Research and Development
Methods for Overlay Networks
Outline

- Mathematical modeling (not covered in this lecture)
- Real implementations
- Simulation
- Emulation
- Practical exercises
Real Implementations

- Letting real users to run the software
- Running on testbeds such as PlanetLab
Real Implementations Demo

- PlanetLab
  
  http://www.planet-lab.org/
Simulation

• Simulation means imitating the key characteristics of a system in a controlled setting

→ Real software is seldom used in simulations.

• Typically happens in simulation time, not in real time

Exceptions: Compiling real software against a simulation framework. Using a bridge between real software and the simulation framework.
Simulation types

- Mathematical
- Ad-hoc
- Discrete-event
Mathematical Simulation

- Often utilizes tools such as MATLAB or Mathematica
- Example: calculating numerical results from a Markovian model
Ad-Hoc Simulation

- Simple simulation programs written “just for the purpose” with a programming language of choice
- Testing of algorithms without network simulation
- Example: Comparing different piece selection algorithms of BitTorrent
Ad-Hoc Simulation Demo

- A BitTorrent piece selection algorithm simulator written in Java.
Discrete-Event Simulation

• Using discrete-event frameworks for simulating the overlay network, including the networking stack
Discrete-Event Simulation

Event Queue

<table>
<thead>
<tr>
<th>Event</th>
<th>Deadline</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event1</td>
<td>00:46</td>
<td>code</td>
</tr>
<tr>
<td>Event2</td>
<td>00:57</td>
<td>code</td>
</tr>
<tr>
<td>Event3</td>
<td>00:58</td>
<td>code</td>
</tr>
<tr>
<td>Event4</td>
<td>00:58</td>
<td>code</td>
</tr>
</tbody>
</table>

Execution

Clock
Discrete-Event Simulation Frameworks

- OMNet++
- NS2
- NS3
Discrete-Event Simulation Frameworks Demo

- OMNet++ OverSim
  http://www.oversim.org/wiki/OverSimFeatures

- Source code of a ns3 simulation
  http://thisblog.runsfreesoftware.com/?q=2009/04/16/first-example-ns-3-comments-explain
Emulation

- Running real software in an emulated network in real time
Emulation Tools

- Bandwidth limitation with Trickle
- Introducing delays and packet loss with NetEM
- Virtual machines with virtual TUN/TAP network interfaces
Emulation Tools Demo

- Trickle
  http://monkey.org/~marius/trickle/trickle.pdf

- NetEM
  http://swik.net/netem/Examples+of+Use
Simulation vs. Emulation

- Simulation is run in simulation time → better scalability
- Simulation allows experimentation in more versatile network conditions and topologies
- Simulation usually involves running simplified software, not real software → Less reliable results in research, difficulty of using simulators as development tools
Combining Simulation and Emulation

- Benefit: possibility to run real software with simulator-like scalability
Combining Simulation and Emulation

- Use a discrete-event simulator for simulating the network
- Trick virtual machines to run in simulation time
- Example: SliceTime
Combining Simulation and Emulation Demo

- SliceTime

Practical Exercises

- Goal: Learn to develop real overlay networking software while using a network simulator as a development tool
Exercise Environment

- Operating system: Linux
- Programming language: C++
- NS3 network simulator
- A class library developed at the department that allows for compiling the same code as a NS3 simulation executable and as real software
Excercise Environment Demo

- The design of the class library to be used in the exercises
Excercise Groups

• Group work allowed, groups of 3 persons maximum

• At least one group member should have C/C++ programming experience

• Groups will be agreed on during this lecture. If you did not attend the lecture and you would like to join a group, please send email to petri.savolainen @ hiit.fi

It is also allowed to do exercises on your own without joining group.