A formal analysis of Bluetooth device discovery.

Marie Duflot, Marta Kwiatkowska, Gethin Norman and Dave Parker

Université Paris XII - University of Birmingham

Outline

- Motivation
- Bluetooth Protocol
- Probabilistic model checking and PRISM
- Modelling
- Results

Motivation

- Bluetooth
  - ➤ increasingly common wireless communication protocol
  - ➤ performance is particularly important in low-power setting
  - ➤ performance is the result of a non-trivial interaction between devices
    - ➜ formal verification desirable
  - ➤ protocol is randomised
    - ➜ need probabilistic formal verification

Bluetooth protocol

Bluetooth overview

- short-range low-power wireless protocol
- frequency hopping over 79/32 frequencies

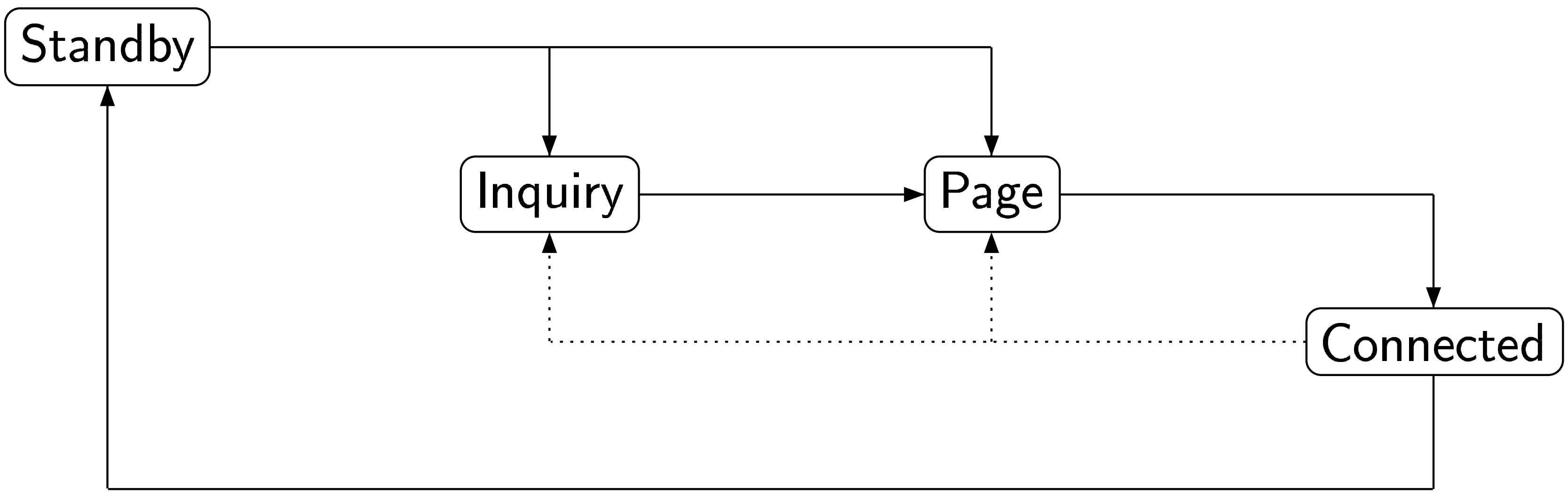
Bluetooth overview

- short-range low-power wireless protocol
- frequency hopping over 79/32 frequencies
  - ➤ need to form piconets
    - – processes know when to send/receive
    - – processes know the hopping frequency
    - – master-slave roles
  - ➤ no communication before initialisation

Bluetooth overview

- short-range low-power wireless protocol
- frequency hopping over 79/32 frequencies
  - ➤ need to form piconets
    - processes know when to send/receive
    - processes know the hopping frequency
    - master-slave roles
  - ➤ no communication before initialisation
- First mandatory step: device discovery

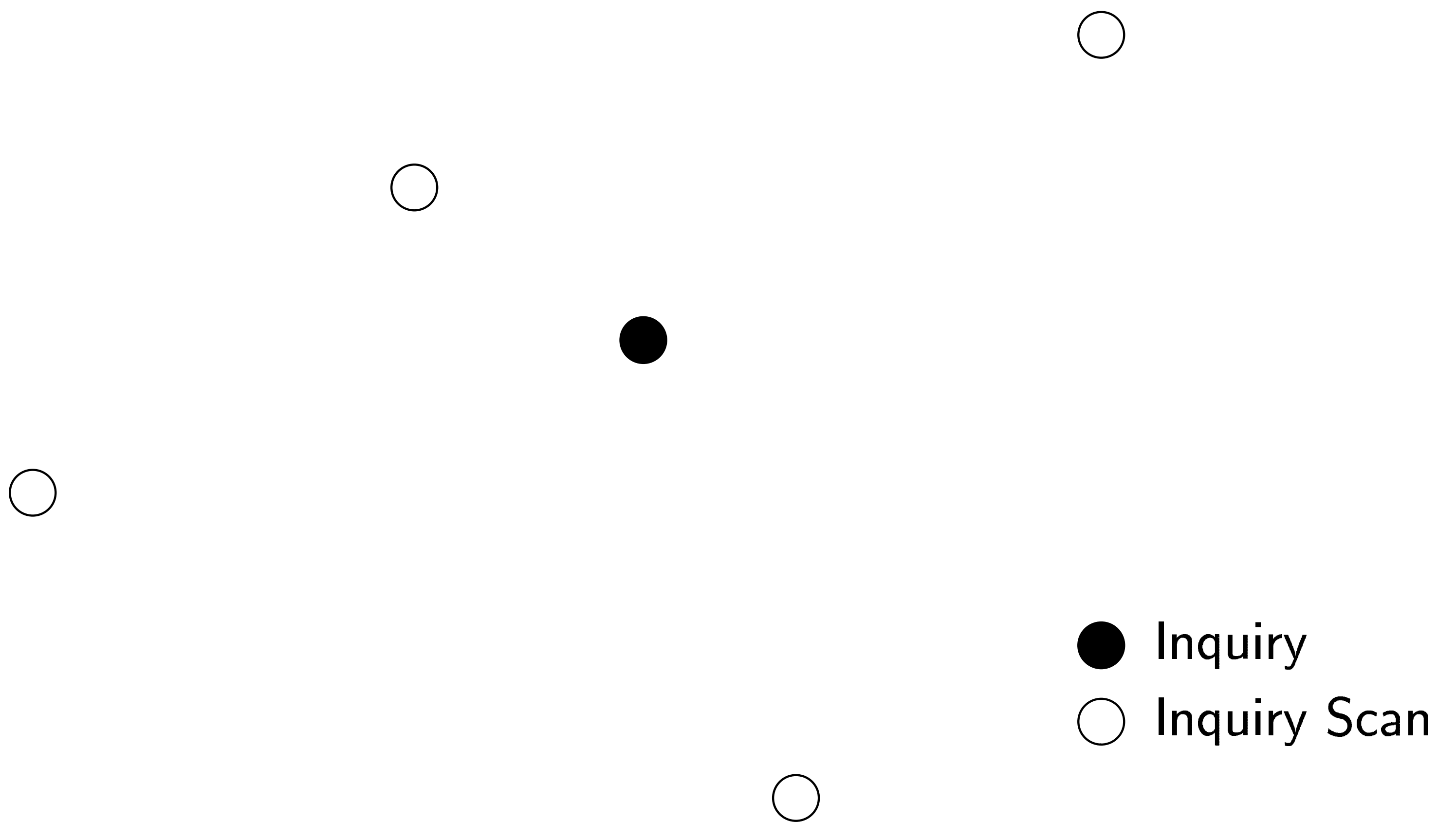
States of a Bluetooth device



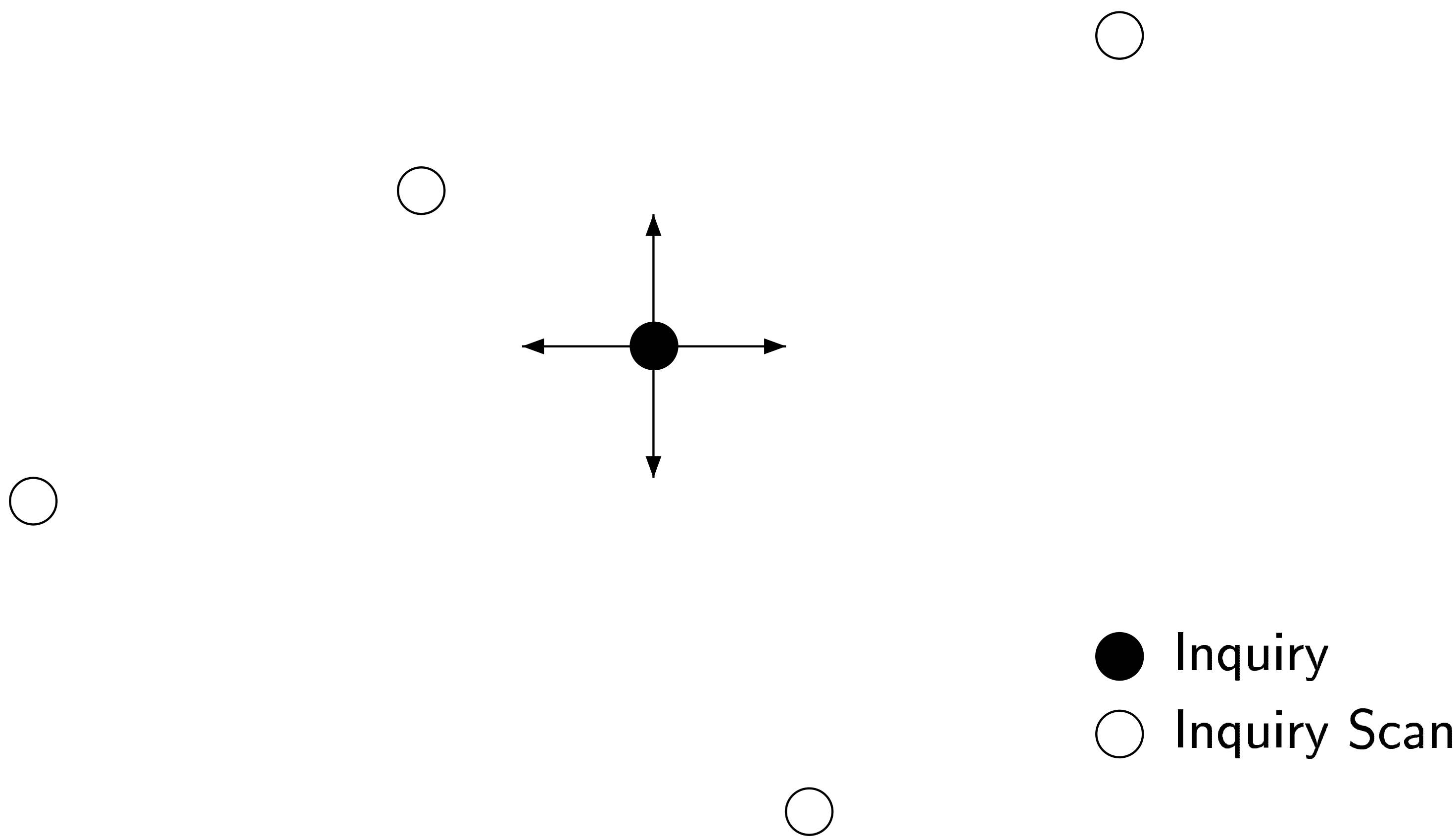
Standby: default operational state

Connected: device ready to communicate in a piconet

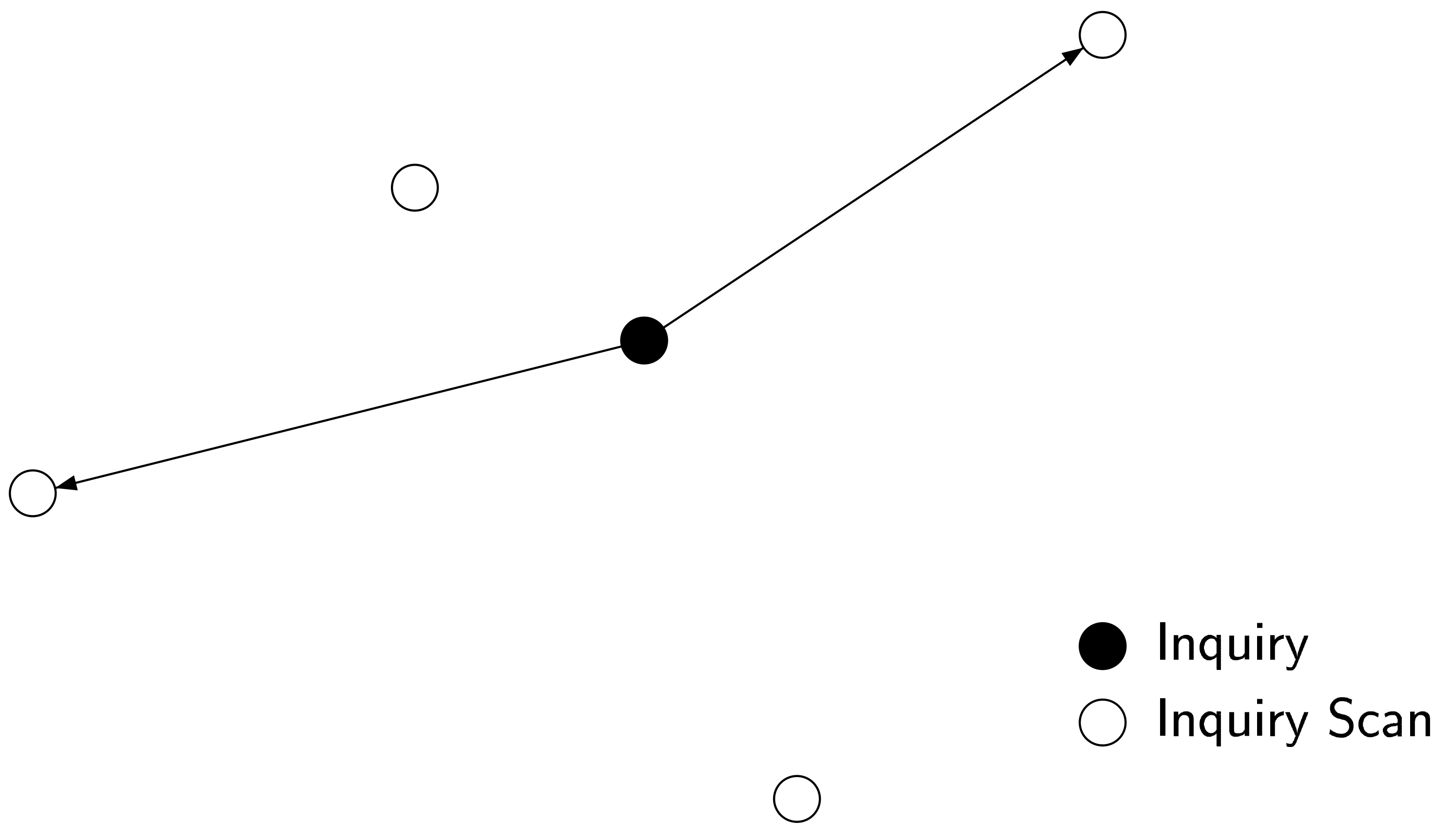
Inquiry (version 1.2)



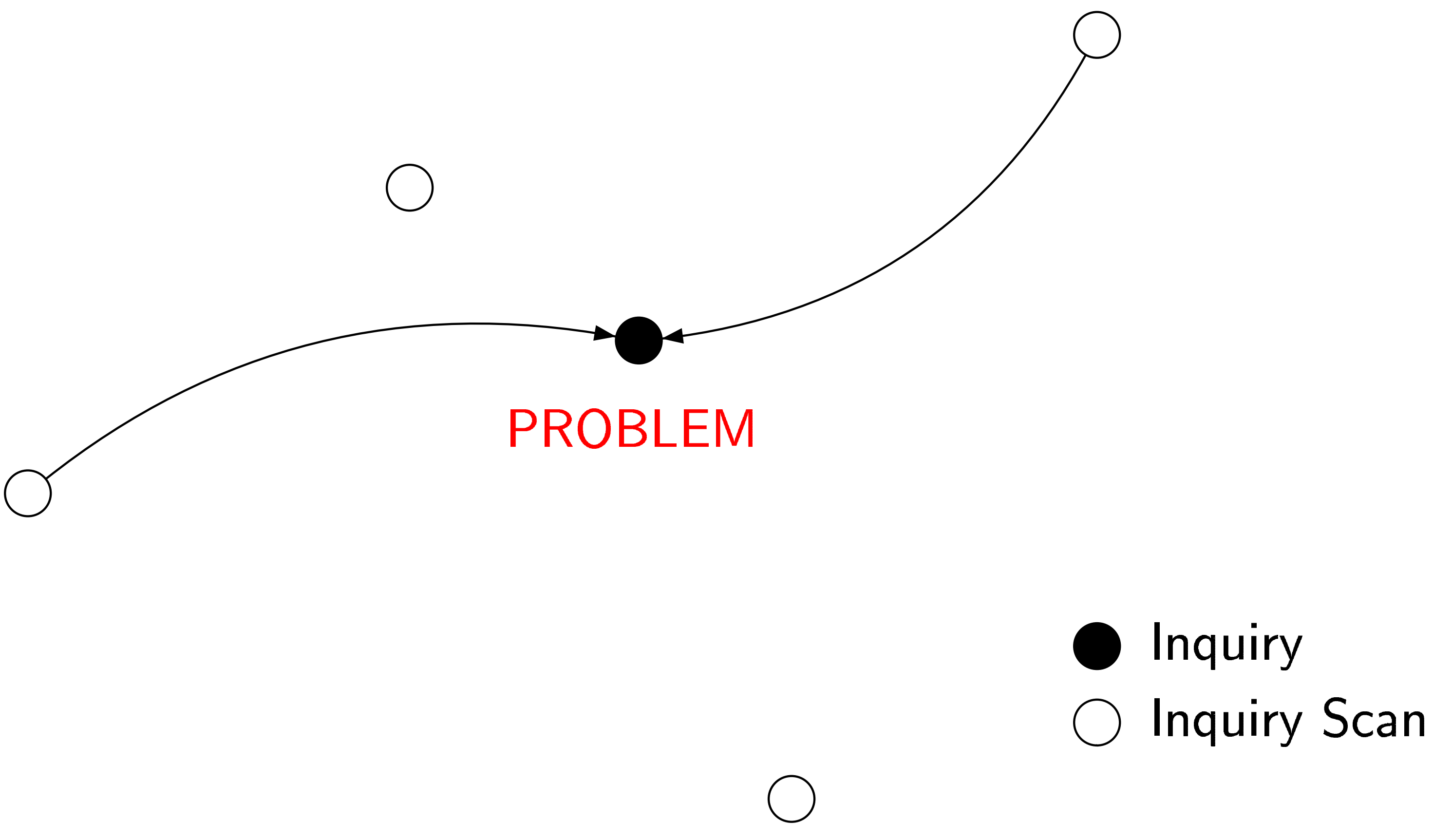
Inquiry (version 1.2)



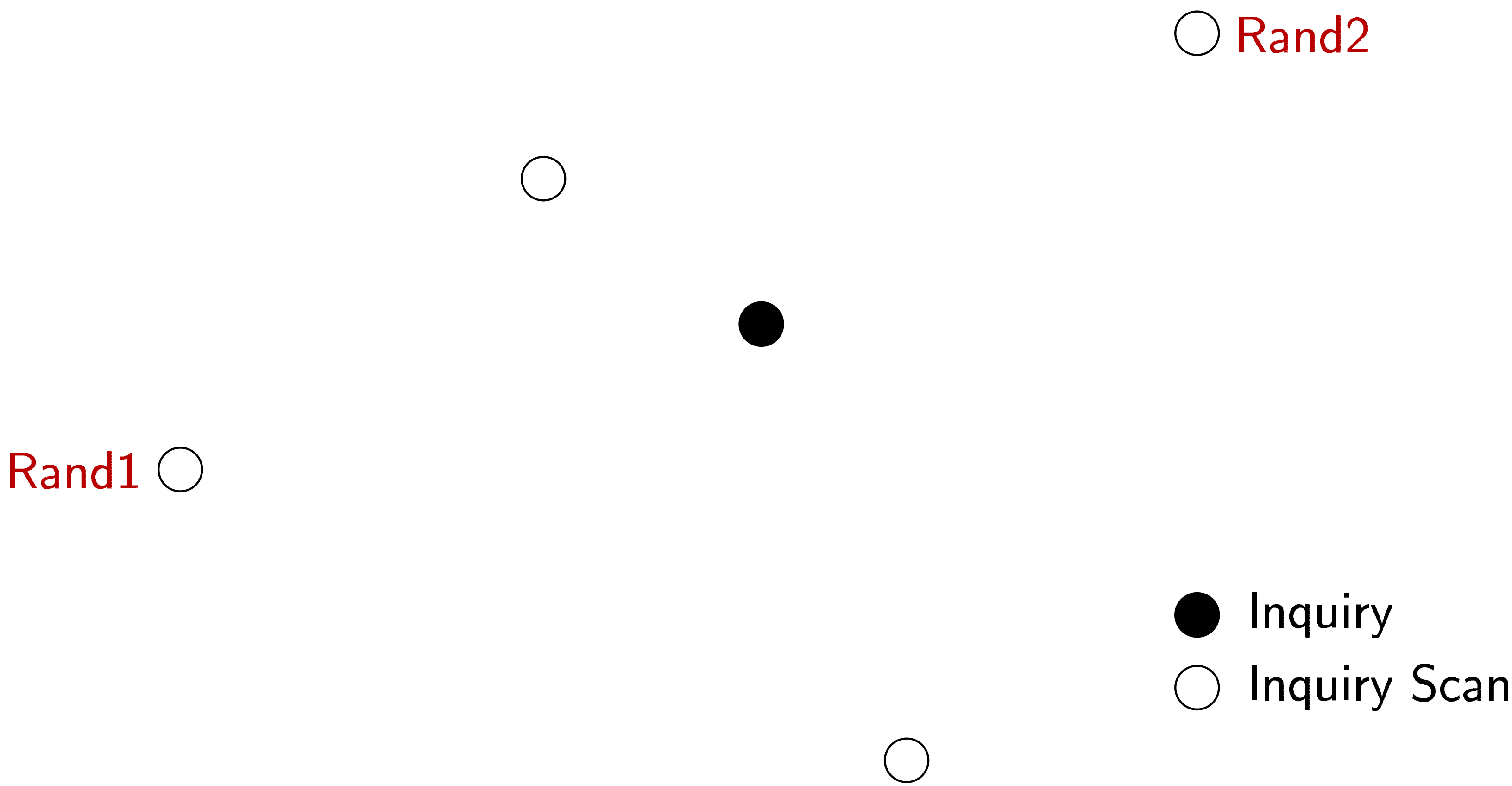
Inquiry (version 1.2)



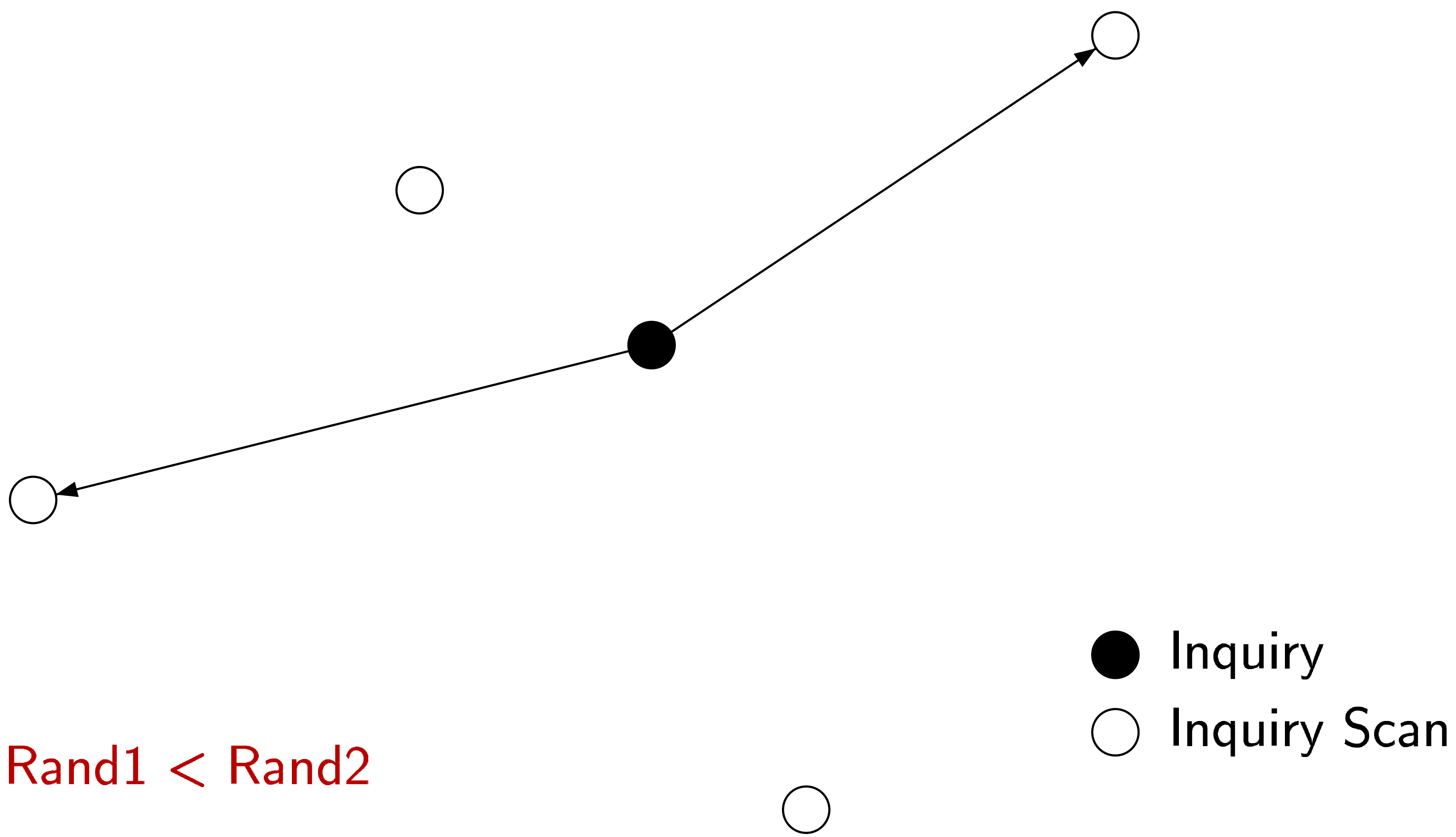
Inquiry (version 1.2)



Inquiry (version 1.2)

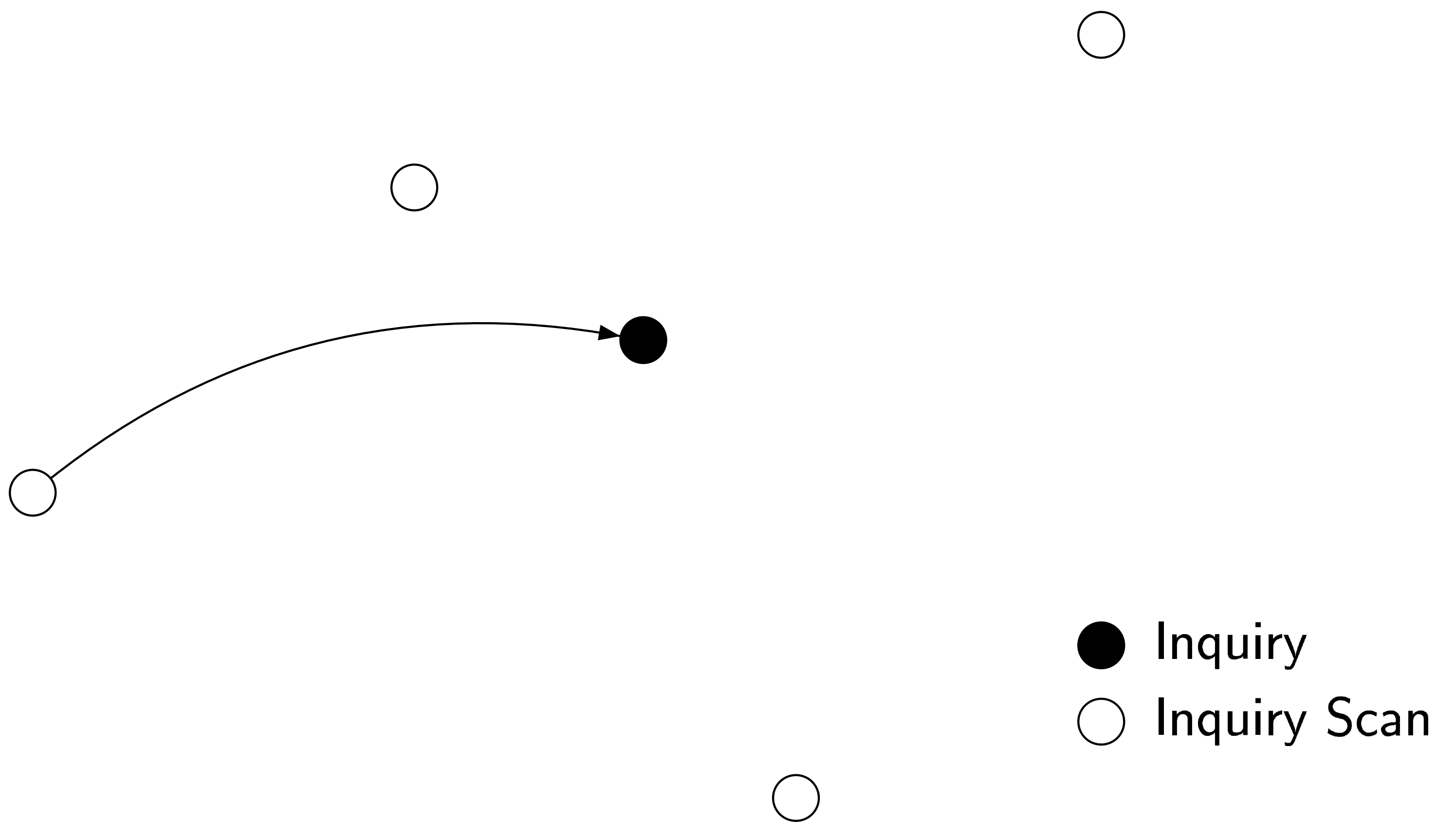


Inquiry (version 1.2)

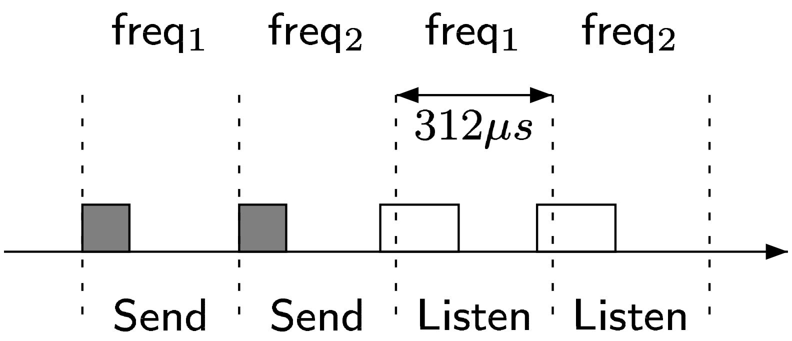


Rand1 < Rand2

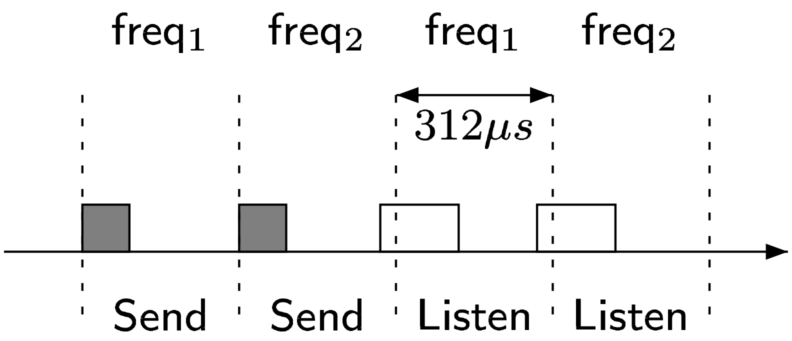
Inquiry (version 1.2)



The sender

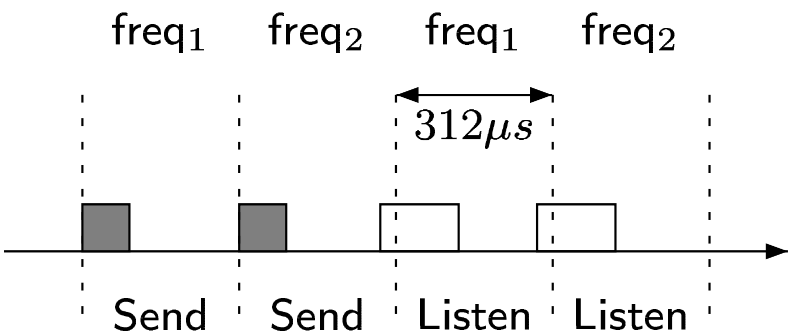


The sender



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

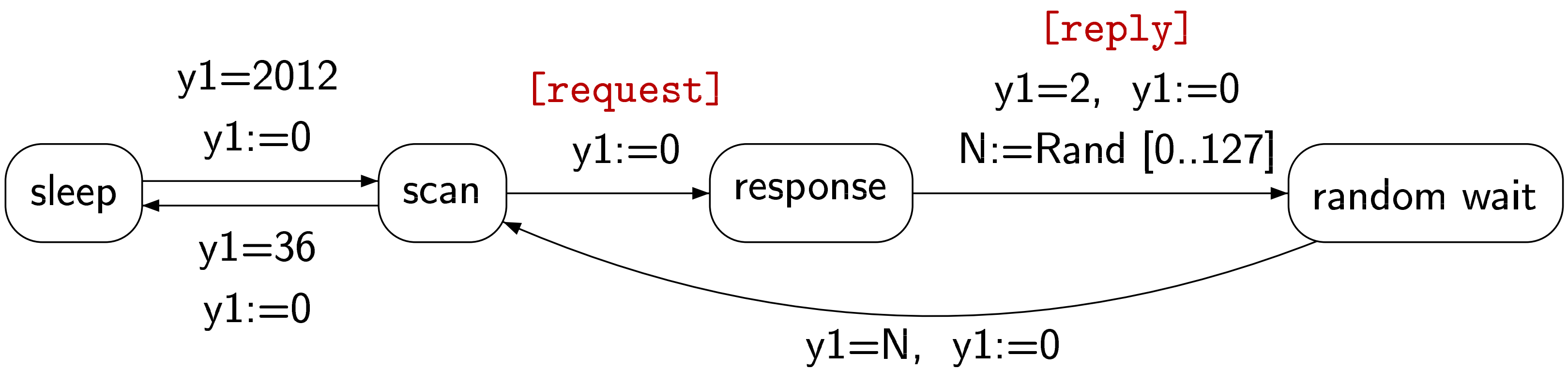
The sender



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

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

The receiver



- [request]: message sent by the sender
- [reply]: message sent by the receiver
- need to compute the frequency at which the receiver is listening (phase)
- phase increased by one each time the receiver replies

Probabilistic model checking and PRISM

Probabilistic model checking

- automatic formal verification of systems which exhibit stochastic behaviour
- conventional model checking:
  - model + property $\longrightarrow$ yes/no
  - graph analysis
- probabilistic model checking:
  - probabilistic model + probabilistic property $\longrightarrow$ yes/no/probability
  - graph analysis, numerical computation

Probabilistic models

- discrete-time Markov chains (DTMCs)
  - ➤ discrete time/probabilities
- continuous-time Markov chains (CTMCs)
  - ➤ real time (exponential distributions)
- Markov decision processes (MDPs)
  - ➤ discrete time/probabilities + nondeterminism

Probabilistic specifications

- PCTL/CSL - probabilistic extensions of CTL
- CTL: universal/existential quantification over paths
  - ➤ $AF\,success$, $EF\,success$
- PCTL: probabilistic quantification over paths
  - ➤ $\mathrm{P}_{\geq 0.98}(F\,success)$
- Extension: Query actual probability values
  - ➤ $\mathrm{P}_{=?}(F\,success)$
- Extension: Add rewards (or costs) to states/transitions of model...
  - ➤ $\mathrm{R}_{=?}(F\,success)$
  - ➤ expected reward cumulated to reach $success$

PRISM

- PRISM: PRobabilIstic Symbolic Model checker
- support for DTMCs/CTMCs/MDPs and PCTL/CSL
- high-level modelling language
  - ➤ based on Reactive Modules [Alur/Henzinger]
- wide range of existing case studies: randomised distributed algorithms, probabilistic security protocols, probabilistic communication protocols, etc.
- freely available: www.cs.bham.ac.uk/~dxp/prism

Modelling the protocol

Modelling formalism

- randomised back-off
  - ➤ need probabilistic model
- discrete time slots (negligible clock drift)
- no nondeterminism
  - ➤ no nondeterministic choice within a device
  - ➤ full synchronisation between devices

➜ discrete-time Markov Chains (DTMCs)

PRISM code: receiver's frequency

```
module frequency1

  z1 : [1..phase];// clock for phase
  f1 : [1..16]; // frequency of receiver
  o1 : [0..1]; // offset of receiver

  // update frequency (1 slot passes)
  [time] z1<phase -> (z1'=z1+1);
  [time] z1=phase -> (z1'=1) & (f1'=f1<16?f1+1:1)& (o1'=f1<16?o1:1-o1);
  // update frequency: something is sent by the receiver
  [reply] true -> (f1'=(f1<16)?f1+1:1) & (o1'=(f1<16)?o1:1-o1);

endmodule
```

State space explosion

- Sender changes state every time slot
- Receiver can wait for 2012 time slots without changing state
- 2 trains of 16 frequencies
- The trains change with time
- A train is repeated 256 times before switching
- The phase changes every 4096 slots
  - ➤ A Huge model
  - ➤ Too many possible initial states

➜ Need to use abstractions

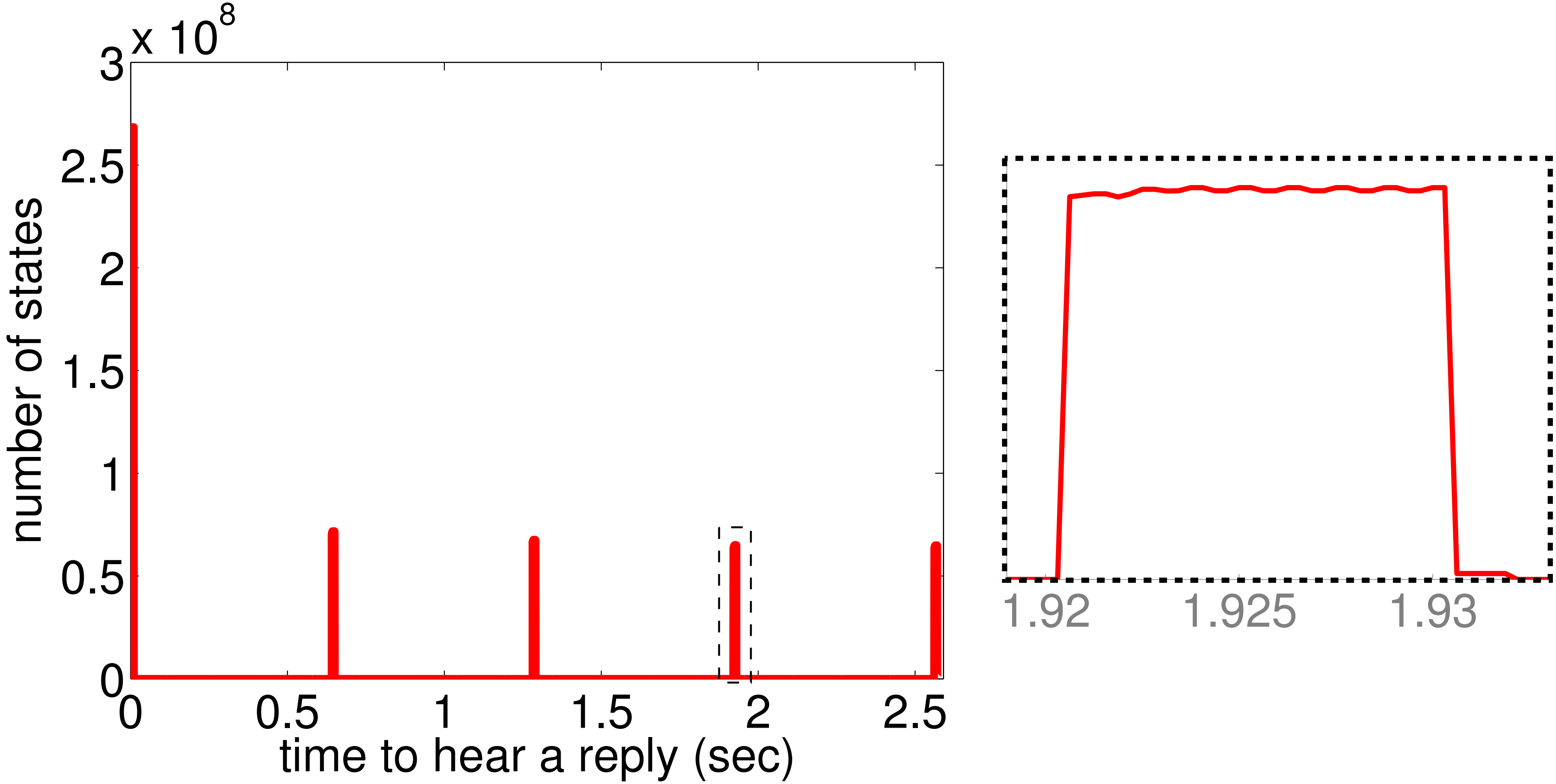
Verification and results

Expected time for one reply

- Big models $\rightarrow$ symbolic implementation (MTBDDs)
- Initial states split into 32 classes (possible initial frequencies)
- 32 models of around 3 thousand million states each
- 55-57 seconds to build one model
- 1-2 seconds to check the property

N	0	1	2	3	4
p	0.5003	0.6335	0.7591	0.8797	1

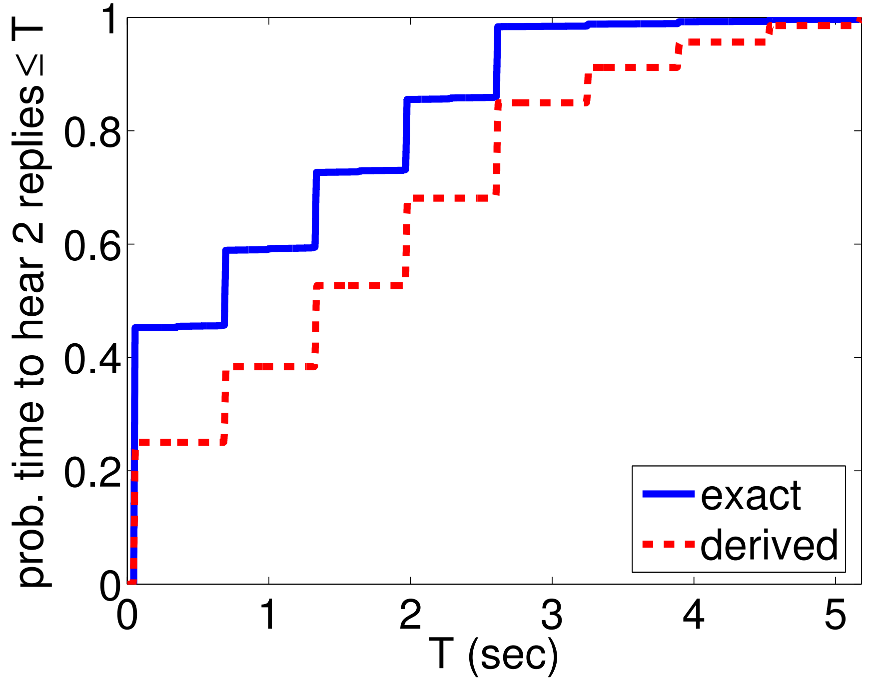
The graph for one reply



Expected time for two replies

- 32 models of 56 thousand million states each
- Time to build a model: 80 mins
- Time to verify the property : 165 mins
- Maximum expected time: 16565 slots (5 seconds)

The graph for two replies



- Need to reallly build the model for two replies
- Approximation via convolution is incorrect

Comparison with version 1.1

- In version 1.1, the receiver waits before sending a reply
- He replies to every second message.
- Expected time to receive a reply is longer

K	$n = 1$	$n = 1$ (v.1.1)	$n = 2$	$n = 2$ (derived)
0	0.500305	0.461240	0.455379	0.250305
1	0.633575	0.596265	0.590829	0.383657
2	0.759062	0.731585	0.728684	0.526981
3	0.879674	0.857913	0.855329	0.681114
4	1	0.984295	0.984218	0.849408
5	1	0.988269	0.988294	0.911750
6	1	0.992398	0.992514	0.956496
7	1	0.996294	0.996519	0.985521
8	1	1	1	1

Conclusions

- Summary
  - A formal analysis of Bluetooth device discovery
  - Quantitative quality of service results on a non simplified model
  - Model-checking *vs* simulation
- Future work
  - Further abstractions and symmetry.
  - Combine model checking and simulation.