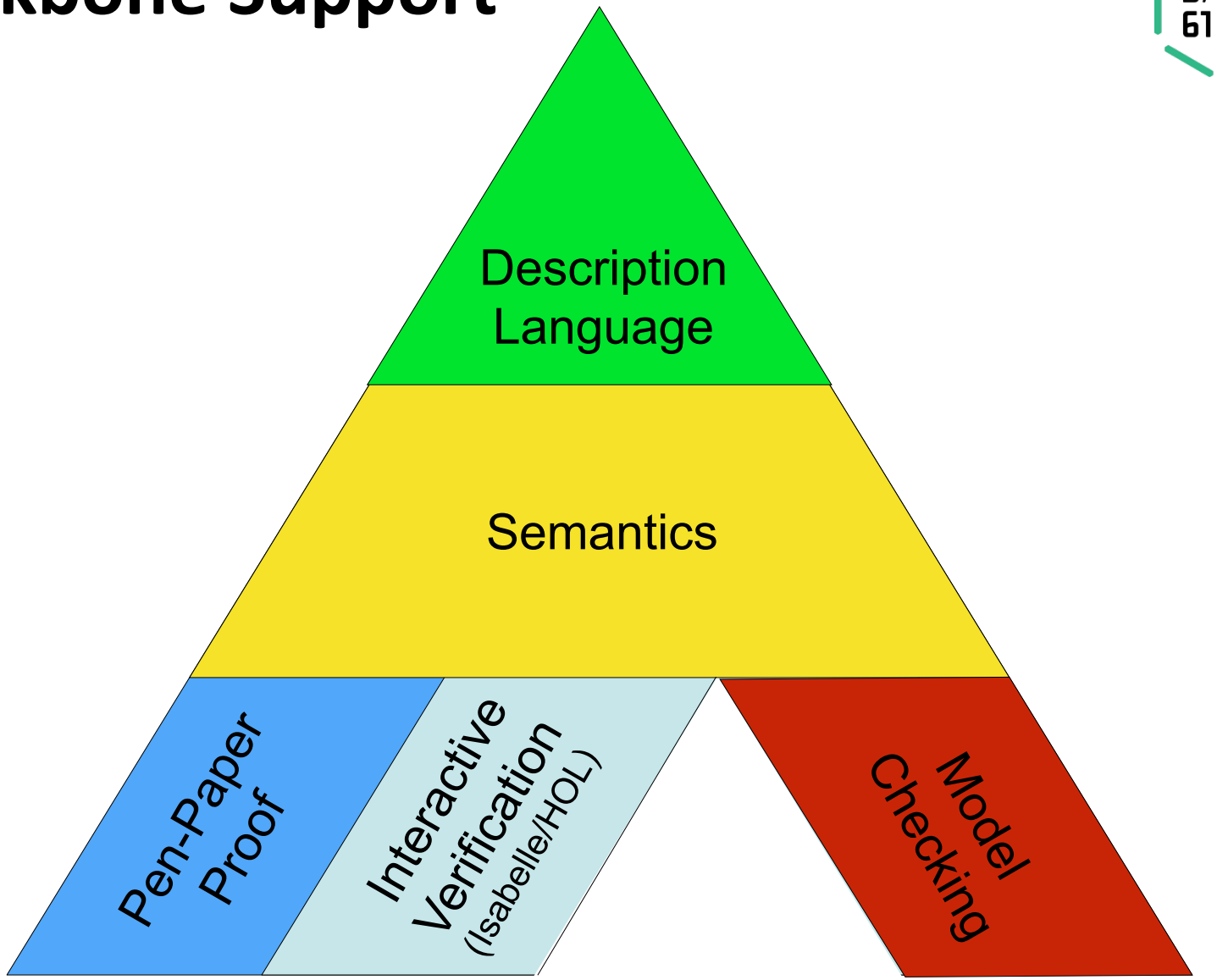
- Specification, verification and analysis of protocols
  - formalise relevant standard protocols
  - analyse the protocols w.r.t. key requirements
  - analyse compliant implementations

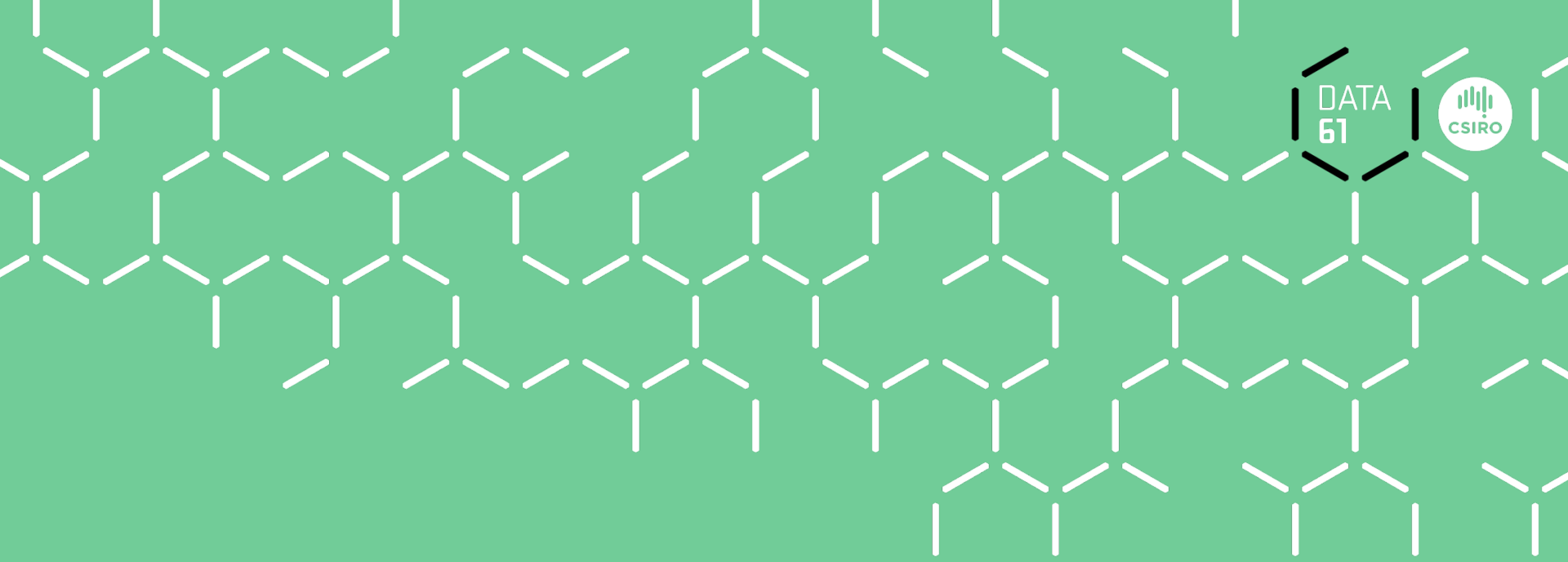
- Development of improved protocols
  - assured protocol correctness
  - improve reliability and performance

Benefits

- Benefits
  - finding and fixing bugs
  - improve reliability and performance
  - proving correctness
  - reduce “time-to-market”

Backbone Support

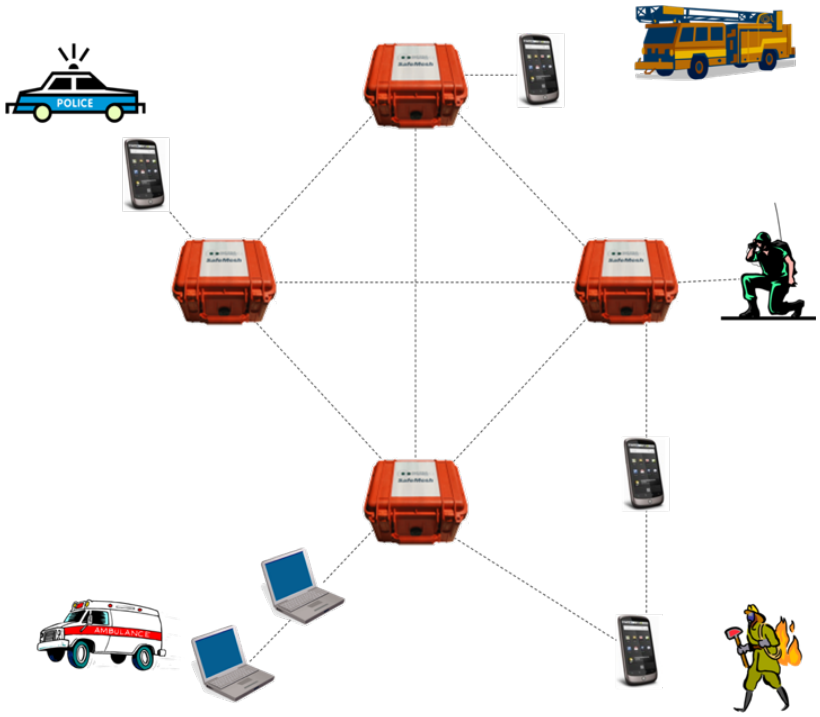




The Process Algebra AWN (Algebra for Wireless Networks)

A Simple Network

- Wireless Network



Features

- Which features should a suitable formal method offer
  - sequential programs on nodes
    - update
    - (non)-deterministic choice
    - guards (if-constructs)
    - loops
    - data + data update
    - sequential composition
    - (function calls)

Features

- Which features should a suitable formal method offer
  - parallel programs on same node
    - synchronisation (send/receive)
  - interaction between different nodes
    - synchronisation (uni-, group-, broadcast)
  - network (topology)
    - how to model links
    - topology changes

Developed Process Algebra

- Description Language (Syntax)

$X(exp_1, \ldots, \exp_n)$	process calls
$P + Q$	nondeterministic
$[\varphi]P$	if-construct (guard)
$[\![\mathtt{var} := exp]\!]P$	assignment followed
$\mathbf{broadcast}(ms).P$	broadcast
$\mathbf{groupcast}(dests, ms).P$	groupcast
$\mathbf{unicast}(dest, ms).P \blacktriangleright Q$	unicast
$\mathbf{send}(ms).P$	send
$\mathbf{receive}(\mathtt{msg}).P$	receive
$\mathbf{deliver}(data).P$	deliver

Developed Process Algebra

- Description Language (Syntax)

$[\varphi]P + [\neg\varphi]Q$	deterministic choice
$P(n) = [\![n := n+1]\!].P(n)$	loops

- Do we need more?

A Simple Example

- Can you describe a simple flooding protocol.
  - informal description:
    - every node has a unique identifier (IP address) and a message (let's say a number) to distribute
    - a node can send its message (together with its IP) at any time
    - if a node receive a message it stores the contents if the message was not handled previously, the message is forwarded to all nodes within transmission range

Flooding

specification follows roughly
https://tools.ietf.org/html/draft-ietf-manet-bcast-00

Process 1 Flooding

```
FLOOD(ip, m, b, store) =def
 1. (receive(ms) .
 2.    /* check message format and distill contents */
 3.    [ ms = msg(ip′, m′) ]
 4.       (
 5.          [ store(ip′) = m′ ]        /* message handled before */
 6.             FLOOD(ip,m,b,store)
 7.          + [ store(ip′) ≠ m′ ]       /* new message */
 8.             ⟦store(ip′) = m′⟧
 9.             broadcast(ms) .
10.             FLOOD(ip,m,b,store)
11.       ))
12. + [ b = false ]        /* message not yet send */
13.    broadcast(msg(ip, m)) . FLOOD(ip,m,true,store)
```

the data structure and the initial state should be straight forward

Timed Protocols

- many protocols depend on timing issues
  (e.g. repetitive tasks)
- the process algebra AWN can easily be extended by time,
  the syntax is extended by a simple data type `TIME`;
  every node maintains a clock/timer `now`

References

- P. Höfner: *Using Process Algebra to Design Better Protocols.* In The Role and Importance of Mathematics in Innovation. Mathematics for Industry 25:87–101, Springer, 2016. doi: 10.1007/978-981-10-0962-4_8
- A. Fehnker, R.J. van Glabbeek, P. Höfner, A. McIver, M. Portmann and W.L. Tan: *A Process Algebra for Wireless Mesh Networks.* In H. Seidl (ed.), Programming Languages and Systems (ESOP'12), Lecture Notes in Computer Science 7211, 295–315, Springer, 2012. doi: 10.1007/978-3-642-28869-2_15
- A. Fehnker, R.J. van Glabbeek, P. Höfner, M. Portmann, A. McIver and W.L. Tan: *A Process Algebra for Wireless Mesh Networks used for Modelling, Verifying and Analysing AODV.* Technical Report 5513, NICTA. 2013. arXiv: CoRR abs/1312.7645
- E. Bres, R.J. van Glabbeek, P. Höfner: *A Timed Process Algebra for Wireless Networks with an Application in Routing (Extended Abstract)*. In Programming Languages and Systems (ESOP'16). Lecture Notes in Computer Science 9632, 95-122, Springer, 2016. doi: 10.1007/978-3-662-49498-1_5