Protocol Verification and State Space Methods

Wojciech Penczek
Institute of Computer Science
Polish Academy of Sciences, Warsaw, Poland
Email: penczek@ipipan.waw.pl /Web: www.ipipan.waw.pl/~penczek/

Lars M. Kristensen
Department of Computer Engineering
Bergen University College, Bergen, Norway
Email: lmkr@hib.no /Web: www.hib.no/ansatte/lmkr
Communication Protocols

- Communication protocols play an increasingly important role in our everyday life:

- **Building blocks** governing the interaction between entities in data communication (Holzmann’91):
  - Service to be provided by the protocol.
  - Assumptions about the environment in which the protocol is executed.
  - Vocabulary of messages used to implement the protocol.
  - Encoding (format) of each message in the vocabulary.
  - Procedure rules guarding the processing of messages.
Protocol Engineering

- The development of protocols involves a number of activities [Liu’89]:

  - Service specification
    - Synthesis/design
      - Functional verification
      - Performance analysis
      - Conformance testing
    - Coding (automated)
      - Testing
  - Protocol specification
  - Protocol implementation

- It is important that protocols are working correctly from the very beginning.
- A key application domain for Petri nets, concurrency theory, and model checking technology.
Protocol Engineering Challenges

- The execution of a protocol may proceed in many different ways, e.g. depending on:
  - Whether messages are lost during transmission.
  - The scheduling of processes (protocol entities).
  - The time at which input is received from the environment.
- Protocols often exhibit complex behaviour and have an infinite number of possible executions:
  - It is easy for the protocol engineer to miss important interaction patterns during design.
  - This may lead to gaps or malfunctions in the protocol design.
  - Makes testing and debugging difficult.
Example: ERDP Protocol

- Protocol for gateway configuration in mobile ad-hoc network:
  - Combination of message loss and scheduling:
    - Inconsistent configuration.
    - Livelocks.
Specification of Protocols

- Based on the construction of **formal executable models** that can be analysed by **computer tools**: 

**Communication Protocol**

- Modelling is beneficial for **insight, completeness**, and **correctness** of the protocol design.
From Models to Verification

- We would like to verify (guarantee) that the protocol is correct (has the desired properties).

**Properties (questions)**

Deadlock free?
A request is always followed by a response?

**Model Checking Computer Tool**

**Answers [yes/no,...]**
State Space Methods

- One of the main approaches to verification of communication protocols:

**Model**

- **Nodes**: set of reachable states
- **Arcs**: occurrences of events
- **Paths**: set of execution sequences

**Guarantees complete coverage of executions:**
- Systematic error detection
  + Verification

**Main challenge:**
- State explosion problem
Outline

- **Introduction to state space-based verification methods and model checking techniques.** [WP]
- **Formal modelling of protocols:**
  - Petri Nets and Timed Automata. [WP]
  - Hierarchical Coloured Petri Nets and CPN Tools. [LMK]
- **Model checking and verification of protocols:**
  - Bounded Parametric Model Checking for Petri Nets. [WP]
  - Explicit state space exploration of Coloured Petri Nets. [LMK]
- **Examples of case studies and application of computer tools for modelling and verification:**
  - **The VerICS Toolkit:** A selection of smaller case studies. [WP]
  - **CPN Tools:** Edge Router Discovery Protocol and the Generic Access Network Architecture. [LMK]
Classical References