Modeling Network
Embedded Systems with
NS-2 and SystemC

Nicola Drago  Franco Fummi
Stefano Martini

Università di Verona - Dipartimento di Informatica

Maurizio Martignano

Sitek S.p.A. Verona
Outline

➢ Motivations
  ➢ Heterogeneous network
  ➢ Embedded network device
  ➢ Networked embedded system

➢ Environment
  ➢ SystemC
  ➢ NS-2
  ➢ The integrations

➢ Case Study

➢ Conclusions
Motivations: Scenario

IP/Ethernet LAN

Gateway System

Dedicated Network (Ex IEEE 1355)

Gateway System

IP/Ethernet LAN

Embedded System
Two possible scenarios can be identified:

- Embedded network devices.
- Networked embedded systems.
Embedded network devices

Physical and data-link layers are usually specific for some types of connectivity, e.g.:

- xDSL,
- PPP (Point to Point),
- PPPoE (Point to Point other Ethernet),
- IEEE 1394 (Fire Wire),
- etc.
Networked embedded systems

Nodes of a distributed system:

- Set top boxes for pay-per-view digital broadcasting.
- Remote process controllers.
- Routers.
- Gateways.

They interact through networks
What offer the market?

Hardware/Software Field

Hardware design:
- VERILOG
- VHDL

Embedded Software Design:
- C/C++

System modeling:
- SystemC

Network Field

Network design and test:
- C++
- Real Network Simulator
- CNET Network Simulator
- NS-2
SYSTEMC 1.1

Embedded Systems

- SC_METHOD
- SC_THREAD
- SC_CTHREAD
- WAIT
- sc_clock
- sc_signal
- sc_bv
- sensitive
- ports

Network Device

- SC_METHOD
- SC_THREAD
- SC_CTHREAD
- WAIT
- sc_clock
- sc_signal
- sc_bv
- sensitive
- ports
SYSTEMC 1.1

Communication Protocol

- SC_METHOD
- SC_THREAD
- SC_CTHREAD
- WAIT
- sc_clock
- sc_signal
- sc_bv
- sensitive
- ports
SYSTEMC 1.1
Communication Protocol

**BENEFITS:**
- Accuracy.
- Automatic synthesis.

**LIMITATIONS:**
- Time to design the model.
- Effort to realize each module.
- Complexity of adding component.
- Simulations Time
Environment

SYSTEMC 2.0
Communication Protocol

ADDING NEW COMPONENT:

• Interfaces.
• Channels.
• Synchronization.
  • Semaphore.
  • Mutex.
  • Buffer.
  • Fifo.
  • ...
Environment

SYSTEMC 2.0
Communication Protocol

**BENEFITS:**
- Accuracy.
- Refinement.
- *Automatic synthesis* ???

**LIMITATIONS:**
- Time to design the model.
- Effort to realize each module.
- Complexity of adding component.
- Simulations Time
Network Simulator 2

Network Simulator 2 (NS-2) is:

- a discrete event simulator targeted at networking research
- part of VINT: Virtual InterNet Testbed
- a collaborative simulation platform

It can be especially used for design and deployment of new wide area Internet protocols.
NS-2 provides substantial support for simulations based on:
- Internet Protocol:
  - IP.
  - TCP.
  - UDP.
- Routing.
- Multicast protocols.
- Wired connection.
- Wireless connection:
  - Local.
  - Satellite.
NS-2 architecture provides:

- C++ performance simulator primitives.
- Extension for the users.
- Network configurations are specified via a script.
- Simulation program written in a TCL-like scripting language.
The proposed methodology to modeling network embedded systems uses:

- **NS-2** to model the heterogeneous network infrastructure.

- **SystemC** to model:
  - **Network devices** participating to the communication.
  - **Embedded systems** participating to distributed system.
Proposed Methodology

Network Topology

NS2

SYSTEMC

Methodology
Simulations
Synthesis

Synthesis
Behavior
RTL
GATE

VHDL

Synthesis
Behavior
RTL
GATE
Proposed Methodology

Simulators Integration

- NS-2 NODE
- Network Device
- NS-2 NODE
- SystemC
- Embedded System (CPU)
- NS-2 NODE
- NS-2
- Embedded System (CPU)
- SystemC
Case study (1)

Networked embedded system

Diagram showing networked embedded system with nodes connected by lines labeled 'NS-2'. The diagram includes labels for 'Embedded System', 'CPU', and 'SystemC'.
Case study (1)

Networked embedded system
## Networked embedded system

### Experimental results

#### Design Time:

<table>
<thead>
<tr>
<th>Design Time</th>
<th>Only SystemC</th>
<th>SystemC</th>
<th>NS-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 CPU</td>
<td>24 h</td>
<td>0 h</td>
<td></td>
</tr>
<tr>
<td>2 CPU</td>
<td>184 h</td>
<td>68 h</td>
<td></td>
</tr>
<tr>
<td>10 Nodes</td>
<td>224 h</td>
<td>69 h</td>
<td></td>
</tr>
<tr>
<td>100 Nodes</td>
<td>274 h</td>
<td>71 h</td>
<td></td>
</tr>
</tbody>
</table>

#### Execution Time:

<table>
<thead>
<tr>
<th>Simulation Time</th>
<th>User SystemC</th>
<th>Sys SystemC</th>
<th>User NS-2</th>
<th>Sys NS-2</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two CPUs only SystemC</td>
<td>545.2 s</td>
<td>17.2 s</td>
<td>0.0 s</td>
<td>0.0 s</td>
<td>562.4 s</td>
</tr>
<tr>
<td>Two CPUs with NS-2</td>
<td>589.7 s</td>
<td>38.0 s</td>
<td>385.7 s</td>
<td>17.4 s</td>
<td>1030.8 s</td>
</tr>
</tbody>
</table>
Case study (2)

Embedded Network Device
Case study (2)

Embedded Network Device

Experimental results

<table>
<thead>
<tr>
<th>Code Line</th>
<th>Simulation Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Behavioral</td>
<td>3651.3 s</td>
</tr>
<tr>
<td>Register Transfer</td>
<td>5252.7 s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code lines</th>
<th>SystemC code lines</th>
<th>NS-2 code lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Behavioral</td>
<td>1637</td>
<td>207</td>
</tr>
<tr>
<td>Register Transfer</td>
<td>1898</td>
<td>207</td>
</tr>
</tbody>
</table>
Conclusions

The proposed methodology can be applied to:

- Embedded network devices
- Networked embedded systems

SystemC BENEFITS: NS-2 BENEFITS:

- Accuracy
- Automatic synthesis.
- Available developed protocols.
- Simplicity to reconfigure.
- Short simulation time