Past and Future of Formal Methods for Wireless Mesh Networks

Peter Höfner

(joint work with Ansgar Fehnker, Rob van Glabbeek, Annabelle McIver and others)

NII Shonan Meeting 17
November 9, 2012

Australian Government
Department of Broadband, Communications and the Digital Economy
Australian Research Council

NICTA Members
NICTA Partners

Past, Present and Future of Formal Methods for Wireless Mesh Networks

Peter Höfner

(joint work with Ansgar Fehnker, Rob van Glabbeek, Annabelle McIver and others)

NII Shonan Meeting 17
November 9, 2012

Australian Government
Department of Broadband, Communications and the Digital Economy
Australian Research Council

NICTA Members
NICTA Partners

Past and Present: Protocol Development

- "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
    - start testing again
  - at some point, people agree on an RFC (standard)

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

Future: Protocol Development

- Is there a method which is more reliable and cost-efficient?
- Is there a way to compare different protocols?
- **New methods required (or finetune/extend existing ones)**

“The original design was so boldly conveived that it was found structurally impossible to build.”

Problems

- Standards (IETF RFCs) are not precise
  - written in English
  - ambiguous (sometimes incomplete)
  - no formal specification
- Compliant implementations
  - have different behaviours
  - are not compatible
  - have serious flaws
- Traditional evaluation techniques: simulation and test-bed
  - expensive
  - limited to (a small number of) specific scenarios
  - error found after years of evaluation
  - barely offer any guarantee for properties such as route discovery

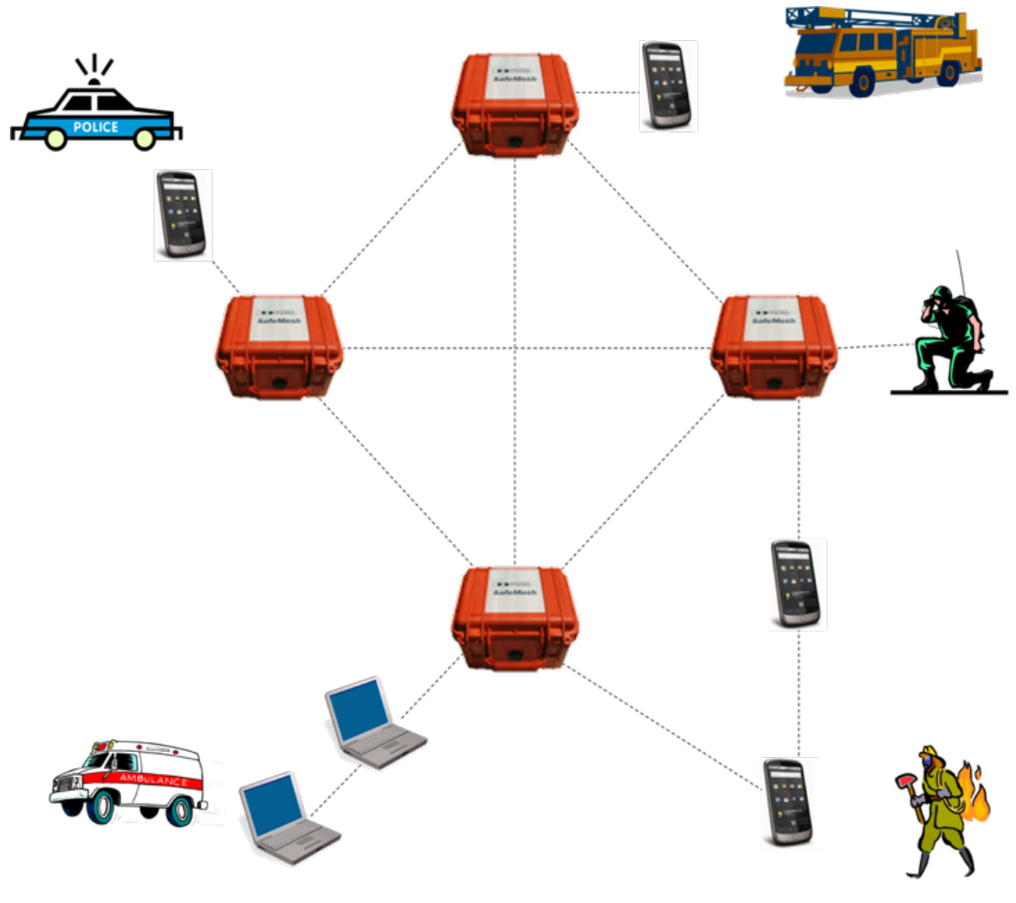
Formal Methods for Mesh Networks

NICTA

- Goal
  - model, analyse, verify and increase the performance of wireless mesh protocols
  - develop suitable formal methods techniques
- Benefits
  - more reliable protocols
  - finding and fixing bugs
  - better performance
  - proving correctness
  - reduce “time-to-market”

Wireless Mesh Networks

- Wireless Mesh Networks (WMNs)
  - key features: mobility, dynamic topology, wireless multihop backhaul
  - quick and low cost deployment
- Applications
  - public safety
  - emergency response, disaster recovery
  - transportation
  - counter-terrorism
  - smart grid
  - ...
- Limitations in reliability and performance



Main Ingredients for WMNs

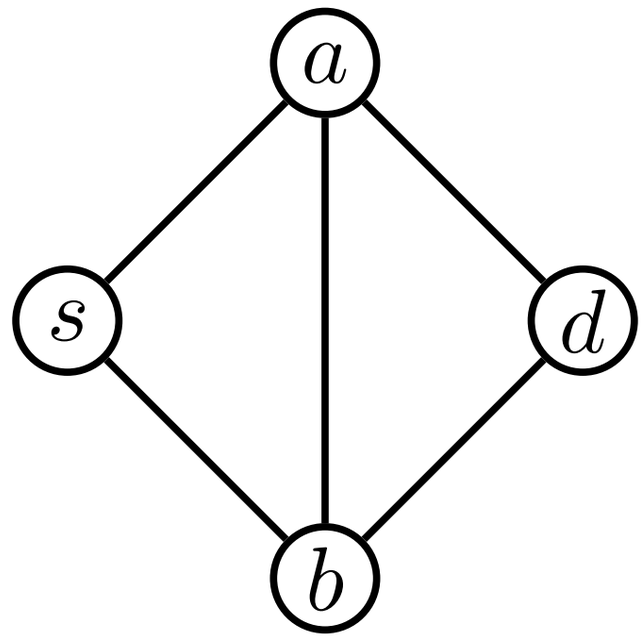
- Network with mobile nodes and dynamic topology
- Messages, which are sent through the network
  - route request (RREQ)
  - route reply (RREP)
  - route error (RERR)
  - ...
- Communication (message sending)
  - broadcast
  - unicast
  - groupcast (multicast/iterative unicast)
- Data
  - routing tables
  - node names

Case Study: AODV

- Ad Hoc On-Demand Distance Vector Protocol
  - Routing protocol for WMNs
  - Ad hoc (network is not static)
  - On-Demand (routes are established when needed)
  - Distance (metric is hop count)
  - Vector (routing table has the form of a vector)
  - Developed 1997-2001 by Perkins, Beldig-Royer and Das (University of Cincinnati)
  - One of the four protocols currently standardised by the IETF MANET working group

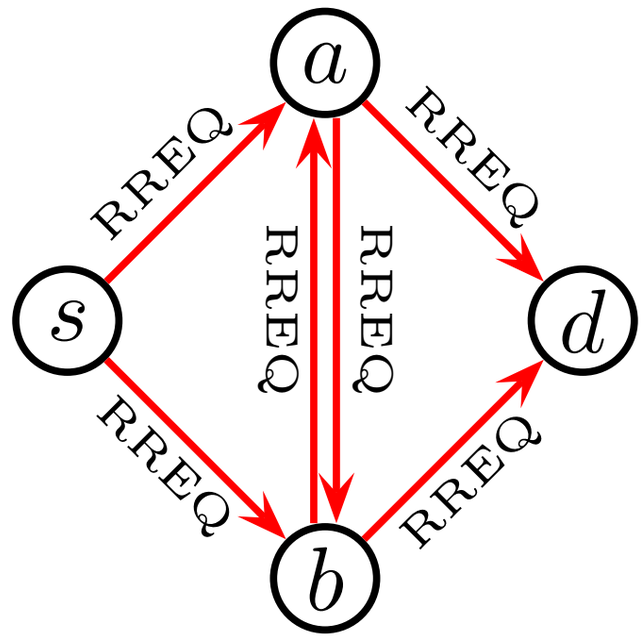
Case Study: AODV

- Main Mechanism
  - if route is needed
    BROADCAST RREQ
  - if node has information about a destination
    UNICAST RREP
  - if unicast fails or link break is detected
    GROUPCAST RERR



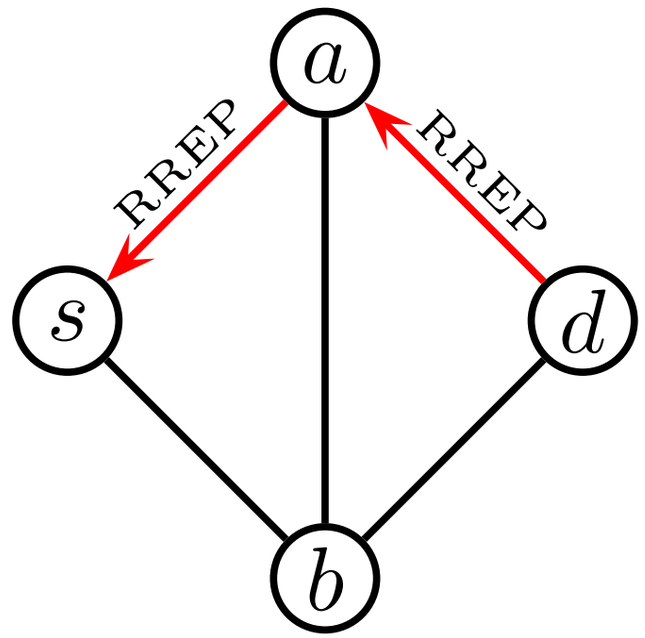
Case Study: AODV

- Main Mechanism
  - if route is needed
    BROADCAST RREQ
  - if node has information about a destination
    UNICAST RREP
  - if unicast fails or link break is detected
    GROUPCAST RERR



Case Study: AODV

- Main Mechanism
  - if route is needed
    BROADCAST RREQ
  - if node has information about a destination
    UNICAST RREP
  - if unicast fails or link break is detected
    GROUPCAST RERR



NICTA

Challenge 1: Formal Specification

unambiguous

concise

precise

Why Formal Specification?

Why Formal Specification?

Complete and Accurate Formalisation of AODV

NICTA

$+$ [$(\texttt{oip},\texttt{rreqid}) \notin \texttt{rreqs}$] /* the RREQ is new to this node */
⟦$\texttt{rt} := \texttt{update}(\texttt{rt},(\texttt{oip},\texttt{osn},\texttt{kno},\texttt{val},\texttt{hops}+1,\texttt{sip},\emptyset))$⟧ /* update the route to oip in rt */
⟦$\texttt{rreqs} := \texttt{rreqs} \cup \{(\texttt{oip},\texttt{rreqid})\}$⟧ /* update rreqs by adding (oip, rreqid) */
(
[$\texttt{dip} = \texttt{ip}$] /* this node is the destination node */
⟦$\texttt{sn} := \max(\texttt{sn},\texttt{dsn})$⟧ /* update the sqn of ip */
/* unicast a RREP towards oip of the RREQ */
unicast(nhop(rt,oip),rrep(0,dip,sn,oip,ip)) . AODV(ip,sn,rt,rreqs,store)
▶ /* If the transmission is unsuccessful, a RERR message is generated */
⟦$\texttt{dests} := \{(\texttt{rip},\texttt{inc}(\texttt{sqn}(\texttt{rt},\texttt{rip}))) \mid \texttt{rip} \in \texttt{vD}(\texttt{rt}) \wedge \texttt{nhop}(\texttt{rt},\texttt{rip}) = \texttt{nhop}(\texttt{rt},\texttt{oip})\}$⟧
⟦$\texttt{rt} := \texttt{invalidate}(\texttt{rt},\texttt{dests})$⟧
⟦$\texttt{store} := \texttt{setRRF}(\texttt{store},\texttt{dests})$⟧
⟦$\texttt{pre} := \bigcup\{\texttt{precs}(\texttt{rt},\texttt{rip}) \mid (\texttt{rip},*) \in \texttt{dests}\}$⟧
⟦$\texttt{dests} := \{(\texttt{rip},\texttt{rsn}) \mid (\texttt{rip},\texttt{rsn}) \in \texttt{dests} \wedge \texttt{precs}(\texttt{rt},\texttt{rip}) \neq \emptyset\}$⟧
groupcast(pre,rerr(dests,ip)) . AODV(ip,sn,rt,rreqs,store)
$+$ [$\texttt{dip} \neq \texttt{ip}$] /* this node is not the destination node */
(
[$\texttt{dip} \in \texttt{vD}(\texttt{rt}) \wedge \texttt{dsn} \leq \texttt{sqn}(\texttt{rt},\texttt{dip}) \wedge \texttt{sqnf}(\texttt{rt},\texttt{dip}) = \texttt{kno}$] /* valid route to dip that is fresh enough */
/* update rt by adding precursors */
⟦$\texttt{rt} := \texttt{addpreRT}(\texttt{rt},\texttt{dip},\{\texttt{sip}\})$⟧
⟦$\texttt{rt} := \texttt{addpreRT}(\texttt{rt},\texttt{oip},\{\texttt{nhop}(\texttt{rt},\texttt{dip})\})$⟧
/* unicast a RREP towards the oip of the RREQ */
unicast(nhop(rt,oip),rrep(dhops(rt,dip),dip,sqn(rt,dip),oip,ip)) .

Proposed Formal Method

- Based on Process Algebra AWN
  - inspired by π-calculus and LOTOS; based on ω-calculus
  - main process expressions

$X(exp_1, \ldots, exp_n)$	process calls
$P + Q$	nondeterministic choice
$[\varphi]P$	if-construct
$[\![\mathtt{var} := exp]\!]P$	assignment followed by P
$\mathbf{broadcast}(ms).P$	broadcast message followed by P
$\mathbf{unicast}(dest, ms).P \blacktriangleright Q$	unicast ms to $dest$; if successful proceed with P; otherwise with Q
$\mathbf{receive}(\mathtt{msg}).P$	receive message

Requirements for Formal Methods

- "Formal languages are useful tools for specifying parts of protocols. However, as of today, there exists no well-known language that is able to capture the full syntax and semantics of reasonably rich IETF protocols."

  [IETF]

- IETF Requirements (for formal methods)
  - relatively easy to extract code
  - complete specification
  - implementation independent

- Easy to use
  - only a few (well-known) programming constructs

NICTA

Challenge 2: Functional Correctness

non-quantitative (qualitative) properties

support by formal methods
(including ATP/ITP and MC systems)

Case Study: AODV

- Properties of AODV
  - route correctness
  - loop freedom
  - route discovery
  - packet delivery

Case Study: AODV

- Properties of AODV
  - route correctness 
  - loop freedom  (at least for some interpretations)
  - route discovery 
  - packet delivery 

Case Study: AODV

- Achievements
  - based on full concise specification of AODV (RFC 3561) (without time)
  - verified/disproved properties
    - route discovery
    - packet delivery
    - loop freedom
      - first (correct) proof
      - disproved loop freedom for variants of AODV (as implemented in at least 3 open source implementations)
  - found several ambiguities, mistakes, shortcomings
  - found solutions for some limitations

Used Technology

- Manual proofs based on process algebra (AWN)
  - allows reasoning for *all* possible scenarios (valid for all topologies)
- ITP: Isabelle/HOL
  - verify proofs
  - generate “proof-templates”
  - automate proofs for variants of the protocol (hopefully)
- Model Checking: Uppaal
  - find counter examples in small topologies (up to 7 nodes)
  - rigorous analysis by checking all 5-node topologies
- Statistical Model Checking: Uppaal
  - allows evaluation of much larger topologies so far we have examples with 100 nodes

NICTA

Challenge 3: Quantitative Analysis

quantitative properties

support by formal methods
(including ATP/ITP and MC systems)

Adding Time and Probabilities

- Timed-AWN seems (more or less) easy and straight forward
  - add time steps
  - allows modelling time-outs

- Probabilistic-AWN
  - Add message loss
    - unreliable links
    - sending interruptions ...

Quality of Protocols

- Quality Measurements used in Simulations
  - Packet Delivery Ration (PDR)
    <# packets delivered> / <# packets sent>
  - Overhead in Routing
    <# of control message> / <# data packets delivered>
  - Delay
    average time to deliver a packet

Problem / Challenge

Problem:
these measurements are situation dependent
(dependent on topology, timing,....)

Problems / Solutions

- Statistical Model Checking
  - topologies of reasonable size
  - which topologies should be chosen
  - what is the value of SMC against simulation-based techniques
- Model Checking
  - seems not reasonable
  - SMC seems more appropriate
- Manual Proofs/ITP systems
  - can we gain more
  - can we abstract from topologies
  - how far can we go without over-abstaction

Conclusion (Summary of Challenges)

NICTA

- Challenge 1: Formal Specification
  - achieved for many protocols (e.g., AODV, DYMO)
  - needs domain-knowledge
  - tools are around (AWN, MODEST, PAT, ...)

- Challenge 2: Formal Correctness
  - classic properties
  - can be (easily) achieved by today's tools

- Challenge 3: Quantitative Analysis
  - allows a more realistic protocols evaluation

Conclusion (Summary of Challenges)

NICTA

- Challenge 1: Formal Specification
  - achieved for many protocols (e.g., AODV, DYMO)
  - needs domain-knowledge
  - tools are around (AWN, MODEST, PAT, ...)

- Challenge 2: Formal Correctness
  - classic properties
  - can be (easily) achieved by today's tools

- Challenge 3: Quantitative Analysis
  - allows a more realistic protocols evaluation

Conclusion (Summary of Challenges)

- Challenge 1: Formal Specification
  - achieved for many protocols (e.g., AODV, DYMO)
  - needs domain-knowledge
  - tools are around (AWN, MODEST, PAT, ...)

- Challenge 2: Formal Correctness
  - classic properties
  - can be (easily) achieved by today's tools

- Challenge 3: Quantitative Analysis
  - allows a more realistic protocols evaluation

Conclusion (Summary of Challenges)

NICTA

- Challenge 1: Formal Specification
  - achieved for many protocols (e.g., AODV, DYMO)
  - needs domain-knowledge
  - tools are around (AWN, MODEST, PAT, ...)

- Challenge 2: Formal Correctness
  - classic properties
  - can be (easily) achieved by today's tools

- Challenge 3: Quantitative Analysis
  - allows a more realistic protocols evaluation

NICTA

From **imagination** to **impact**