# Model checking for probability and time: from theory to practice

Marta Kwiatkowska
School of Computer Science
THE UNIVERSITY OF BIRMINGHAM
www.cs.bham.ac.uk/~mzk
www.cs.bham.ac.uk/~dxp/prism

LICS 2003

# Overview

- Motivation
  - Why probability?  How does it help?
- A glimpse of theory, how it all began
  - Probabilistic models, specification languages, algorithms
- Making theory work in practice
  - Implementing a probabilistic model checker (PRISM)
- Case studies of real-world protocols, what we have achieved
  - Crowds anonymity protocol
  - IPv4 dynamic configuration protocol
  - Root contention in IEEE 1394 FireWire
- Challenges for future

# Computing in the past…

Mainframes
Wired networks, modems
Text only I/O devices
Need for expert help to run, configure…

# The future: ubiquitous computing

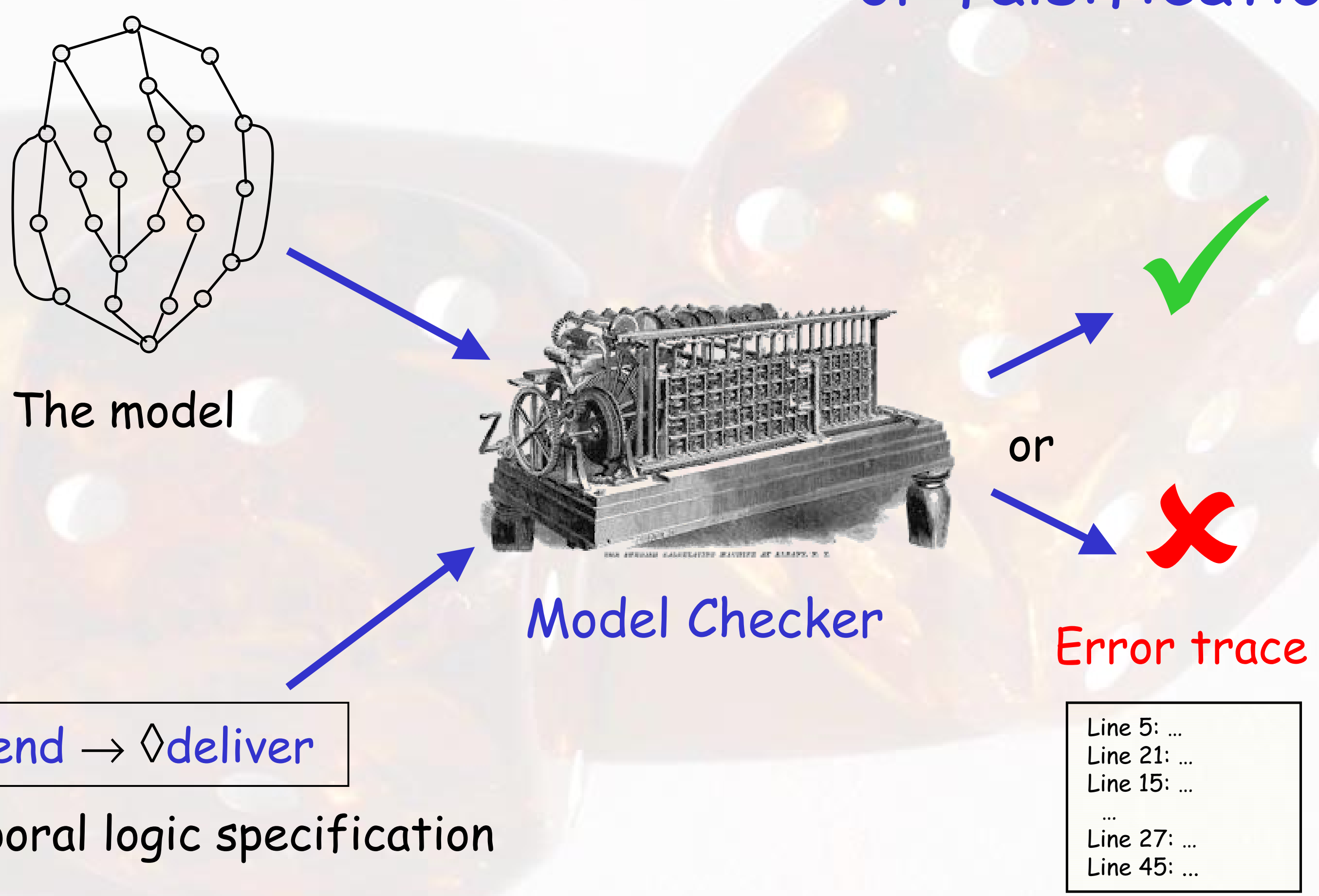

Mobile, wearable, wireless devices (WiFi, Bluetooth)
Ad hoc, dynamic, ubiquitous computing environment
Security, privacy, anonymity protection on the Internet
Self-configurable - no need for men/women in white coats!
Fast, responsive, power efficient, …

# Probability helps

- In distributed 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
    - “the chance of shutdown is at most 0.1%”
    - “the probability of a frame delivered within 5ms is at least 0.91”
  - To quantify environmental factors in decision support
    - “the 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”

# Verification via model checking…

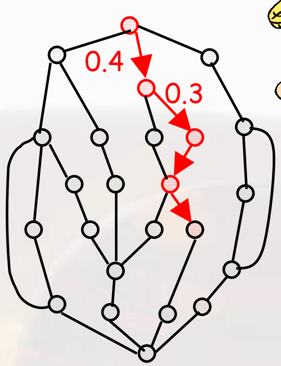
## or falsification?



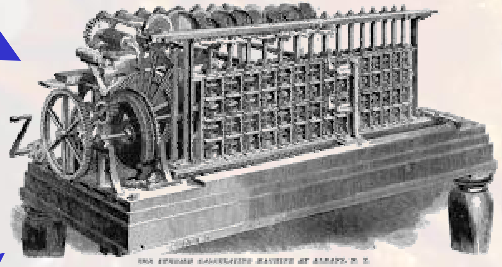

Also refinement checking, equivalence checking, …

# Probabilistic model checking…

in a nutshell




Probabilistic model



Probabilistic Model Checker

$\text{send} \rightarrow \mathcal{P}_{\geq 0.9}(\Diamond \text{deliver})$

Probabilistic temporal logic specification

✓

or

✗

or

The probability

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

# A historical interlude

## Probabilistic Methods in Verification

(PROBMIV'98)

A Pre-LICS'98 Workshop

19-20 June 1998, Indianapolis, Indiana, USA

### Workshop description and aims

*Scientific Justification:* While there has been a steady current of research activity in probabilistic logics and systems for some years, little experimental work has been done up until now. This situation is beginning to change. Randomization has proved effective in deriving efficient distributed algorithms and is now widely used in practical applications, to mention computer networks and graphics. However, randomized algorithms are notoriously difficult to verify: the proofs of their correctness are complex, and therefore argued informally, and thus appropriate formal methods and tools are called for. These have to combine a variety of dissimilar techniques, from conventional proof theory and model checking, through systems modelling to linear algebra and probability theory.

www.cs.bham.ac.uk/~mzk/probmiv98.html

- Questions asked in panel session
  - Randomization – is it really used widely?
  - Where are the tools? heuristics?
  - Did you find any bugs?
- In this talk
  - Some answers
  - As always, new challenges!

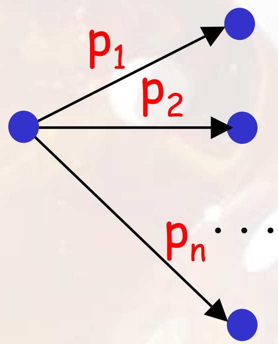
# Probability elsewhere

- 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 models: discrete

- Discrete time and probability
  - Discrete time Markov chains (DTMCs): probabilistic choice only
  - Markov decision processes (MDPs): probabilistic choice and nondeterminism

- Dense real-time, discrete probability
  - Probabilistic timed automata (PTAs): probabilistic choice, nondeterminism and dense real-time clocks




$\sum_i p_i = 1$

# Theory timeline: discrete models

Qualitative (with probability 1 or 0)

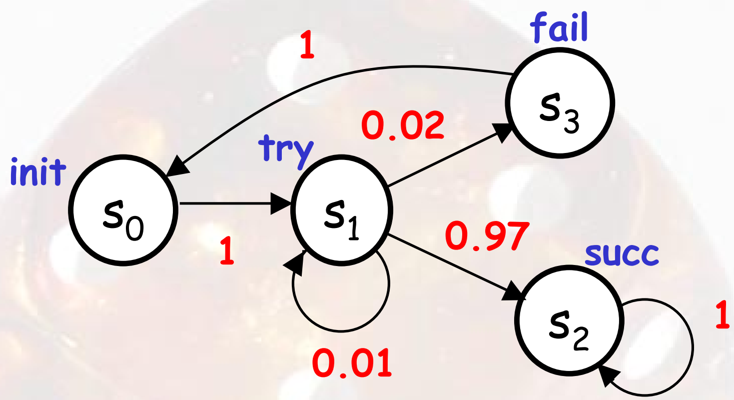
1983 Hart-Sharir-Pnueli
1985 Vardi
1988 Courcoubetis-Yannakakis

Quantitative (with arbitrary probability)

1991 Larsen-Skou (probab. bisimulation)
1994 Hansson-Jonsson (DTMC model checking)
1995 Bianco-de Alfaro (MDP model checking)
1995 Segala-Lynch (probab. simulation)
1997 Huth-Kwiatkowska [LICS] (probab. mu-calculus)
1997 Baier et al (DTMC model checking)
1998 Baier-Kwiatkowska (MDPs + fairness)
1999 Kwiatkowska-Norman-Segala-Sproston (PTAs)
2001 Kwiatkowska-Norman-Sproston (infinite state)

# Discrete-Time Markov Chains (DTMCs)

- Features:
  - Only probabilistic choice in each state




- Formally, $(S, s_0, P, L)$:
  - $S$ finite set of states
  - $s_0$ initial state
  - $P: S \times S \rightarrow [0,1]$ probability matrix, s.t. $\sum_{s'} P(s,s') = 1$, all $s$
  - $L: S \rightarrow 2^{AP}$ atomic propositions

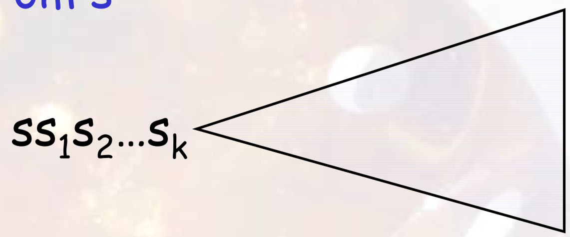
- Unfold into infinite paths $s_0 s_1 s_2 s_3 s_4 \ldots$ s.t. $P(s_i, s_{i+1}) > 0$, all $i$

- Probability for finite paths, multiply along path

  e.g. $s_0\, s_1\, s_1\, s_2$ is $1 \cdot 0.01 \cdot 0.97 = 0.0097$

# Probability space

- Intuitively:
  - Sample space = infinite paths $Path_s$ from s
  - Event = set of paths
  - Basic event = cone




- Formally, $(Path_s, \Omega, Pr)$
  - For finite path $\omega = ss_1\ldots s_n$, define probability

$$P(\omega) = \begin{cases} 1 \text{ if } w \text{ has length one} \\ P(s,s_1) \cdot \ldots \cdot P(s_{n-1},s_n) \text{ otherwise} \end{cases}$$

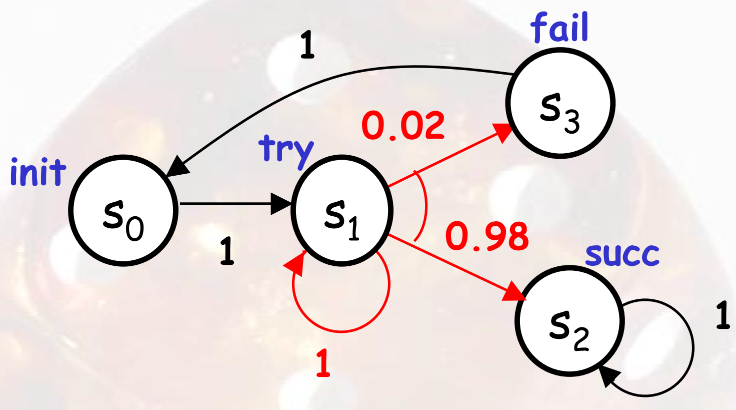
  - Take $\Omega$ least $\sigma$-algebra containing cones

$$C(\omega) = \{ \pi \in Path_s \mid \omega \text{ is prefix of } \pi \}$$

  - Define $Pr(C(\omega)) = P(\omega)$, all $\omega$
  - $Pr$ extends uniquely to measure on $Path_s$

# Markov Decision Processes (MDPs)

- Features:
  - Nondeterministic choice
  - Parallel composition of DTMCs




- Formally, $(S, s_0, Steps, L)$:
  - $S$ finite set of states
  - $s_0$ initial state
  - Steps maps states $s$ to sets of probability distributions $\mu$ over $S$
  - $L: S \rightarrow 2^{AP}$ atomic propositions

- Unfold into infinite paths $s_0 \mu_0 s_1 \mu_1 s_2 \mu_2 s_3 \ldots$ s.t. $\mu_i(s_i, s_{i+1}) > 0$, all i

- Probability space induced on $Path_s$ by adversary (policy) A mapping finite path $s_0 \mu_0 s_1 \mu_1 \ldots s_n$ to a distribution from $s_n$

# The logic PCTL: syntax

- Probabilistic Computation Tree Logic [HJ94,BdA95,BK98]
  - For DTMCs/MDPs
  - New probabilistic operator, e.g. send $\rightarrow \mathcal{P}_{\geq 0.9}(\Diamond \text{deliver})$

- The syntax of state and path formulas of PCTL is:

$$\phi ::= \text{true} \mid a \mid \phi \land \phi \mid \neg\phi \mid \mathcal{P}_{\sim p}(\alpha)$$
$$\alpha ::= X\,\phi \mid \phi\, U\, \phi$$

where $p \in [0,1]$ is a probability bound and $\sim \in \{ <, >, \ldots \}$

- Subsumes the qualitative variants [Var85,CY95]
  $\mathcal{P}_{=1}(\alpha)$, $\mathcal{P}_{>0}(\alpha)$

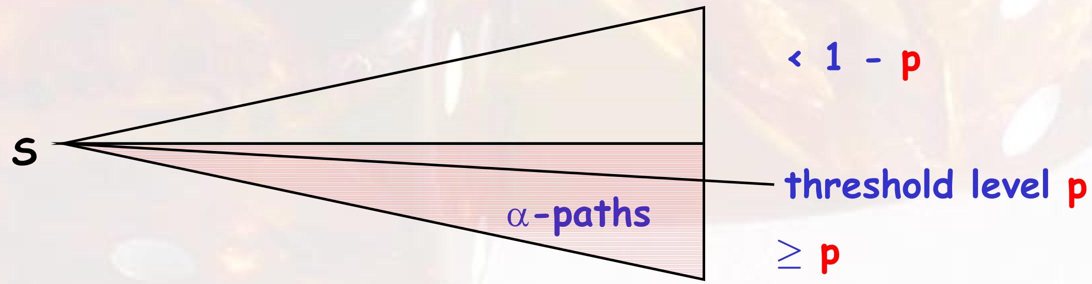
- Extension with cost/rewards and expectation operator $\mathcal{E}_{\sim c}(\phi)$

# The logic PCTL: semantics

- Semantics is parameterised by a class of adversaries Adv
  - "under any scheduling, the probability bound is true at state s"
  - reasoning about worst-case/best-case scenario

- The probabilistic operator is a quantitative analogue of $\forall$, $\exists$

$s \vDash_{Adv} \mathcal{P}_{\sim p}(\alpha) \quad \Leftrightarrow \quad Pr^A \{ \pi \in Path^A_s \mid \pi \vDash_{Adv} \alpha \} \sim p$ for all $A \in Adv$




- Semantics of remaining formulas standard

# PCTL semantics

- Semantics is parameterised by a class of adversaries Adv
  - "under any scheduling, the probability bound is true at state s"
  - reasoning about worst-case/best-case scenario

- The probabilistic operator is a quantitative analogue of $\forall$, $\exists$

$$s \vDash_{Adv} \mathcal{P}_{\sim p}(\alpha) \quad \Leftrightarrow \quad Pr^{A}\{\pi \in Path^{A}_{s} \mid \pi \vDash_{Adv} \alpha\} \sim p \text{ for all } A \in Adv$$

- Semantics of remaining formulas standard:

| | | |
|---|---|---|
| $s \vDash_{Adv} a$ | $\Leftrightarrow$ | $a \in L(s)$ |
| $s \vDash_{Adv} \neg\phi$ | $\Leftrightarrow$ | $s \vDash_{Adv} \phi$ |
| $s \vDash_{Adv} \phi_1 \wedge \phi_2$ | $\Leftrightarrow$ | $s \vDash_{Adv} \phi_1$ and $s \vDash_{Adv} \phi_2$ |
| $\pi \vDash_{Adv} X\, \phi$ | $\Leftrightarrow$ | $\pi = s_0 \cdots$ and $s_1 \vDash_{Adv} \phi$ |
| $\pi \vDash_{Adv} \phi_1\, U\, \phi_2$ | $\Leftrightarrow$ | $\pi = s_0 \cdots$ and $\exists\, k$ s.t. $s_k \vDash_{Adv} \phi_2$ and $\forall\, j < k\, .\, s_j \vDash_{Adv} \phi_1$ |

# The logic PCTL: model checking

- By induction on structure of formula, as for CTL
- For the probabilistic operator and Until, solve
  - recursive linear equation (DTMCs)
  - linear optimisation problem (form of value iteration)
  - typically iterative solution methods
- Need to combine
  - conventional graph traversal
  - numerical linear algebra and linear optimisation
- Qualitative properties (probability 1, 0) proceed by graph traversal [Var85,dAKNP97]

# PCTL model checking for DTMCs

- By induction on structure of formula
- For the probabilistic operator
  - $Sat(\mathcal{P}_{\sim p}(X\ \phi)) \Leftrightarrow \{s \in S \mid \sum_{s' \in Sat(\phi)} P(s,s') \sim p\}$
  - $Sat(\mathcal{P}_{\sim p}(\phi_1\ U\ \phi_2)) \Leftrightarrow \{s \in S \mid x_s \sim p\}$

  where $x_s$, $s \in S$, are obtained from the recursive linear equation

$$x_s = \begin{cases} 0 & \text{if } s \in S^{no} \\ 1 & \text{if } s \in S^{yes} \\ \sum_{s' \in S} P(s,s') \cdot x_{s'} & \text{if } s \in S \backslash (S^{no} \cup S^{yes}) \end{cases}$$

  and

  $S^{yes}$ – states that satisfy $\phi_1\ U\ \phi_2$ with probability exactly 1

  $S^{no}$ - states that satisfy $\phi_1\ U\ \phi_2$ with probability exactly 0

# PCTL model checking for DTMCs

- For the remaining formulas standard:

| | | |
|---|---|---|
| Sat(a) | = | L(a) |
| Sat($\neg\phi$) | = | S\Sat($\phi$) |
| Sat($\phi_1 \wedge \phi_2$) | = | Sat($\phi_1$) $\cap$ Sat($\phi_2$ ) |

- $S^{yes}$, $S^{no}$ can be precomputed by graph traversal [Var85] (or BDD fixed point computation)

- Need to combine
  - Conventional graph-theoretic traversal
  - Numerical linear algebra

# PCTL model checking for MDPs

- $S^{yes}$, $S^{no}$ can also be precomputed by graph traversal (BDD fixed point) [dAKNP97]

- The linear equation generalises to linear optimisation problems solvable iteratively, e.g.

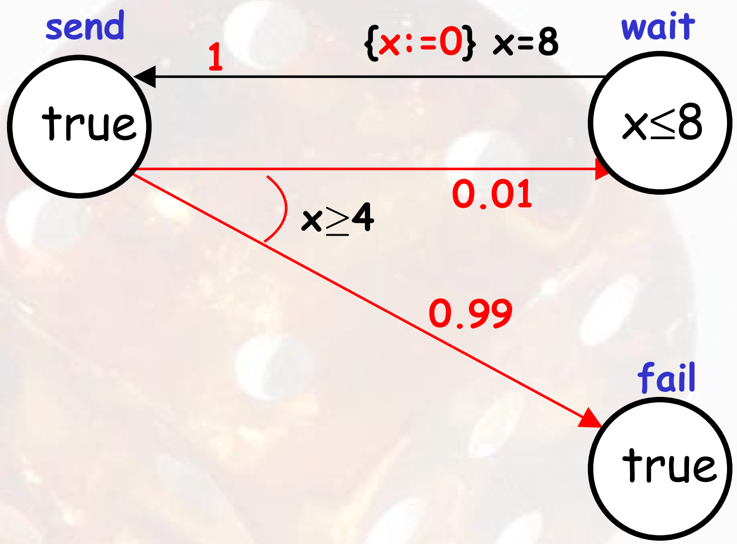
$$Sat(\ \mathcal{P}_{\geq p}(\phi_1\ \mathrm{U}\ \phi_2)\ ) \qquad \Leftrightarrow \qquad \{s \in S \mid x_s \geq p\}$$

$$x_s \ = \begin{cases} 0 & \text{if } s \in S^{no} \\ 1 & \text{if } s \in S^{yes} \\ \min_{\mu \in Steps(s)} \sum_{s' \in S} \mu(s') \cdot x_{s'} & \text{if } s \in S \backslash (S^{no} \cup S^{yes}) \end{cases}$$

- Need to combine
  - Conventional graph-theoretic traversal
  - Linear optimisation (simplified value iteration)

# Probabilistic Timed Automata: syntax

- Features:
  - Clocks, x, real-valued, upper bound $R_{max}$
  - Can be reset, e.g. x:=0
  - Clock constraints, e.g. $x \geq 5$
  - Probabilistic transitions




- Formally, $(S, s_0, Inv, prob, L)$:
  - S finite set of locations
  - $s_0$ initial location
  - Inv maps locations s to invariant clock constraints
  - prob probabilistic edge relation that yields the probability of moving from s to s' if enabled at s, resetting specified clocks
  - L: $S \rightarrow 2^{AP}$ atomic propositions

# PTAs: semantics

- Inherit infinite state space from Timed Automata
- Obtain infinite-state Markov decision process
  - Assume n clocks, clock valuations are points $\mathbf{v} \in \mathbb{R}^n_{\geq 0}$
  - States are (s,v), location-valuation pairs s.t. $\mathbf{v} \models \mathrm{Inv}(s)$
  - Transitions are
    - time elapse by some time t if Inv(s) satisfied along, and
    - discrete probabilistic transitions induced from prob
- Unfold into paths $s_0\, a_0\, \mu_0\, s_1\, a_1\, \mu_1\, s_2\, a_2\, \mu_2\, s_3 \ldots$
- Adversaries select transition-distribution, time divergent
- Potential undecidability, adapt decidable methods for TAs
  - uncountably many states, finite probabilistic branching

# PTAs: semantics

- Assume n clocks, $t, t' \in \mathbb{R}_{\geq 0}$, $\mathbf{v}, \mathbf{v}' \in \mathbb{R}^n_{\geq 0}$ clock valuations
  States: (s,**v**), where s location, **v** clock valuation, $\mathbf{v} \models Inv(s)$
  Transitions (ranged over by $a_i$):
    time elapse $(s,\mathbf{v}) \rightarrow^{\tau} (s,\mathbf{v}+t)$, with probability 1
      if Inv(s) satisfied by **v**+t and **v**+t' for all $0 \leq t' \leq t$
    discrete transition $(s,\mathbf{v}) \rightarrow^{\mu} (s',\mathbf{v}')$
      if $\exists\, \mu \in$ prob enabled at (s,**v**) and probability of moving to (s',**v**') resetting clocks in X is induced from prob
- Unfold into paths $s_0\, a_0\, \mu_0\, s_1\, a_1\, \mu_1\, s_2\, a_2\, \mu_2\, s_3 \ldots$
- Adversaries select $(a, \mu)$, time divergent
- Obtain infinite-state Markov decision process
  - Uncountably many states
  - Finite probabilistic branching

# The logic PTCTL

- Probabilistic Timed CTL for PTAs [KNSS99,KNSS02]
  - Based on TCTL [AD94]
  - Add probabilistic operator $\mathcal{P}_{\sim p}(\cdot)$ of PCTL

- Syntax

  $\phi$ ::= true | a | $\zeta$ | $\phi \wedge \phi$ | $\neg\phi$ | z.[$\phi$] | $\mathcal{P}_{\sim p}(\phi_1 \ \mathrm{U} \ \phi_2)$

  where z ranges over formula clocks, $\zeta$ are clock constraints over formula and system clocks

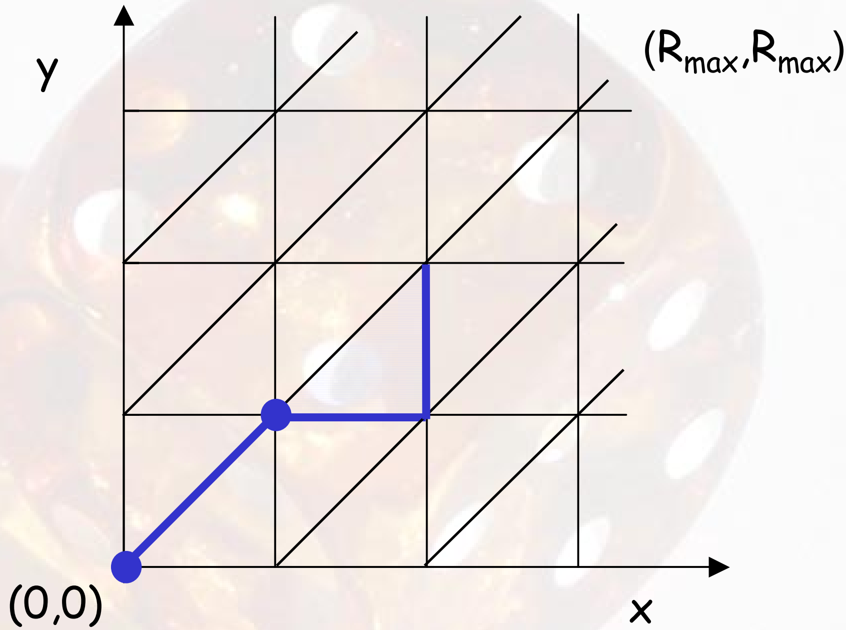
- Example: z.[ $\mathcal{P}_{\geq 0.98}$ ($\Diamond$ delivered $\wedge$ z < 5)]

  "under any scheduling, with probability $\geq$ 0.85 the message is correctly delivered within 5 ms"

- Semantics derived from PCTL and TCTL

# Model checking for PTAs: regions

- Region equivalence
  - finite partition of (P)TA state space
  - time abstract region graph




- Quotient preserves satisfaction
  - clock constraints
  - (P)TCTL formulas

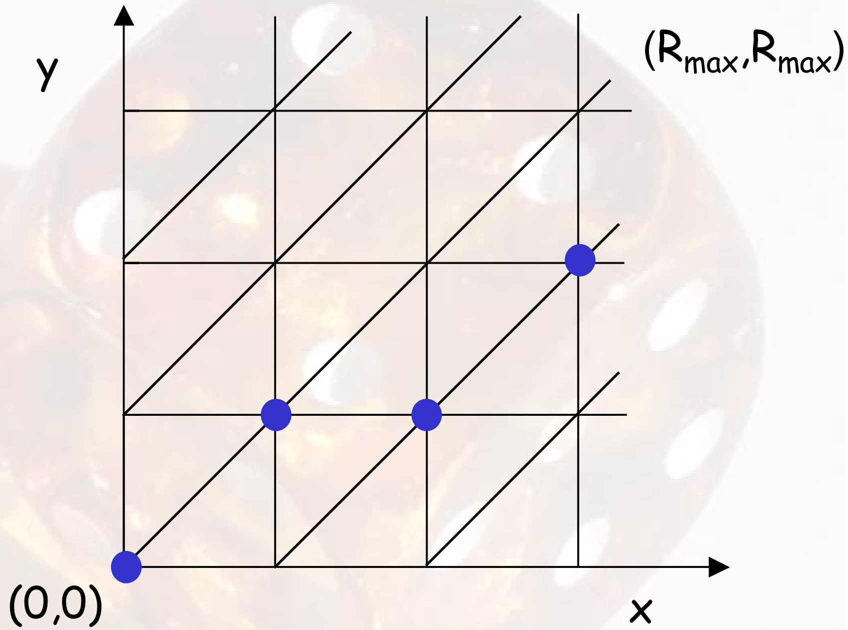
- Construct time-abstract MDP over regions
- Translate PTCTL to PCTL, model check the MDP

- Problem: high complexity, exponential in number of clocks

# Model checking for PTAs: digital clocks

- ε-digitisation [HMP92]
  - restrict to closed, diagonal free PTAs
  - integer-valued clocks

- Digitisation preserves
  - minimum/maximum reachability probability
  - minimum/maximum expected reachability

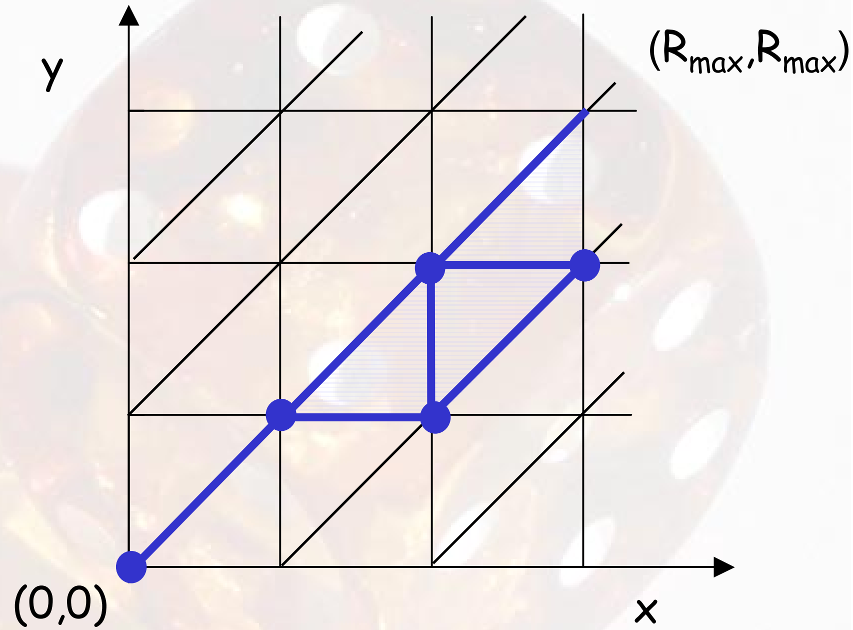



- Can build a finite-state MDP directly, model check the MDP
- Expressiveness restriction no problem, often very efficient

- Problem: state space explosion for large constants

# Model checking for PTAs: symbolic

- Zones
  - conjunctions of atomic constraints of the form $x \sim c$ and $x - y \sim c$, where $\sim \in \{<, \leq, \geq, >\}$
  - time abstract zone graph



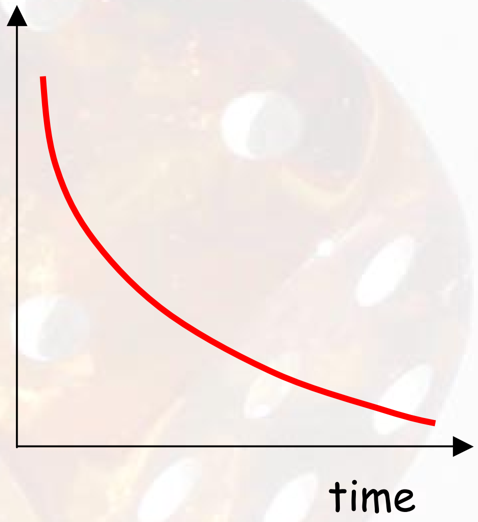

- Min/max probabilities need not be preserved

- Construct time-abstract MDP over zones
- Model check reachability on the MDP, forwards (post)
- Model check PTCTL, backwards (pre)

- Problem: loss of on-the-fly

# Probabilistic models: continuous

- Continuous probability distributions, discrete space
  - Continuous time Markov chains and generalisations (CTMCs, GSMPs): mainly exponential distributions, no nondeterminism
  - Continuous PTAs, Interactive Markov chains (IMCs): admit nondeterminism
- Continuous probability distributions, continuous space
  - Labelled Markov Processes (LMPs): no nondeterminism, reactive




$\int_0^{+\infty} f(x)dx = 1$

# Theory timeline: continuous models

Continuous distributions

1991 Alur-Courcoubetis-Dill (GSMPs)

1996 Aziz-Sanwal-Singhal-Brayton (logic CSL)

1998 de Alfaro (long-run average)

1999 Baier, Katoen, Hermanns (CTMC model checking)

2000 Baier, Haverkort, Hermanns, Katoen (uniformis.)

2000 Kwiatkowska-Norman-Segala-Sproston (cont. PTAs)

Continuous space, approximation

1997 Blute-Desharnais-Edalat, Panangaden [LICS] (bisim. LMPs)

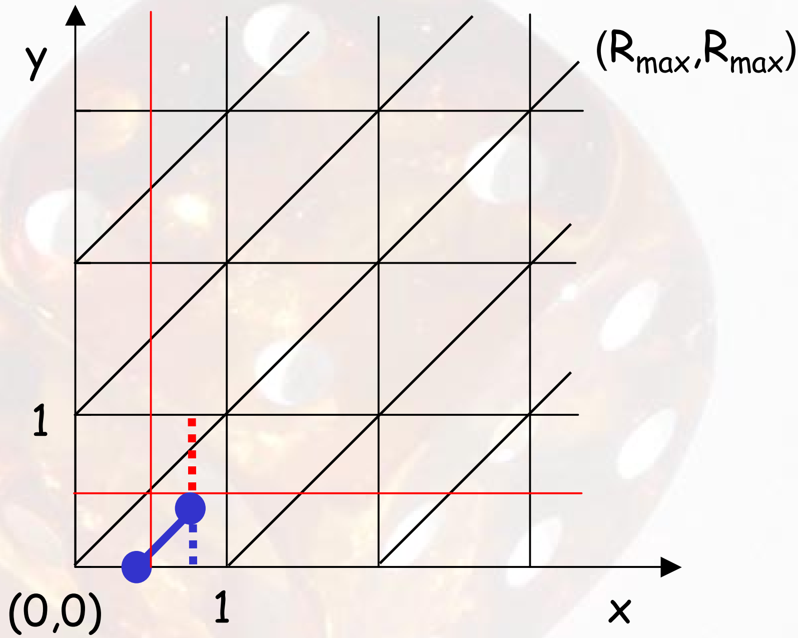
1999 Desharnais (logic LMPs)

2000 Desharnais-Gupta-Jagadeesan-Panangaden [LICS] (metric)

2003 Desharnais-Danos [LICS] (approx. LMPs)

# Continuous PTAs

- Allow clock reset according to cont. probability distribution
- Region graph no longer works [Alur]
  - Set x to random[0,1], y to 0
  - When x < 1, reset y to random[0,1]
  - Consider transitions x=1, y=1
  - If y < 0.5, x = 1 first, else don't know (error)




- Can approximately model check by subdividing region graph
- Problem: prohibitive complexity!!!

# Probabilistic model checking in practice

- Model construction: probability matrices
  - Enumerative
    - Manipulation of individual states
    - Size of state space main limitation
  - Symbolic
    - Manipulation of sets of states
    - Compact representation possible in case of regularity
- Temporal logic model checking: currently limited to
  - discrete probab/space models
  - CTMCs (omitted from presentation, see paper)
- Simulation
  - Admits more general distributions

# What is involved… more detail

- For DTMCs/MDPs:
  - Graph-theoretic algorithms (BDD fixpoinT0
  - Linear equation system solving (for DTMCs)
  - Linear optimisation (for MDPs)
  - Probability-1 and probability-0 precomputation step
    - improved efficiency via BDD fixed point calculation

- For PTAs, reduce to MDP model checking:
  - Closed diagonal-free PTAs can be model checked directly
  - Or, via forwards/backwards zone graph exploration iterating post/pre operations, then build an MDP over zones

- Continuous models
  - Translation to DTMCs (for CTMCs), exponential distributions

# Timeline: probabilistic verification tools

## Discrete time Markov chains

- 1994 TPWB (Hansson)
- 1998 ProbVerus (Hartonas-Garmhausen, et al)

## Markov decision processes

- 2000 PRISM (Probabilistic Symbolic Model Checker)
- 2001 Rapture

## Continuous time Markov chains

- 2000 ETMCC (Erlangen-Twente Markov Chain Checker)
- 2001 PRISM

## Probabilistic timed automata

- 2001 KRONOS+PRISM
- 2002 PRISM (digital clocks, costs/rewards)

# The PRISM tool: overview

- Functionality
  - Direct support for models: DTMCs, MDPs and CTMCs
  - Probabilistic temporal logic model checking
  - Extension with costs/rewards, expectation operator
  - Connection from KRONOS to PRISM for PTAs
- Input languages
  - System description
    - probabilistic extension of reactive modules [Alur and Henzinger]
  - Logics: PCTL and CSL (Continuous Stochastic Logic)
- Implementation
  - Symbolic model construction (MTBDDs), uses CUDD [Somenzi]
  - Three numerical computation engines
  - Written in Java and C++

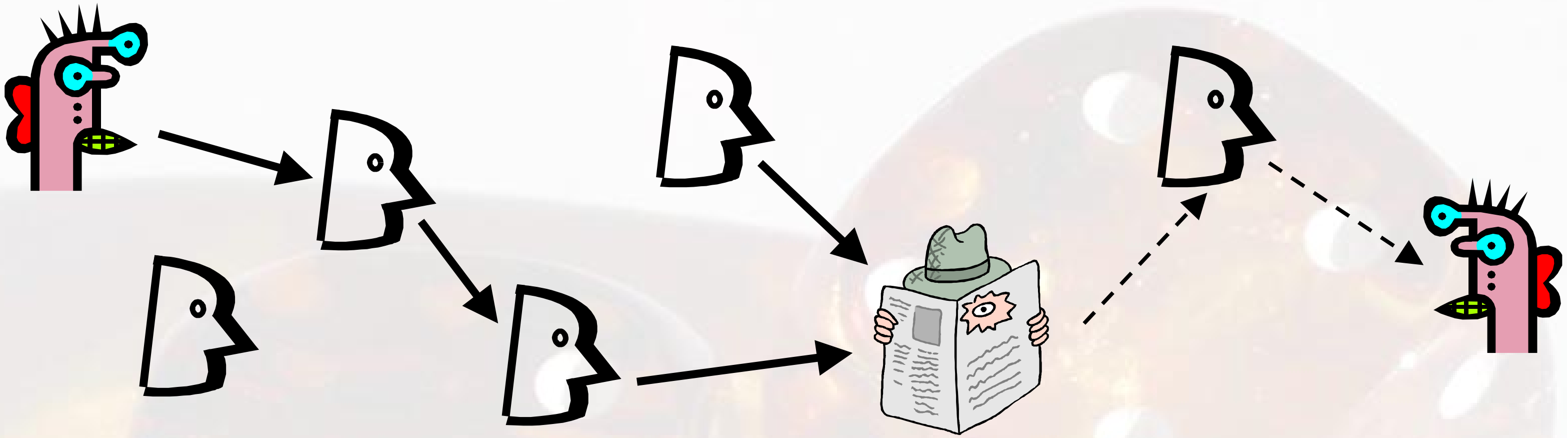
# The PRISM tool: implementation

- Numerical engines
  - Symbolic, MTBDD based
    - Fast construction, reachability analysis
    - Very large models if regularity
  - Enumerative, sparse-matrix based
    - Generally fast numerical computation
    - Model size up to millions
  - Hybrid
    - Speed comparable to sparse matrices for numerical calculations
    - Limited by size of vector
- Experimental results
  - Several large scale examples: $10^{10}$ - $10^{30}$ states
  - No engine wins overall
  - See www.cs.bham.ac.uk/~dxp/prism

# PRISM real-world case studies

- MDPs/DTMCs
  - Crowds anonymity protocol [Reiter & Rubin] (by Shmatikov)
  - Probabilistic contract signing (by Norman & Shmatikov)
  - Randomised consensus protocol [Aspnes & Herlihy]
  - Randomised Byzantine Agreement [Cachin, Kursawe and Shoup]
- CTMCs
  - Dynamic Power Management [Qiu, Wu & Pedram] (joint work with Shukla and Gupta)
- PTAs
  - IPv4 dynamic configuration [Cheshire, Adoba, Guttman]
  - Root contention in IEEE 1394 FireWire
  - IEEE 802.11 (WiFi) Wireless LAN MAC protocol

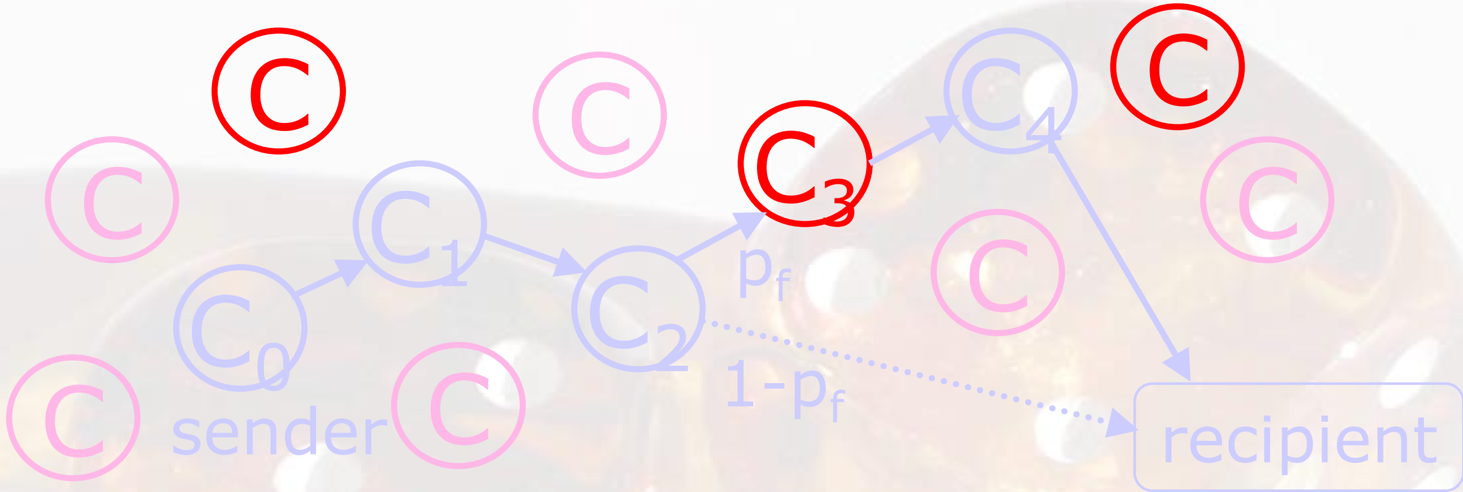
# Case study: Anonymity [by Shmatikov]



- Main idea, gossip-based
  - Hide source of messages by routing them randomly
  - Routers cannot tell if the apparent source of the message is the actual sender, or simply another router
  - Secure against local attackers
- Existing implementations
  - Crowds, Freenet, onion routing, etc.

Shmatikov, CSFW'02, to appear in Journal of Computer Security

# Crowds [Reiter, Rubin 98]

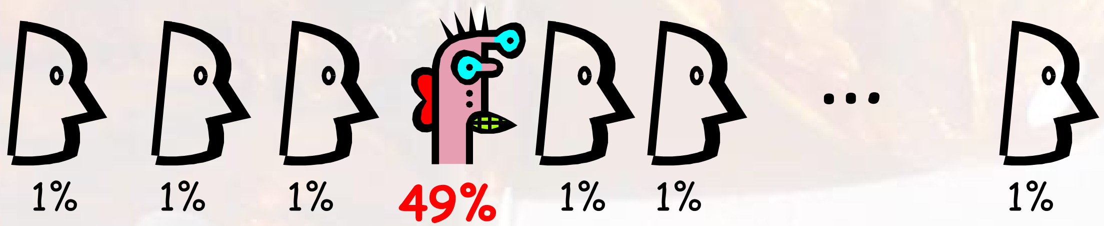



- Sender randomly chooses a path through the crowd
- Some routers are honest, some corrupt
- To formulate path, honest routers:
  - with probability $p_f$ route to the next member on the path or itself
  - with probability $1-p_f$ send directly to the recipient
- Once formulated, path is used for sending messages
- New paths must be established when members join or leave

# What does Anonymity mean?

- Beyond suspicion
  - The observed source of the message is no more likely to be the actual sender than anybody else (considered for single path)

- Probable innocence (holds for Crowds if few corrupt routers)
  - Probability < 50% that the observed source of the message is the actual sender
  - But is it enough? Can attackers relate multiple paths?




Maybe OK for plausible deniability (repudiation)

# The Anonymity study

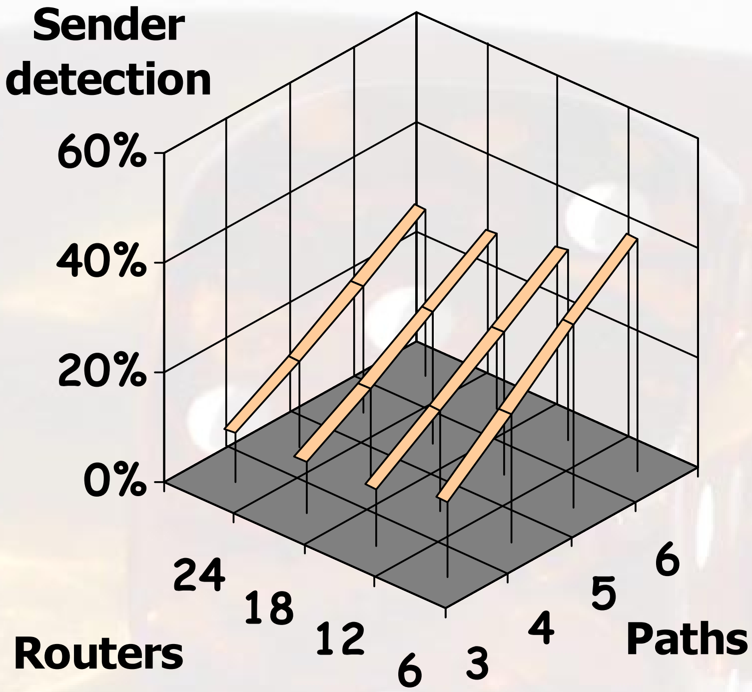
- Seeking clarity of the meaning of Anonymity…

- Used the PRISM probabilistic model checker to
  - model realistic configurations of the protocol as DTMCs
  - automatically compute and probabilities for finite configurations
  - plot graphs

- Key properties:
  - Sender detection: What is the probability of observing the actual sender more than once over multiple paths?
    [also studied independently – analytically]

  - Attacker's confidence: What is the probability of observing only the actual sender more than once?

# Sender detection (multiple paths)



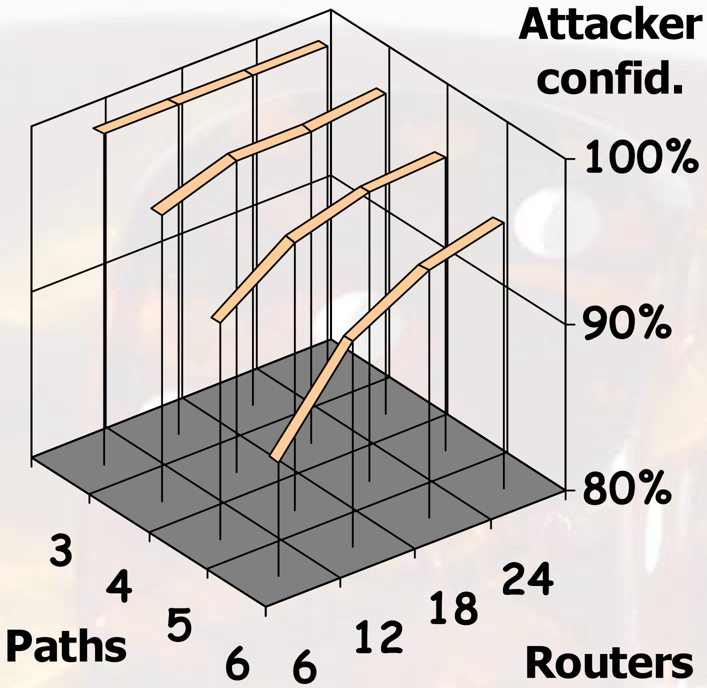

1/6 of routers are corrupt

- All configurations satisfy probable innocence
- Probability of observing the actual sender increases with the number of paths observed
- … but drops with the increase in crowd size

Is this an attack?

- Building new paths unavoidable
- Crowds has no mechanism for preventing attacker from correlating same-sender paths
  - e.g. decoy traffic (onion routing)

# Attacker's confidence




1/6 of routers are corrupt

- Confidence = observing only the actual sender
- Confidence grows with crowd size
- Maybe this is not so strange
  - Actual sender appears in every path
  - … others only with small probability

## Is this an attack?

- Large crowds: lower probability to catch senders but higher confidence that the caught agent is the sender

# Case Study: IPv4 Zeroconf Protocol

- IPv4 Zeroconf [Cheshire-Adoba-Guttman 2002]
  - new IETF standard for dynamic self-configuration of network interfaces
  - link-local (no routers within the interface)
  - no requirement of an active DHCP server
  - aimed at home networks, wireless ad hoc networks, hand-held devices
  - “plug and play”

- Self-configuration (zero-effort!)
  - performs assignment of IP addresses
  - symmetric, distributed protocol
  - uses random choice and timing delays

www.ietf.org/internet-drafts/draft-ietf-zeroconf-ipv4-linklocal-07.txt

# IPv4 Zeroconf Standard


- Main idea
  - Select an IP address out of 65024 at random
  - Send a probe querying if address in use, and listen for r seconds
  - If positive reply received, restart
  - Otherwise, continue sending probes (n = 4) and listening (r secs)
  - If no reply, start using the new IP number
- Set r = 2 seconds for unreliable networks, 0.2 otherwise

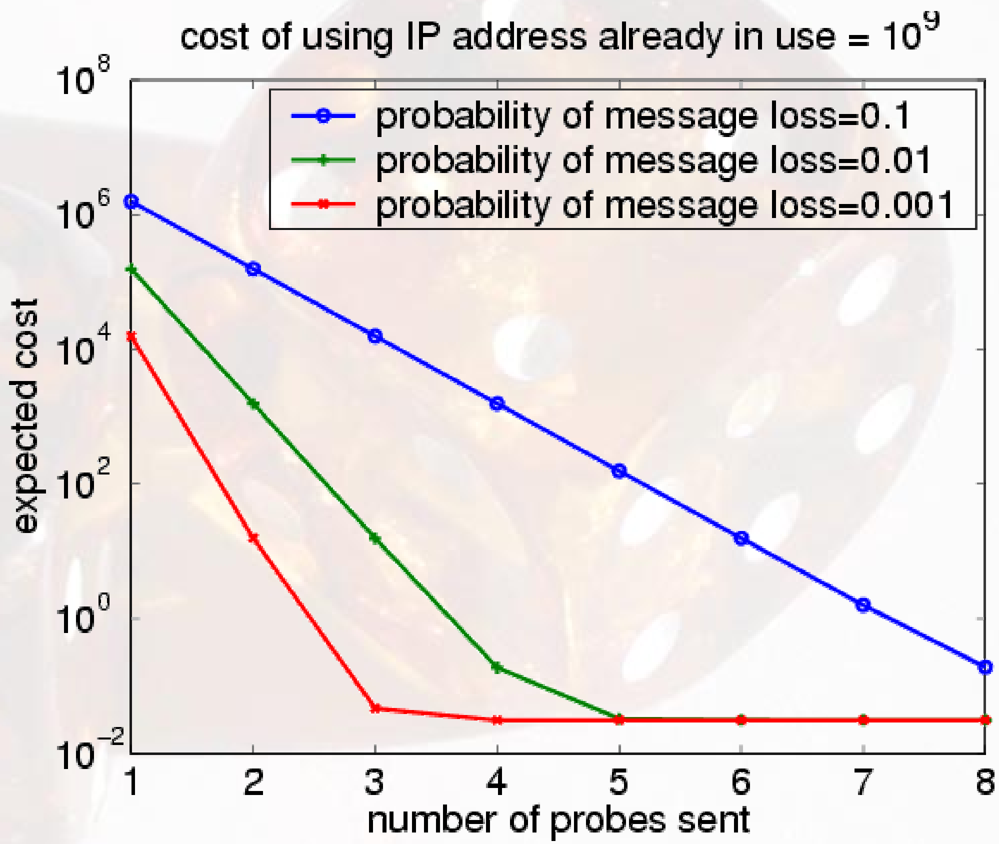
# Will it work?

- What can go wrong…
  - IP number may be in use but
    - Probes or replies may get lost/delayed, host may be too busy
    - Necessary to kill active TCP/IP connections if address collision – high cost!
  - Self-configuration delays may become unacceptable
    - Would you wait 8 seconds to self-configure your PDA?
  - No justification for parameters n, r

- Case studies:
  - DTMC and Markov reward models, analytical [Bohnenkamp-van der Stok-Hermanns-Vaandrager'03] and [Andova-Katoen-03]
  - DTMC model using PRISM, this talk
  - TA model using UPPAAL [Zhang-Vaandrager'02]
  - PTA model with digital clocks using PRISM [Kwiatkowska-Norman-Sproston'03]

# Cost versus performance trade-off

- Expected cost of OK or error
- Vary the number of probes sent and probability of message loss
- Necessary to increase the number of probes to reduce the expected cost
- Expected time then increases

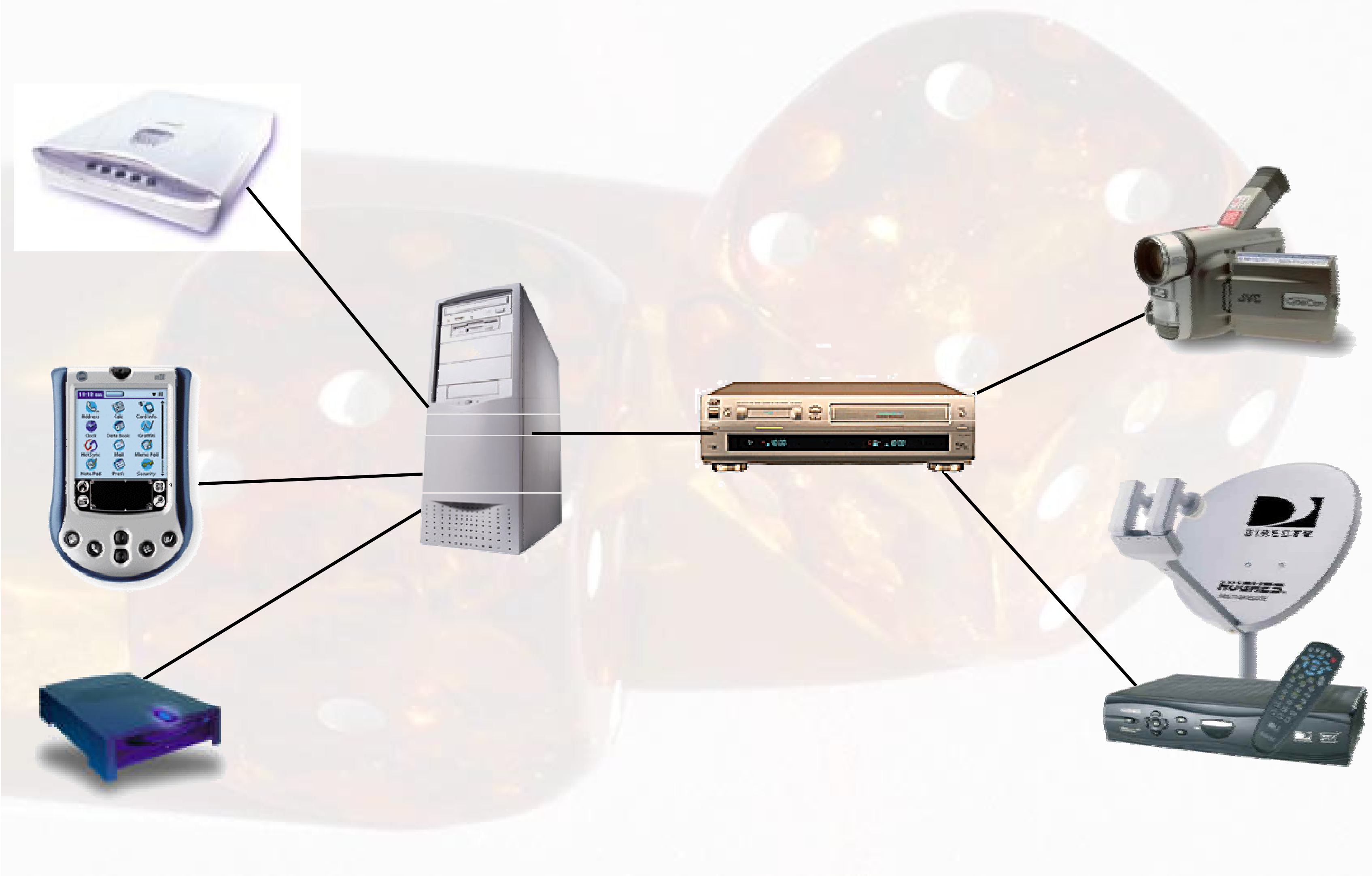


# Case Study: FireWire Protocol

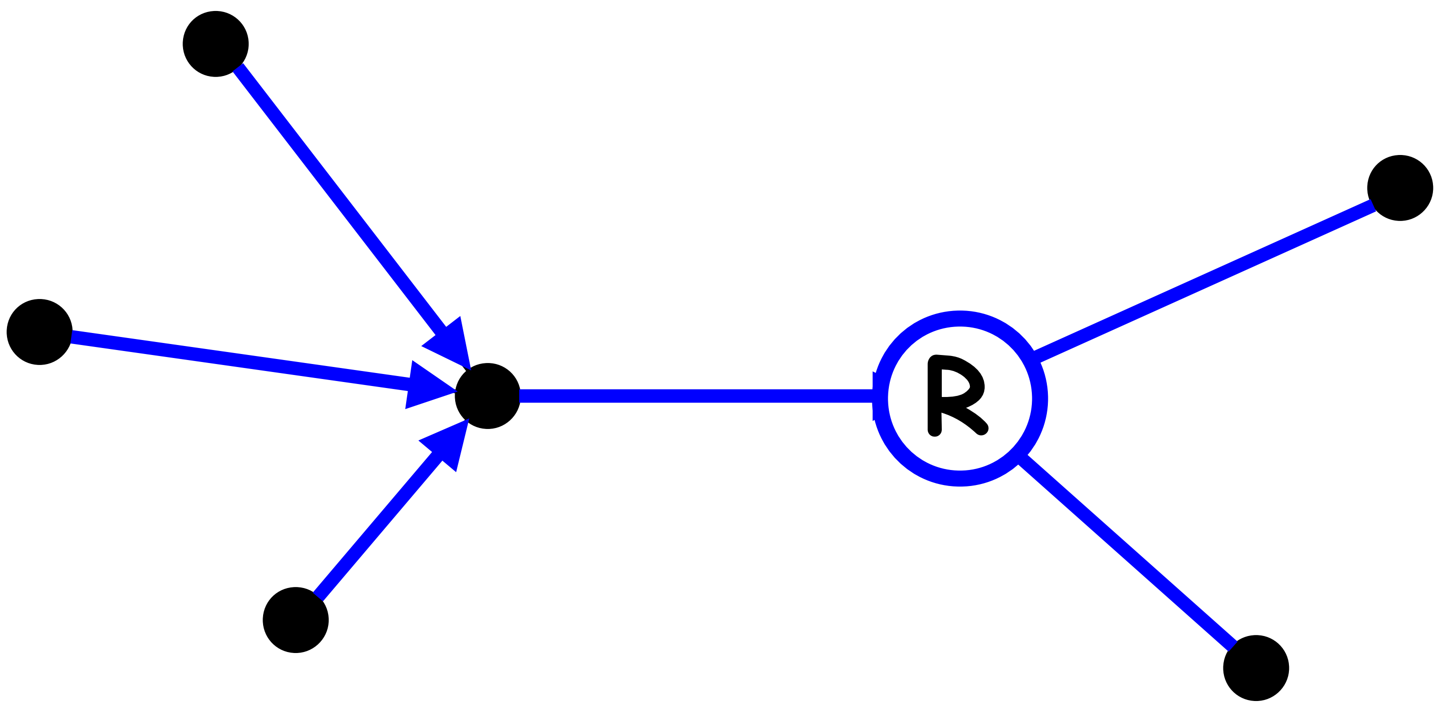
- FireWire (IEEE 1394)
  - one of fastest standards, high data rate
  - multimedia data
  - originally by Apple, mid-90s
  - winner of 2001 PrimeTime Emmy Engineering Award
  - no requirement for a single PC (acyclic topology, not tree)
  - "plug and play"

- Initial configuration
  - involves leader election
  - symmetric, distributed protocol
  - uses electronic coin tossing and timing delays: PTA model

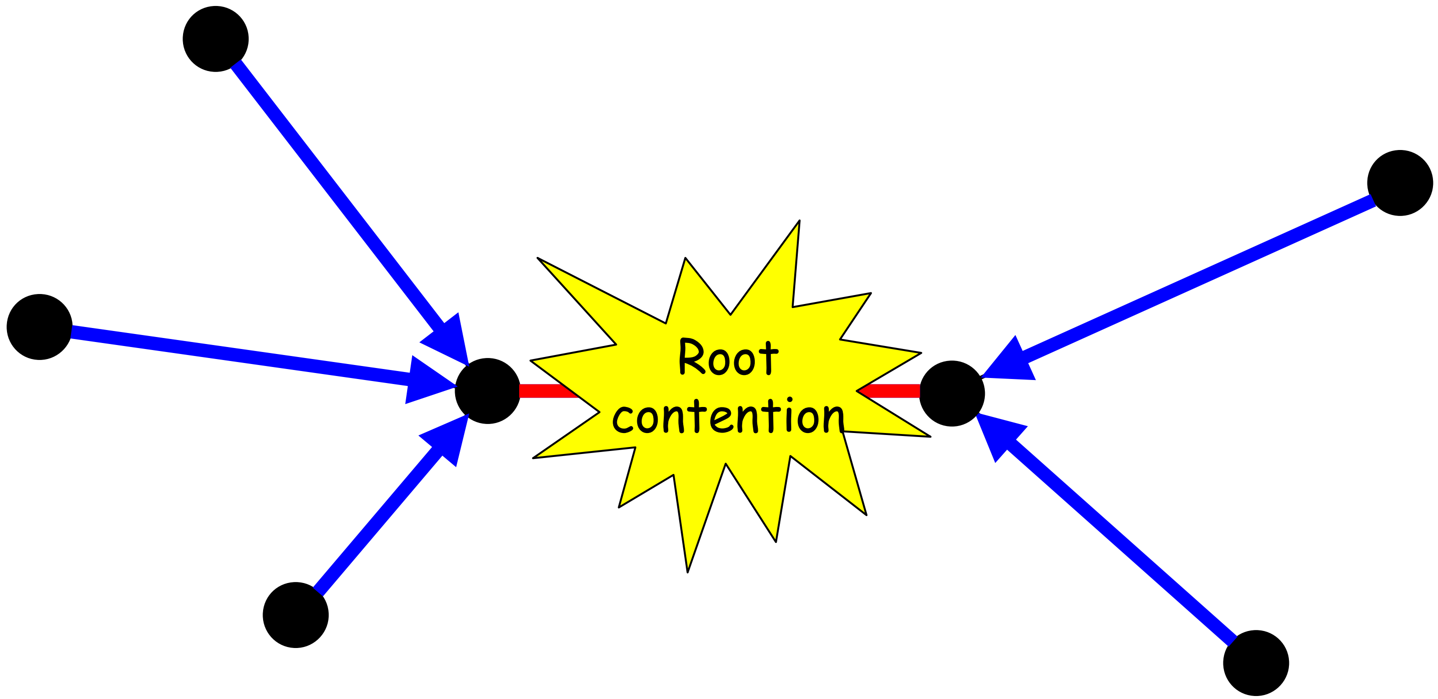
# Typical FireWire Configuration



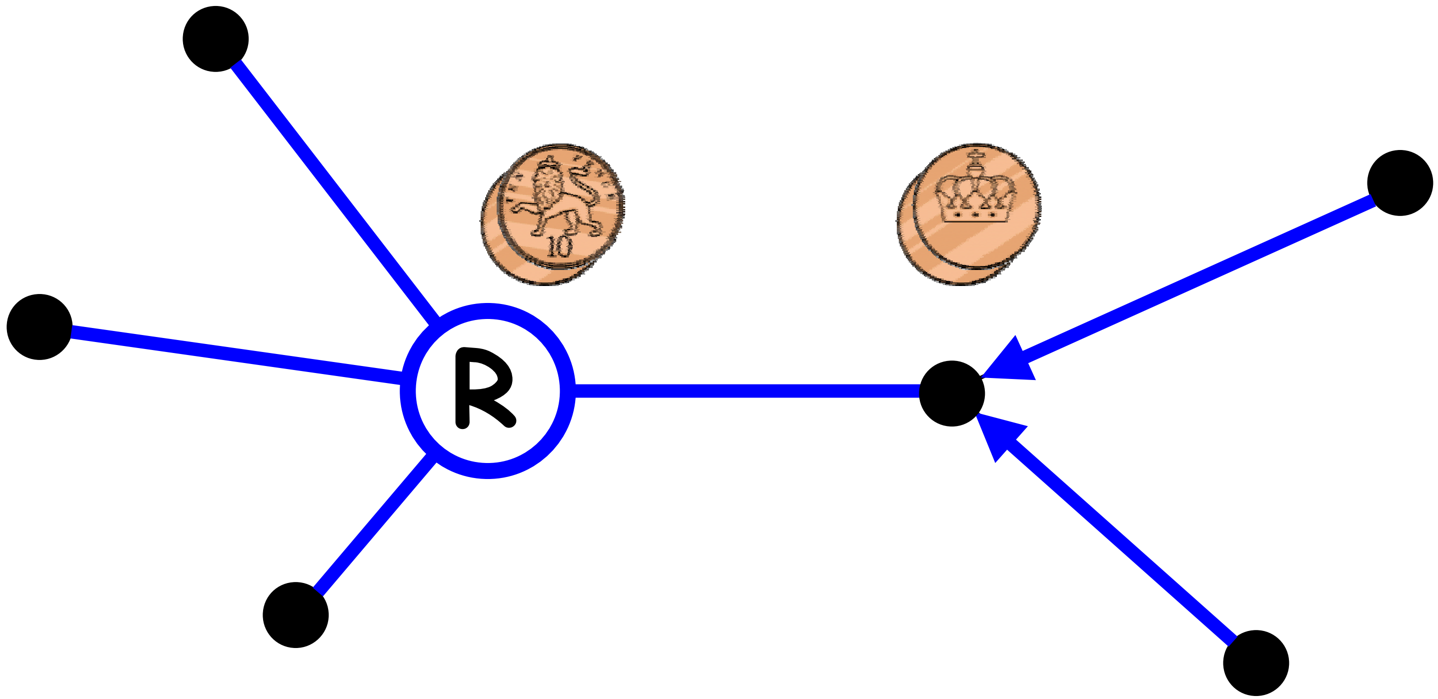
# FireWire Initial Configuration



# FireWire Root Contention



# FireWire Root Contention



# FireWire Analysis

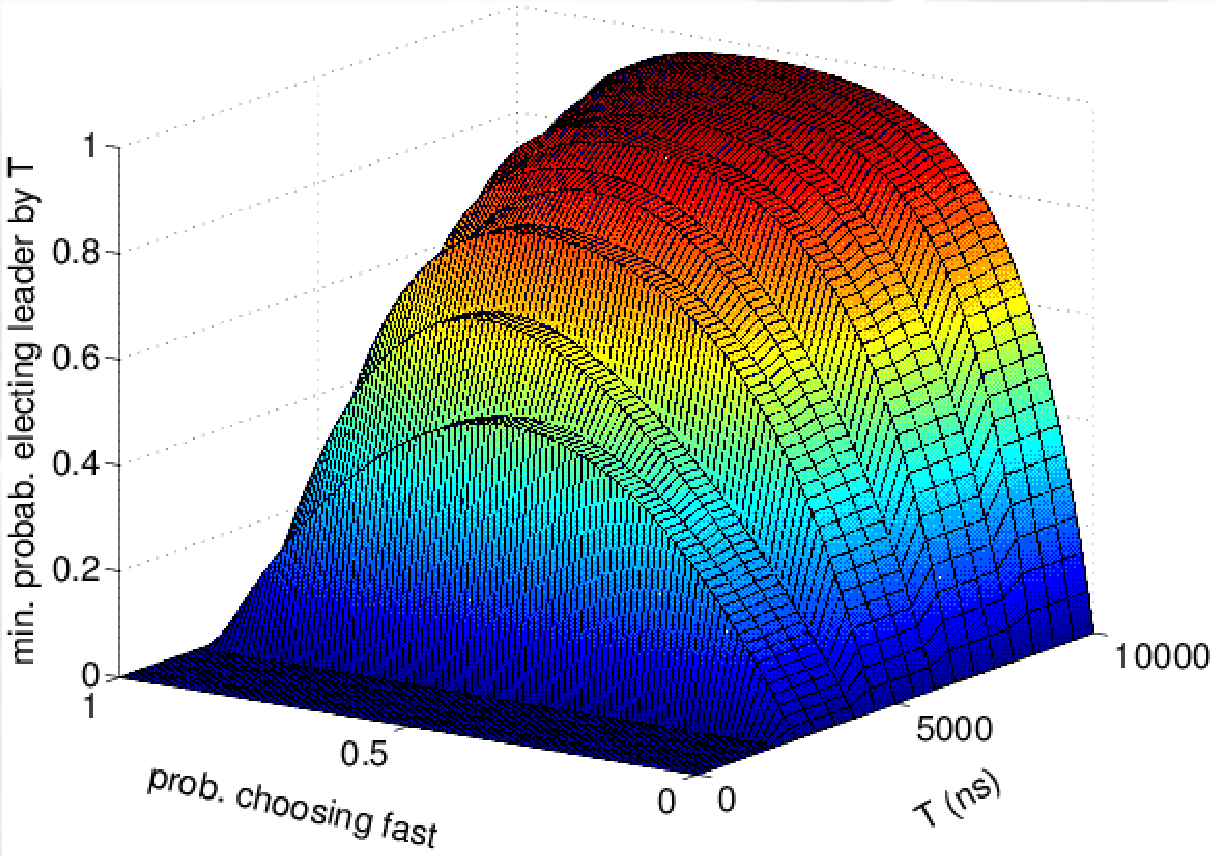
- Real-time properties
  - analysed by Vandraager and Stoelinga
  - used the UPPAAL model checker
  - shown correct wires longer than standard

- Probabilistic analysis
  - used UPPAAL & PRISM model checkers [KNS03, DNK02]
  - timing delays taken from standard
  - established that root contention resolved with probability 1
  - also considered expected time to root contention
  - a peculiarity found… (conjectured by Stoelinga)

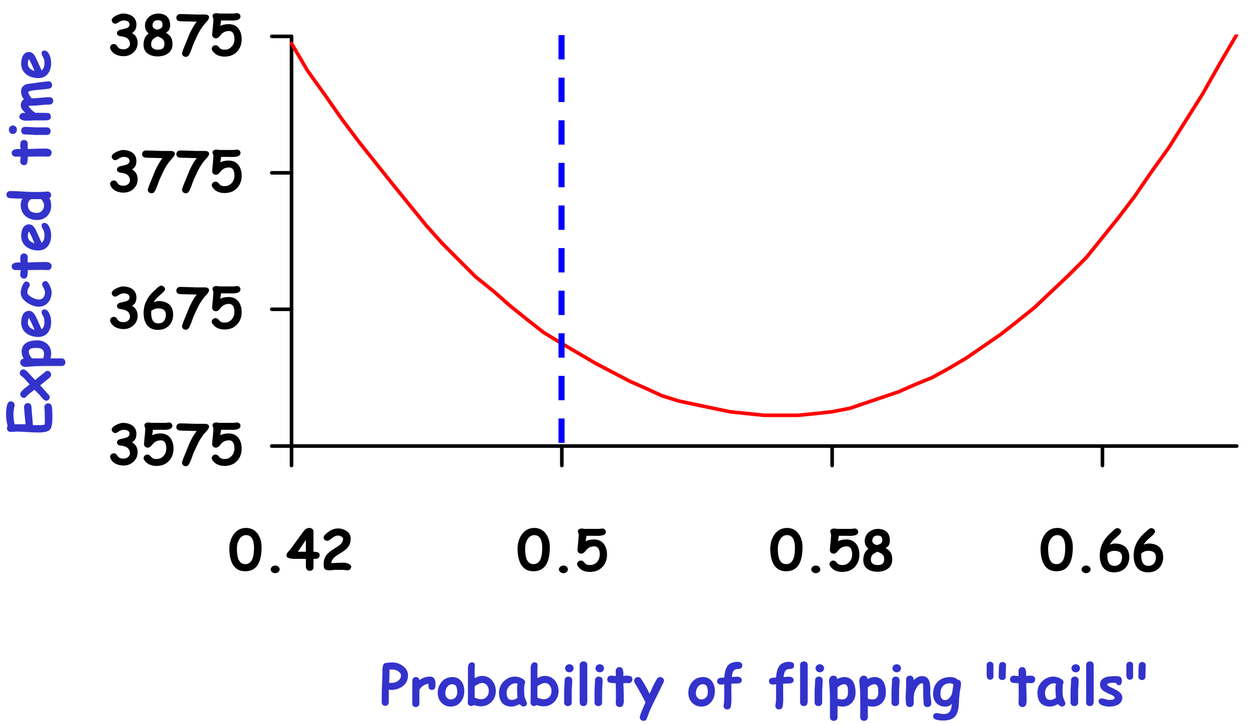
- Further analyses at various levels of abstraction, see special issue on FireWire

Formal Aspects of Computing (2003) 14: 295-318

# FireWire: Analysis Results



# Unfair coin gives advantage!



# Successes so far

- Fully automatic, no expert knowledge needed for
  - Probabilistic reachability and temporal logic properties
  - Expected time/cost
- Tangible results!
  - 5 cases of "unusual behaviour" found, ca 20 case studies
  - Greater level of detail, may expose obscure dependencies
- PRISM tool robust
  - Simple model description language
  - Broad class of models
  - Large, realistic models often possible
  - Flexible property language
  - Choice of engines

# But…

- Models monolithic and finite-state only
  - Emphasis on efficiency
  - No decomposition, abstraction
  - No data reduction
- State-space explosion has not gone away…
  - Heuristics for MTBDDs/BDDs sometimes fail
  - Parallelise? Disk-based?
- Limited expressiveness
  - Only PCTL plus extensions
  - Only exponential distributions
  - No direct support for PTAs
  - No continuous space models
  - No mobility

# Challenges for future

- Exploiting structure
  - Abstraction, data/equivalence quotient, (de)compositionality…
  - Parametric probabilistic verification?
- Proof assistant for probabilistic verification?
- Approximation methods?
- Efficient methods for continuous models
  - Continuous PTAs? Continuous time MDPs? LMPs?
- More expressive specifications
  - Probabilistic LTL/PCTL*/mu-calculus?
- Real software, not models!

- More applications
  - Nano-designs
  - Quantum cryptographic protocols
  - Mobile ad hoc network protocols

# Collaborators, contributors – thanks!

Rajeev Alur, Christel Baier, Stefano Cattani, Ed Clarke, Sadie Creese, Pedro D'Argenio, Conrado Daws, Luca de Alfaro, Amani El-Rayes, Stephen Gilmore, Michael Goldsmith, Rajeesh Gupta, Vicky Hartonas-Garmhausen, Boudewijn Haverkort, Holger Hermanns, Ulrich Herzog, Andrew Hinton, Joe Hurd, Michael Huth, Jane Hillston, Bertrand Jeannet, Joost-Pieter Katoen, Kim Larsen, Annabelle McIver, Rashid Mehmood, Carroll Morgan, Gethin Norman, Colin O'Halloran, Antonio Pacheco, Prakash Panangaden, Dave Parker, Sylvain Peyronnet, Mark Ryan, Roberto Segala, Vitaly Shmatikov, Sandeep Shukla, Markus Siegle, Jeremy Sproston, Moshe Vardi, Fuzhi Wang, Irfan Zakiuddin

EPSRC QinetiQ BRITISH COUNCIL THE UNIVERSITY OF BIRMINGHAM

# PRISM Contributors