Java Extensions for OMNeT++ 5.0

Henning Puttnies, Peter Danielis, Christian Koch, Dirk Timmermann
University of Rostock, Germany
Motivation

- Network simulators: evaluation of innovative applications/protocols
- Java:
  - Very predictable + easy to debug
  - Relatively platform independent (heterogeneous IoT scenarios)
- Combination of both in two steps:
  1. Java simulation models for early evaluation
  2. Derive Java prototype implementation to evaluate the approach on different platforms
# State of the Art – Java Network Simulators

<table>
<thead>
<tr>
<th>Name</th>
<th>Existing Modules for Reuse</th>
<th>Simulation Models in Java</th>
<th>Still under Maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>NS-3</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>JNS (NS-2 in JAVA)</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>JNetworkSim</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>JProwler</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Java Simulator</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Psimulator2 (for education)</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

No existing simulator supports Java network simulation sufficiently
Concept – Class Diagrams (Similar to OMNeT 4.6)

- **Base on Java Native Interface (JNI)**
- **MyModel**: Java simulation model
- **jsimple.exe**: 
  - OMNeT++ simulation kernel
  - Extension modules (*.cc)
- **JSimpleModule.cc**: 
  - Inherits from cSimpleModule.cc
  - **javaPeer**: pointer to SimpleModule.java
- **JSimpleModule.java**: 
  - Java wrapper for JSimpleModule.cc
  - **swigCPtr**: pointer to JSimpleModule.cc
Concept – Execution Order (similar to OMNeT 4.6)

- *jsimple.exe*: simulation executable
- *MySim.ini*: loads a *.ned* file that uses *MyModule*
- *initialize()*: calls *initJVM()*
- *initJVM()*: starts JVM
- JVM is a shared library that can execute *.class* files
  \(\rightarrow\) Execution of Java simulation models is possible
- *MyModule()*: calls *JSimpleModule()*
- *JSimpleModule()*: calls C++ code
Evaluation – Combination of INET and Java

- Important for reuse of existing modules from INET
- Never demonstrated before OMNeT++ 5.0
- **EtherHost**: example Ethernet host
- **myEtherHost**: own Ethernet host
- **EtherLLC**: Link Layer
- **IEtherMAC**: Mac Layer
- **EtherEchoSrv**:
  - Registers at *EtherLLC*
  - Echoes received Ethernet packets
  - Modifies Ethernet packets (switches SRC and DEST)

Combination of INET and Java is possible
## Conclusion and Outlook

<table>
<thead>
<tr>
<th>Name</th>
<th>Existing Modules for Reuse</th>
<th>Simulation Models in Java</th>
<th>Still Under Maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>NS-3</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>JNS (NS-2 in JAVA)</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>JNetworkSim</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>JProwler</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>Java Simulator</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>Psimulator2 (for education)</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Java + OMNeT 5.0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

- **Outlook:** Java extensions for OMNeT++ 5.1 (minor effort)
Thank you for your attention. Questions?