Industrial Application of Coloured Petri Nets for Protocol Verification

Lars M. Kristensen
Department of Computer Engineering
Bergen University College, NORWAY
Email: lmkr@hib.no /Web: www.hib.no/ansatte/lmkr
Practical Applications

- CPNs and state space methods have been widely used for protocol verification purposes:
  - Danfoss Flowmeter Systems.
  - Bang & Olufsen Beolink System.
  - Ericsson Edge Router Discovery Protocol.
  - Several Internet protocols (e.g., WAP, IOTP, TCP, DCCP, SIP, DYMO).
  - ...

- For a comprehensive list of examples, see: [http://www.cs.au.dk/CPnets/intro/industrial.shtml](http://www.cs.au.dk/CPnets/intro/industrial.shtml)
Overview

- **Two examples of industrial application of CPN technology for protocol verification.**

- **Specification and Validation of an Edge Router Discovery Protocol for Mobile Ad-hoc Networks:**
  

- **Formal Specification and Validation of Secure Connection Establishment in a Generic Access Network Scenario:**
  
Specification and Validation of an Edge Router Discovery Protocol for Mobile Ad Hoc Networks
Project Aims and Setup

- **Project context:**
  - Development of the **Edge Router Discovery Protocol (ERDP)** for MANETs based on the IPv6 NDP Protocol.
  - Apply of Coloured Petri Nets (CPNs) and CPN Tools in the development of protocol software.
  - The software engineers were given a 6-hours course on CPN modelling.

- **Application of CPN technology:**
  - A CPN model was constructed constituting a formal specification of the ERDP protocol.
  - State space exploration was applied to conduct a formal verification of key properties of ERDP.
  - Modelling and verification helped in identifying several omissions and errors in the design.
Edge Router Discovery Protocol

- Protocol for IPv6 prefix configuration executed between edge routers and gateways:

![Diagram showing the Edge Router Discovery Protocol network topology](image)

- Internet
- Edge router
- Gateway
- Mobile ad-hoc network
Configuration of a gateway

Currently no assigned prefixes

Unicast router solicitation (RS)

Update prefixes

Periodical multi-cast of unsolicited router advertisement (RA)

Prefixes have a limited lifetime – must be refreshed – otherwise they will expire

The wireless link is unreliable

New prefix

Unicast solicited router advertisement (RA)
The Modelling Phase

- CPN modelling applied for specification of the protocol software design:
  - First a conventional natural language specification was developed by the protocol software engineers.
  - Protocol engineers was given a 6-hour course on CPNs.
  - Next a CPN model reflecting the specification was developed.

- The ERDP protocol and the CPN model was then developed in an iterative process:
  - CPN model discussed and reviewed in each iteration.
  - CPN model used as a basis for discussion of protocol design.
  - Interactive simulation used for detailed investigations of the protocol software.
Module Hierarchy

Abstract view

Gateway
- ProcessUnsolicitedRA
- ReceiveSolicitedRA
- SendRouterSolicitation
- GWDiscardPrefixes

EdgeRouter
- SendUnsolicitedRA
- ERDiscardPrefixes
- ProcessRS

GW_ER_Link
- NoUnusedPrefix
- AssignNewPrefix
ERDP Top-level Module

Three substitution transitions

Four packet buffers
Results from Modelling

- Several software design issues and errors were identified in the modelling phase:

<table>
<thead>
<tr>
<th>Category</th>
<th>Review 1</th>
<th>Review 2</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incompleteness and ambiguity in the ERDP specification</td>
<td>3</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>Errors in the protocol</td>
<td>2</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>Simplifications of the protocol</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Additions</td>
<td>4</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Total</td>
<td>11</td>
<td>13</td>
<td>24</td>
</tr>
</tbody>
</table>

- Approximately 70 person-hours were used on CPN modelling and reviews.
State Space Exploration

- State space exploration was pursued after the three iterations of modelling.

- The first step was to obtain a finite state space:
  - The ERDP CPN model can have an arbitrary number of tokens on the packet buffers.
  - An upper integer bound of 1 was imposed on each of the packet buffers (GWIn, GWOOut, ERIn, EROut).
  - This also prevents overtaking among the packets transmitted across the wireless link.
  - The number of tokens simultaneously on the four packet buffers was limited to 2.
Verification of ERDP

- Key property of the ERDP protocol:

  From any state with a non-configured prefix $P$ it is possible to reach a state where $P$ is consistently configured.

- Investigated using state space exploration starting from the simplest possible configuration.
Simplest Configuration
[one prefix, no loss, no expiration]

- **State space:** 46 nodes and 65 arcs.
- A single **dead marking**.

- **Visual inspection** showed that the dead marking is an inconsistently configured state:
  - The edge router has assigned a prefix to the gateway.
  - BUT, the gateway is not configured with the prefix.

- **The error-trace was visualised by means of a message sequence chart.**

- **Demonstrates that errors tend to manifest themselves even in simple configurations.**
Error trace MSC

- The edge router sends two unsolicited RAs.

- The first one gets through and we obtain a **consistent configuration** with prefix P1.

- When the second reaches the edge router there are no unassigned prefixes available.

- A Solicited RA with the empty list of prefixes is sent.

- The gateway updates its prefixes to be the empty list.
Revised configuration
[One prefix, no loss, no expiration]

- The protocol was revised such that the edge router always replies with the list of all currently assigned prefixes.

- **State space:** 34 nodes and 49 arcs.

- **No dead markings and 11 home markings** (constituting a single terminal SCC).

- Inspection showed that all home markings are consistently configured with the prefix.
  - It is **always possible** to reach a **consistently configured state** for the prefix.
  - When such a state has been reached, the protocol entities will remain consistently configured.
Results from Verification

- The verification was conducted in three steps where assumptions were gradually removed.

  - **Step 1 [no packet loss and no expire of prefixes]**:
    - Synchronisation error between edge router and gateway.
    - The error was corrected and the key property was verified.

  - **Step 2 [packet loss on wireless link added]**:
    - Synchronisation error when certain unsolicited RAs was lost.
    - Livelock error in processing of router advertisement in gateway.
    - The errors were corrected and the key property was verified.

  - **Step 3 [expire of prefixes added]**:
    - Property verified: Consistent configuration always possible.
State Space Statistics

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>No loss/No expire</th>
<th>Loss/No expire</th>
<th>Loss/Expire</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>34</td>
<td>49</td>
<td>68</td>
<td>160</td>
</tr>
<tr>
<td>2</td>
<td>72</td>
<td>121</td>
<td>172</td>
<td>425</td>
</tr>
<tr>
<td>3</td>
<td>110</td>
<td>193</td>
<td>337</td>
<td>851</td>
</tr>
<tr>
<td>4</td>
<td>148</td>
<td>265</td>
<td>582</td>
<td>1,489</td>
</tr>
<tr>
<td>5</td>
<td>186</td>
<td>337</td>
<td>926</td>
<td>2,390</td>
</tr>
<tr>
<td>6</td>
<td>224</td>
<td>409</td>
<td>1,388</td>
<td>3,605</td>
</tr>
<tr>
<td>7</td>
<td>262</td>
<td>481</td>
<td>1,987</td>
<td>5,185</td>
</tr>
<tr>
<td>8</td>
<td>300</td>
<td>553</td>
<td>2,742</td>
<td>7,181</td>
</tr>
<tr>
<td>9</td>
<td>338</td>
<td>625</td>
<td>3,672</td>
<td>9,644</td>
</tr>
<tr>
<td>10</td>
<td>376</td>
<td>697</td>
<td>4,796</td>
<td>12,625</td>
</tr>
</tbody>
</table>

- When a state space had been generated, the verification of the key properties was be done in a few seconds.
Lessons Learned

- **Start state space exploration from the simplest possible configurations:**
  - Errors often manifest themselves in the simplest configurations and with the strongest assumptions.
  - The assumptions are then gradually lifted and larger configurations considered.

- **For the ERDP protocol state explosion was not a problem.**

- **The key properties could be verified for the number of prefixes envisioned in practice.**

- **Both modelling and state space exploration played a central role in validating the protocol.**
Conclusions from Project

- The construction the CPN model improved the **quality** of the ERDP design specification.
- **Non-trivial design errors** were identified and fixed in the course of modelling and verification.

- CPN and CPN Tools were powerful enough to specify and validate real-world protocol software.
- Approximately 100 person-hours over 4-months were used for modelling and verification.
Formal Specification and Validation of Secure Connection Establishment in a Generic Access Network Scenario
The GAN Architecture

- This subproject is concerned with the **Generic Access Network** (GAN) architecture.
- Currently being developed by the 3rd Generation Partnership Project [www.3gpp.org](http://www.3gpp.org).
- Supports access to telephone network services (e.g., messaging and voice calls) via IP networks:

![GAN Architecture Diagram]

Mobile Station  
Security Gateway  
GAN Controller  

**Step 2**: Establish GAN controller address
GAN at TietoEnator

- A specific instantiation of the GAN architecture:
  - Define the scope of the protocol software to be developed.
  - Specify detailed design and usage of the protocol software.
- Main purpose of the modelling was to specify the use of:
  - The Dynamic Host Configuration Protocol (DHCP) for IP address configuration of the mobile station.
  - The IP security (IPsec) protocols for encryption and authentication.
  - The use of the Internet Key Exchange (IKE) protocol for negotiation of IPsec parameters.
- Use simulation and state space analysis to validate the completeness and correctness of the GAN scenario.
CPN Model Overview

- A hierarchical CPN model consisting of 31 modules organised in four levels:
  - **Network nodes**: the mobile station, wireless router, security gateway(s), and the GAN controller(s).
  - **Protocol entities**: DHCP, IPsec, IKEv2, GAN signalling, and the Internet Protocol (IP) network layer.

- Developed in close interaction with TietoEnator protocol engineers over a period of 3 months.
- Initial CPN model constructed based on a textual description of the GAN Scenario considered.
- Protocols engineers did not have any previous knowledge of Coloured Petri Nets.
Mobile Station

- All network nodes structured similarly and reuses the IP and physical layer modules.

SPD: used by IPsec to specify which packets are allowed to be sent/received by the IP layer module.

RT: used in conjunction with encapsulation/decapsulation and the secure tunnels.

Adr: used for configuration of the IP layer module.
IP Layer Modelling
Mobile Station
GAN Layer Modelling

[Mobile Station]

- Specifies the steps in establishing a GAN connection.
- Explicit graphical representation of control flow and packet flow.
- The security policy database and routing table are accessed and updated in the individual steps.
- GAN layer of other network nodes are organised similarly.
Modelling Protocol Entities

IKE initiator [mobile station]

IKE responder [security gateway]
The modelling level at which building an **executable model** was powerful in making design issues explicit.
Simulation and MSCs

- Single-step simulation and message sequence charts (MSCs) were used for initial validation.
- Detailed inspection of control flow, packet flow, security policy databases, and routing tables.
- Conducted jointly with the protocol engineers at TietoEnator in two formal meetings.
- Enabled discussions and resulted in several further modifications to the CPN model.
Example MSC

- IKE phases of step 2 in GAN connection establishment.
- Generated directly from the CPN model using the BRITNeY animation tool.
- Presents the operation of the protocols in a form well-known to protocol engineers.
- Focus on message exchange between peer protocol entities.
State Space Analysis

- State space analysis was subsequently used to verify the GAN connection establishment.
- Key correctness criteria:
  
  * Always possible to reach a state where the GAN connection is properly established \([AG EF \varphi]\)*

- State space has 3,854 nodes and a single dead marking \(M\) which is also a home marking.
- \(M\) represents a state where the connection has been properly established.
Conclusions

- The construction of the CPN model:
  - Used to specify the specific instance of the GAN architecture to be implemented by TietoEnator.
  - Developed and reviewed in close co-operation with TietoEnator protocol engineers.
  - Spans multiple protocols and protocol layers which is a key characteristic of the GAN architecture.

- Benefits of the CPN model for development:
  - Useful in capturing the scope and initial design of the protocol software to be developed.
  - Useful in detailing and validating the message exchanges that were not explicit in the initial textual GAN specification.
  - A high degree of confidence in the design has been obtained.
Verification in Perspective

- Modelling and state space verification for system validation goes hand in hand:
  
  Ericsson Edge Router Discovery Protocol Project

<table>
<thead>
<tr>
<th>Category</th>
<th>Review 1</th>
<th>Review 2</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incompleteness and ambiguity in the ERDP specification</td>
<td>3</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>Errors in the protocol</td>
<td>2</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>Simplifications of the protocol</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Additions</td>
<td>4</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Total</td>
<td>11</td>
<td>13</td>
<td>24</td>
</tr>
</tbody>
</table>

- When pragmatically applied current methods can be used to obtain useful results on real systems:
  - Compact CPN modelling means that the full state space can usually be explored for the smallest configurations.
  - Advanced methods in many cases allow the system configurations occurring in practice to be verified.

Three subtle behavioural errors found