Analysing mobile networks via probabilistic model checking

Marta Kwiatkowska
School of Computer Science
THE UNIVERSITY OF BIRMINGHAM

Joint work with

Marie Duflot, Gethin Norman,
Dave Parker, Jeremy Sproston

Background

- FGUC: Foundations of Global Ubiquitous Computing
  - EU activity & this workshop
- SGUC: Science for Global Ubiquitous Computing (GC2)
  - One of 7 UK Grand Challenges (GC2), related to FGUC
  - Rigorous foundation for tools and techniques
- Also GC4: Scalable Ubiquitous Computing Systems
  - Design, engineering, managing ubiquitous systems
  - Tools and techniques
- This talk, focus on a component of GC2
  - Mobile ad hoc network protocols
  - Probability: why needed, challenges
  - Verification techniques and tools

Ubiquitous computing: the trends…

- Devices, ever smaller
  - Laptops, phones, PDAs, …
  - Sensors, motes, …
- Networking, wireless, wired & global
  - Mobile ad hoc
  - Wireless everywhere
  - Internet everywhere
  - Global connectivity
- Systems/software
  - Decentralised
  - Self-organising
  - Self-configuring
  - Autonomous
  - Adaptive
  - Context-aware

Ubiquitous computing: users expect…



- …assurance of
  - safety
  - correctness
  - performance
  - reliability

- For example:
  - Is my e-savings account secure?
  - Can someone bluesnarf from my phone?
  - How fast is the communication from my PDA to printer?
  - Is my mobile phone energy efficient?
  - Is the operating system reliable?
  - Can the laptop recover from faults with no effort on my part?

Probability helps

- In distributed (de-centralised) co-ordination algorithms
  - As a symmetry breaker
    - "leader election is eventually resolved with probability 1"
  - In gossip-based routing and multicasting
    - "the message will be delivered to all nodes with high probability"
- When modelling uncertainty in the environment
  - To quantify failures, express soft deadlines, QoS
    - "probability of frame being delivered within 5ms is at least 0.91"
  - To quantify environmental factors in decision support
    - "expected cost of reaching the goal is 100"
- When analysing system performance
  - To quantify arrivals, service, etc. characteristics
    - "in the long run, mean waiting time in a lift queue is 30 sec"

Real-world protocol examples

- Protocols featuring randomisation
  - Randomised back-off schemes
    - IEEE 802.11 (WiFi) Wireless LAN MAC protocol
  - Random choice of waiting time
    - Bluetooth, device discovery phase
  - Random choice of routes to destination
    - Crowds, anonymity protocol for internet routing
  - Random choice of a timing delay
    - Root contention in IEEE 1394 FireWire
  - Random choice over a set of possible addresses
    - IPv4 dynamic configuration (link-local addressing)
  - and more

- Continuous probability distribution needed to model network traffic, node mobility, random delays…

Probability elsewhere

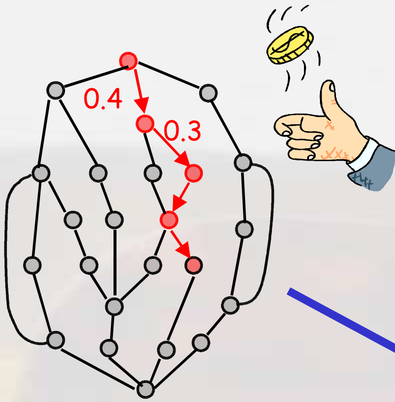
- In performance modelling
  - Pioneered by Erlang, in telecommunications, ca 1910
  - Models: typically continuous-time Markov chains
  - Emphasis on steady-state and transient probabilities

- In stochastic planning
  - Cf Bellman equations, ca 1950s
  - Models: Markov decision processes
  - Emphasis on finding optimum policies

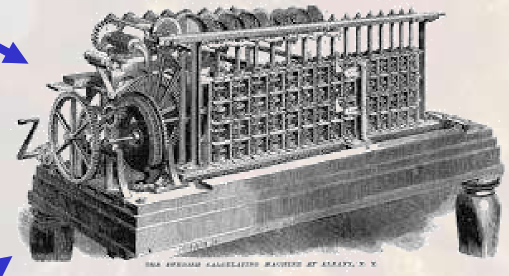
- Our focus, probabilistic model checking
  - Distinctive, on automated verification for probabilistic systems
  - Temporal logic specifications, automata-theoretic techniques
  - Shared models
  - Exchanging techniques with the other two areas

Probabilistic model checking…

in a nutshell

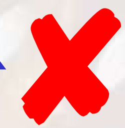


Probabilistic model



Probabilistic Model Checker

or



or

The probability

State 5: 0.6789
State 6: 0.9789
State 7: 1.0
…
State 12: 0
State 13: 0.1245

$\text{send} \rightarrow \mathbf{P}_{>0.9}(\lozenge \text{deliver})$

Probabilistic temporal logic specification

Probabilistic model checking with PRISM

- Models
  - Discrete-Time Markov Chains (DTMCs)
  - Markov Decision Processes (MDPs)
  - Continuous-Time Markov Chains (CTMCs)
  - Probabilistic Time Automata (PTAs)

- Specifications (informally)
  - "probability of shutdown occurring is at most…"
  - "probability of delivery within time deadline is …"
  - "expected time to message delivery is …"
  - "expected power consumption is …"

- Specifications (formally)
  - Probabilistic extensions of temporal logic (PCTL, CSL, PTCTL)
  - Probability, time, cost/rewards

Extending PRISM with mobility

- Models in PRISM
  - are described in reactive modules
    - :: extend with mobility, dynamic topology
    - :: extend with geographical positioning
    - :: extend with context-awareness
  - are finite-state, static and often huge
    - :: verification support for compositionality, abstraction
    - :: techniques for infinite state systems
    - :: combine with simulation-based methods
- Specifications
  - are temporal logic based:
    - :: add location-awareness
    - :: more expressive logics?

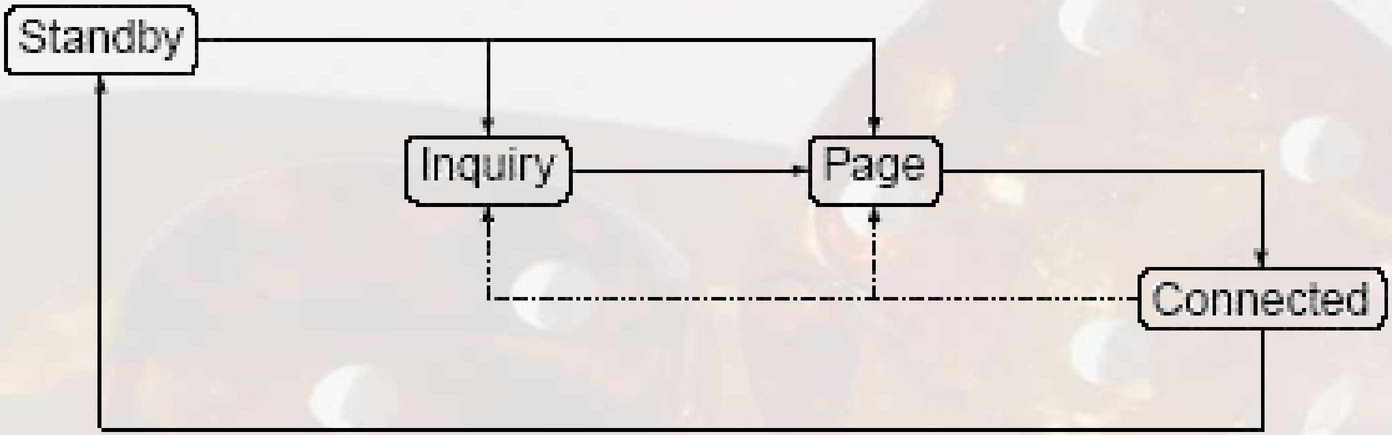
PRISM real-world case studies

- MDPs/DTMCs
  - Bluetooth device discovery [ISOLA'04]
  - Crowds anonymity protocol (by Shmatikov) [JCS 2004]
  - Randomised consensus [CAV'01]
  - Randomised Byzantine Agreement [FORTE'02]
  - NAND multiplexing for nanotechnology (with Shukla) [VLSI'04]
- CTMCs
  - Dynamic Power Management (with Shukla and Gupta) [HLDVT'02]
  - Dependability of embedded controller [INCOM'04]
- PTAs
  - IPv4 Zeroconf dynamic configuration [FORMATS'03]
  - Root contention in IEEE 1394 FireWire [FAC 2003, STTT 2004]
  - IEEE 802.11 (WiFi) Wireless LAN MAC protocol [PROBMIV'02]

Bluetooth protocol overview

- Short-range low-power wireless protocol
  - Personal Area Networks (PANs)
  - Open standard, versions 1.1 and 1.2
  - Widely available in phones, PDAs, laptops, …
- Uses frequency hopping scheme
  - To avoid interference (uses unregulated 2.4GHz band)
  - Pseudo-random frequency selection over 32 of 79 frequencies
  - Inquirer hops faster
  - Must synchronise hopping frequencies
- Network formation
  - Piconets (1 master, up to 7 slaves)
  - Self-configuring: devices discover themselves
  - Master-slave roles

States of a Bluetooth device



- Master looks for device, slave listens for master
- Standby: default operational state
- Inquiry: device discovery
- Page: establishes connection
- Connected: device ready to communicate in a piconet

Why focus on device discovery?

- Performance of device discovery crucial
  - No communication before initialisation
  - First mandatory step: device discovery
- Device discovery
  - Exchanges information about slave clock times, which can be used in later stages
  - Has considerably higher power consumption
  - Determines the speed of piconet formation

Frequency hopping

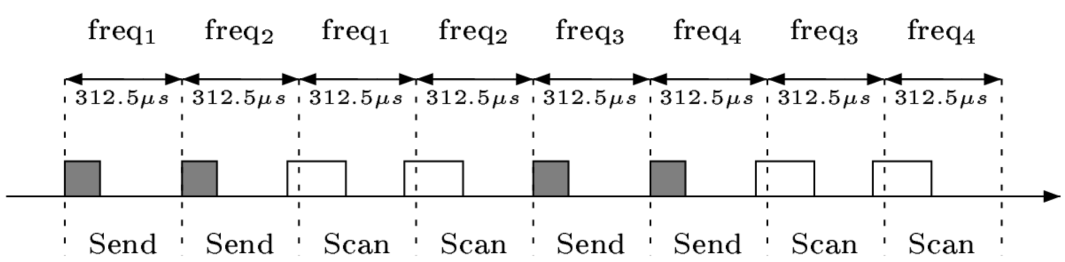


Fig. 1. Timing of the inquiring device's behaviour

- Clock CLK, 28 bit free-running, ticks every 312.5µs
- Inquiring device (master) broadcasts inquiry packets on two consecutive frequencies, then listens on the same two (plus margin)
- Potential slaves want to be discovered, scan for messages
- Frequency sequence determined by formula, dependent on bits of clock CLK (k defined on next slide):

$$freq = [CLK_{16-12}+k+ (CLK_{4-2,0}-CLK_{16-12}) \bmod 16] \bmod 32$$

Frequency hopping sequence

freq = [CLK_{16-12}+k+ ($CLK_{4-2,0}$- CLK_{16-12}) mod 16] mod 32

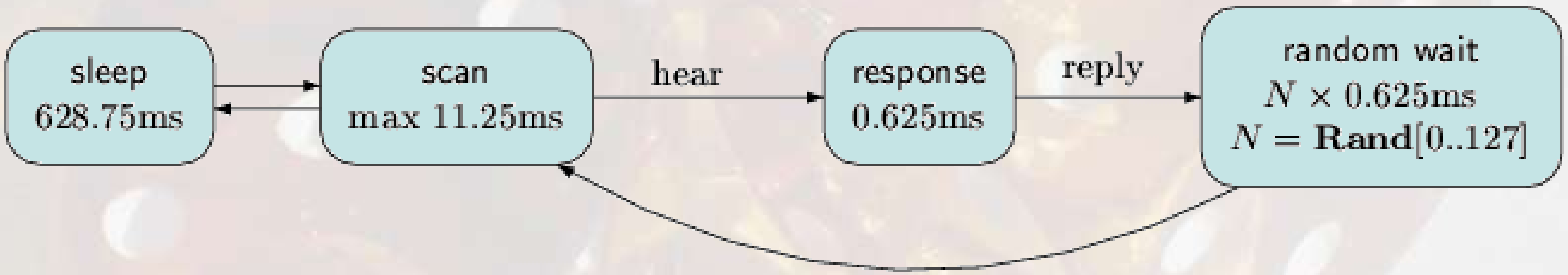
- Two trains (=lines)
- k is offset that determines which train
- Swaps between trains every 2.56 sec
- Each line repeated 128 times

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	2	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	2	3	20	21	22	23	24	25	26	27	28	29	30	31	32
17	18	19	20	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	6	7	8	9	10	11	12	13	14	15	16
1	2	3	4	5	6	23	24	25	26	27	28	29	30	31	32
1	2	3	4	5	6	7	24	25	26	27	28	29	30	31	32
17	18	19	20	21	22	23	24	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	10	11	12	13	14	15	16
1	2	3	4	5	6	7	8	9	10	27	28	29	30	31	32
1	2	3	4	5	6	7	8	9	10	11	28	29	30	31	32
17	18	19	20	21	22	23	24	25	26	27	28	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	14	15	16
1	2	3	4	5	6	7	8	9	10	11	12	13	14	31	32
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	32
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
17	18	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	4	5	6	7	8	9	10	11	12	13	14	15	16
1	2	3	4	21	22	23	24	25	26	27	28	29	30	31	32
1	2	3	4	5	22	23	24	25	26	27	28	29	30	31	32
17	18	19	20	21	22	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	8	9	10	11	12	13	14	15	16
1	2	3	4	5	6	7	8	25	26	27	28	29	30	31	32
1	2	3	4	5	6	7	8	9	26	27	28	29	30	31	32
17	18	19	20	21	22	23	24	25	26	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	12	13	14	15	16
1	2	3	4	5	6	7	8	9	10	11	12	29	30	31	32
1	2	3	4	5	6	7	8	9	10	11	12	13	30	31	32
17	18	19	20	21	22	23	24	25	26	27	28	29	30	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	16

Sending and receiving in Bluetooth

- Sender: broadcasts inquiry packets, sending according to the frequency hopping sequence, then listens, and repeats

- Receiver: follows the frequency hopping sequence, own clock



- Listens continuously on one frequency
- If hears message sent by the sender, then replies on the same frequency
- Random wait to avoid collision if two receivers hear on same frequency

Bluetooth modelling

- Very complex interaction
  - Genuine randomness, probabilistic modelling essential
  - Devices make contact only if listen on the right frequency at the right time!
  - Sleep/scan periods unbreakable, much longer than listening
  - Cannot scale constants (approximate results)
  - Cannot omit subactivities, otherwise oversimplification
- Huge model, even for one sender and one receiver!
  - Initial configurations dependent on 28 bit clock
  - Cannot fix start state of receiver, clock value could be arbitrary
  - 17,179,869,184 possible initial states
- But is a realistic future ubiquitous computing scenario!

What about other approaches?

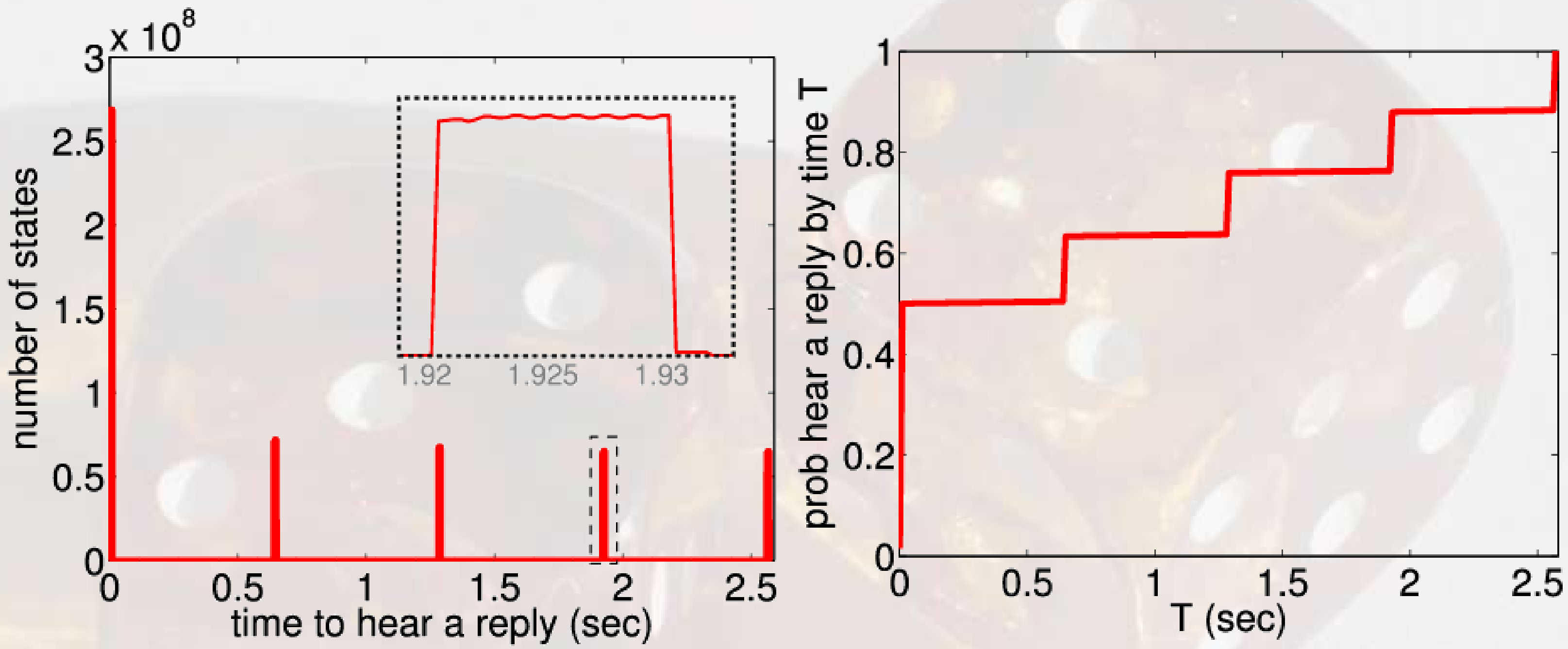
- Indeed, others have tried…
  - network simulation tools (BlueHoc)
  - analytical approaches
- But
  - simulations obtain averaged results, in contrast to best/worst case analysis performed here
  - analytical approaches require simplifications to the model
  - it is easy to make incorrect probabilistic assumptions, as we can demonstrate
- There is a case for all types of analyses, or their combinations…

Lessons learnt…

- Must optimise/reduce model
  - Assume negligible clock drift
  - Discrete time, obtain a DTMC
  - Manual abstractions, combine transitions, etc
  - Divide into 32 separate cases
  - Success (exhaustive analysis) with one/two replies

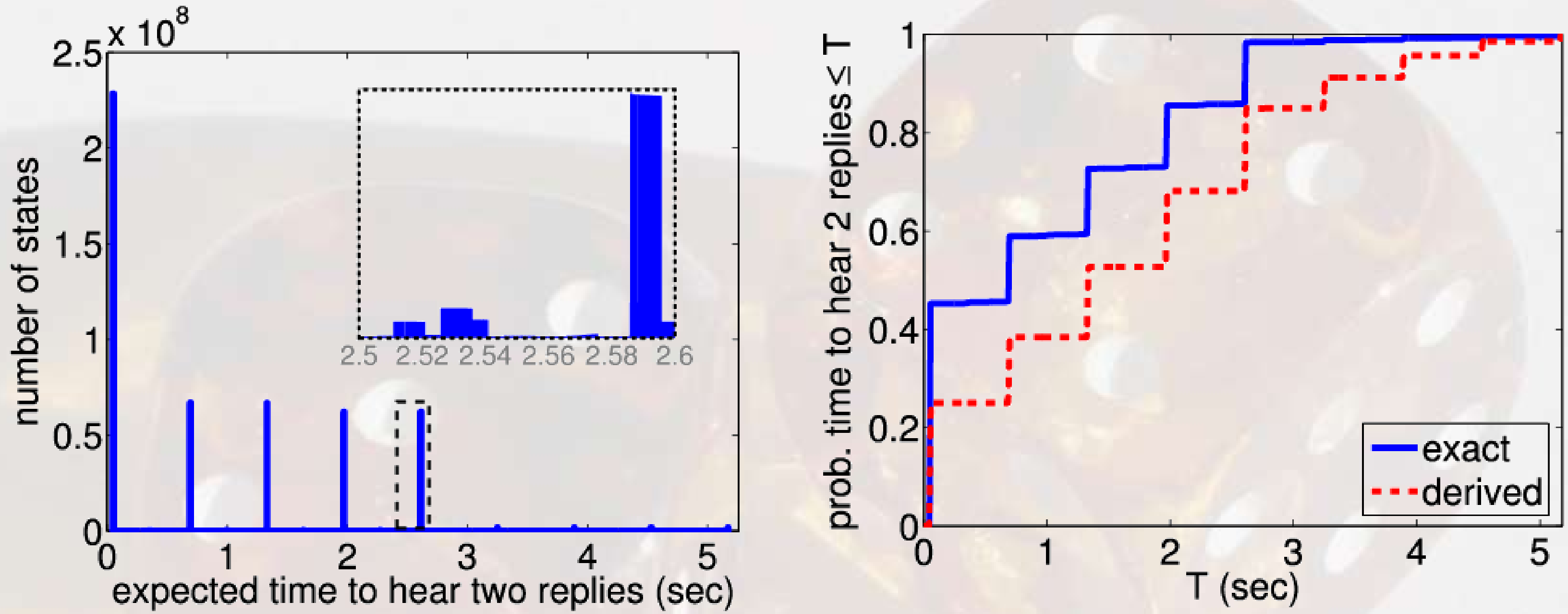
- Observations
  - Work with realistic constants, as in the standard
  - Analyse v1.2 and 1.1, confirm 1.1 slower
  - Show best/worst case values, can pinpoint scenarios which give rise to them
  - Also obtain power consumption analysis

Time to hear 1 reply



- Max time to hear is 2.5716sec, in 921,600 possible initial states, (Min 635μs)
- Cumulative: assume uniform distribution on states when receiver first starts to listen

Time to hear 2 replies



- Max time to hear is 5.177sec (16,565 slots), in 444 possible initial states
- Cumulative (derived): assumes time to reply to 2nd message is independent of time to reply to 1st (incorrect, compare with exact curve obtained from model checking)

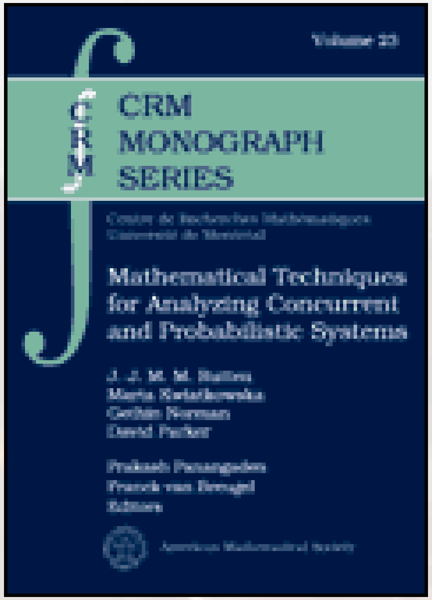
Related projects

- FORWARD (this case study, see ISOLA'04)
  - Performance modelling of MAC layer of Bluetooth
  - Security analysis of Bluetooth
- Modelling and verification of mobile ad hoc network protocols
  - Modelling language with mobility and randomisation
  - Model checking algorithms & techniques
  - Tool development & implementation
  - Modelling timing properties of AODV
- Focus on properties
  - "probability of delivery within time deadline is …"
  - "expected time to device discovery is …"
  - "expected power consumption is …"

Challenges for future

- Exploiting structure
  - Abstraction, data reduction, compositionality…
  - Parametric probabilistic verification?
- Proof assistant for probabilistic verification
- Extension for mobility
- Extension for hybrid systems
- Simulation, statistical testing [Younes]
- Approximation methods
- Continuous PTAs
  - Efficient model checking methods?
- More expressive specifications
  - Probabilistic LTL/PCTL*/mu-calculus?
- Real software, not models!

For more information…



J. Rutten, M. Kwiatkowska, G. Norman and D. Parker
Mathematical Techniques for Analyzing

P. Panangaden and F. van Breugel (editors), CRM Monograph Series, vol. 23, AMS
March 2004

www.cs.bham.ac.uk/~dxp/prism/

- *Case studies, statistics, group publications*
- Download, version 2.0 (> 750 users)
- Publications by others and courses that feature PRISM…