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
- Inquiry Scan
Inquiry (version 1.2)
The sender

freq_1  freq_2  freq_1  freq_2

Send  Send  Listen  Listen

312 µs
The sender

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

freq1 freq2 freq1 freq2

312μs

Send Send Listen Listen

freq = \left[ CLK_{16-12} + k \right]
+ \left( CLK_{4-2,0} - CLK_{16-12} \right) \mod 16 \mod 32
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 $\rightarrow$ yes/no
  - graph analysis
- probabilistic model checking:
  - probabilistic model + probabilistic property $\rightarrow$ 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
  - $P_{\geq 0.98}(F_{success})$
- Extension: Query actual probability values
  - $P_{\geq?}(F_{success})$
- Extension: Add rewards (or costs) to states/transitions of model...
  - $R_{\geq?}(F_{success})$
  - 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/~dwp/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)
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 → 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

<table>
<thead>
<tr>
<th>N</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>0.5003</td>
<td>0.6335</td>
<td>0.7591</td>
<td>0.8797</td>
<td>1</td>
</tr>
</tbody>
</table>
The graph for one reply

A formal analysis of Bluetooth device discovery.
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 really 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

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