An Open-source and Declarative Approach Towards Teaching Large-scale Networked Systems Programming

Harjot Gill, Taher Saeed, Qiong Fei, Zhuoyao Zhang, Boon Thau Loo
University of Pennsylvania

ACM SIGCOMM Education Workshop 2011

http://netdb.cis.upenn.edu/cis553projects/
Introduction

- Networked Systems (CIS 553) class at Penn
  - Class started in 2007
  - Large class: 80+ students. Majority are Masters students
  - Cross listed with TCOM graduate course
  - One of the largest graduate level courses at Penn in CIS/TCOM

http://netdb.cis.upenn.edu/cis553projects/
Introduction

• Networked Systems (CIS 553) class at Penn
  – Class started in 2007
  – Large class: 80+ students. Majority are Masters students
  – Cross listed with TCOM graduate course
  – One of the largest graduate level courses at Penn in CIS/TCOM

• Syllabus:
  – Review of undergraduate networking material
  – Advanced topics on application overlay networks, mobile networks etc.
  – Two large programming assignments (routing protocol and distributed hash table implementation)
  – Group of 3-4 students work on the projects
  – In-class demonstrations
Key Challenges and Contributions

• Key challenges: finding an appropriate platform for students to develop course projects. Desired features:
  – High degree of realism
  – Ease of configuration and testing
  – Ease of emulation
  – Well organized code base to exercise software engineering skills
  – Projects can be completed within a few weeks

http://netdb.cis.upenn.edu/cis553projects/
Key Challenges and Contributions

• Key challenges: finding an appropriate platform for students to develop course projects. Desired features:
  – High degree of realism
  – Ease of configuration and testing
  – Ease of emulation
  – Well organized code base to exercise software engineering skills
  – Projects can be completed within a few weeks

• Key contributions:
  – Development of new projects on the open-source ns-3 platform
  – Use of declarative networking for final project

http://netdb.cis.upenn.edu/cis553projects/
## Alternatives

<table>
<thead>
<tr>
<th>Platform</th>
<th>Fishnet ('07-'09)</th>
<th>ns-2</th>
<th>ns-3 ('10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>JAVA</td>
<td>Multiple- C/C++/OTcl</td>
<td>C/C++ &amp; Python (scripts)</td>
</tr>
</tbody>
</table>
**Alternatives**

<table>
<thead>
<tr>
<th>Platform</th>
<th>Fishnet ('07-'09)</th>
<th>ns-2</th>
<th>ns-3 ('10)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Language</strong></td>
<td>JAVA</td>
<td>Multiple- C/C++/OTcl</td>
<td>C/C++ &amp; Python (scripts)</td>
</tr>
<tr>
<td><strong>Sockets, IP Addresses, Devices, Channels</strong></td>
<td>✗</td>
<td>✓</td>
<td>✓✓</td>
</tr>
<tr>
<td><strong>Test-bed/Emulation support</strong></td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Animation support</strong></td>
<td>✗</td>
<td>✓✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Mobility support</strong></td>
<td>✗</td>
<td>✓</td>
<td>✓✓</td>
</tr>
<tr>
<td><strong>PCAP support</strong></td>
<td>✗</td>
<td>✓✓</td>
<td>✓✓</td>
</tr>
</tbody>
</table>
# Alternatives

<table>
<thead>
<tr>
<th>Platform</th>
<th>Fishnet ('07-'09)</th>
<th>ns-2</th>
<th>ns-3 ('10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>JAVA</td>
<td>Multiple- C/C++/OTcl</td>
<td>C/C++ &amp; Python (scripts)</td>
</tr>
<tr>
<td>Sockets, IP Addresses, Devices, Channels</td>
<td>✗</td>
<td>✓</td>
<td>✓✓✓</td>
</tr>
<tr>
<td>Test-bed/Emulation support</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Animation support</td>
<td>✗</td>
<td>✓✓✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mobility support</td>
<td>✗</td>
<td>✓</td>
<td>✓✓✓</td>
</tr>
<tr>
<td>PCAP support</td>
<td>✗</td>
<td>✓✓</td>
<td>✓✓✓</td>
</tr>
<tr>
<td>Current open-source community activity</td>
<td>✗</td>
<td>✓</td>
<td>✓✓✓</td>
</tr>
<tr>
<td>Documentation</td>
<td>✓</td>
<td>✓✓</td>
<td>✓✓</td>
</tr>
<tr>
<td>Code complexity</td>
<td>✓✓</td>
<td>✓</td>
<td>✓✓</td>
</tr>
</tbody>
</table>
ns-3 Architecture

Simulator

Channels and Mobility Models

Simulator Event Queue and Scheduler (default or real-time)

http://netdb.cis.upenn.edu/cis553projects/
Course Related Customizations I: Driver and Skeleton Code

- Driver and skeleton code: 1500 lines of initial code provided
- ns-3 base-code size: 90K lines (ns-3.9)
- Students have to look at only a few hundred lines

http://netdb.cis.upenn.edu/cis553projects/
Course Related Customizations I: Driver and Skeleton Code

- Driver and skeleton code: 1500 lines of initial code provided
- ns-3 base-code size: 90K lines (ns-3.9)
- Students have to look at only a few hundred lines

http://netdb.cis.upenn.edu/cis553projects/
Course Related Customizations I: Driver and Skeleton Code

- Driver and skeleton code: 1500 lines of initial code provided
- ns-3 base-code size: 90K lines (ns-3.9)
- Students have to look at only a few hundred lines

http://netdb.cis.upenn.edu/cis553projects/
Course Related Customizations I: Driver and Skeleton Code

- Driver and skeleton code: 1500 lines of initial code provided
- ns-3 base-code size: 90K lines (ns-3.9)
- Students have to look at only a few hundred lines

http://netdb.cis.upenn.edu/cis553projects/
Course Related Customizations I:
Driver and Skeleton Code

- Driver and skeleton code: 1500 lines of initial code provided
- ns-3 base-code size: 90K lines (ns-3.9)
- Students have to look at only a few hundred lines

http://netdb.cis.upenn.edu/cis553projects/
Course Related Customizations I: Driver and Skeleton Code

- Driver and skeleton code: 1500 lines of initial code provided
- ns-3 base-code size: 90K lines (ns-3.9)
- Students have to look at only a few hundred lines

http://netdb.cis.upenn.edu/cis553projects/
Course Related Customizations I: Driver and Skeleton Code

- Driver and skeleton code: 1500 lines of initial code provided
- ns-3 base-code size: 90K lines (ns-3.9)
- Students have to look at only a few hundred lines

http://netdb.cis.upenn.edu/cis553projects/
Course Related Customizations I: Driver and Skeleton Code

- Driver and skeleton code: 1500 lines of initial code provided
- ns-3 base-code size: 90K lines (ns-3.9)
- Students have to look at only a few hundred lines

http://netdb.cis.upenn.edu/cis553projects/
Course Related Customizations II: New Emulation Design

- Transport library
  - Default ns-3 requires *sudo* permission for emulation (due to raw-sockets)
  - Granting *sudo* permission is problematic in a large class setting
  - Transport library uses regular UDP/TCP sockets
Networked Systems Project 1: Routing Protocols

- Implement neighbor discovery and routing (link-state and distance vector)
- Extra credits: MANET routing, path-vector, metric based routing, etc.
- Students wrote 1500+ lines of code in 3 weeks
- RouteInput/RouteOutput: Compatibility with ns-3’s IP stack

http://netdb.cis.upenn.edu/cis553projects/
Networked Systems Project 1: Routing Protocols

- Implement neighbor discovery and routing (link-state and distance vector)
- Extra credits: MANET routing, path-vector, metric based routing, etc.
- Students wrote 1500+ lines of code in 3 weeks
- RouteInput/RouteOutput: Compatibility with ns-3’s IP stack

http://netdb.cis.upenn.edu/cis553projects/
Implement neighbor discovery and routing (link-state and distance vector)
Extra credits: MANET routing, path-vector, metric based routing, etc.
Students wrote 1500+ lines of code in 3 weeks
RouteInput/RouteOutput: Compatibility with ns-3’s IP stack

http://netdb.cis.upenn.edu/cis553projects/
Networked Systems Project 2a: PennSearch: DHT-Search

- Built as an overlay above Project 1’s routing protocol
- Students implement Chord DHT [SIGCOMM ‘01] with search engine layer on top
- Students wrote 4000+ lines of code
- Extra credits: sophisticated ranking algorithms for searching (e.g. TF-IDF), TCP support, churn handling, bloom filters, etc.

http://netdb.cis.upenn.edu/cis553projects/
Project 2 Architecture

PennSearch Node

Search Engine

Chord Overlay

Addr ← lookup(K)

Search Engine

Chord Overlay

Addr ← lookup(K)

Chord Protocol, e.g. stabilization, finger fixing. Uses UDP for communication.

PennSearch Node

http://netdb.cis.upenn.edu/cis553projects/
PennSearch: Keyword-based Search over Chord

http://netdb.cis.upenn.edu/cis553projects/
PennSearch: Keyword-based Search over Chord

Query: “T₁ AND T₂”
PennSearch: Keyword-based Search over Chord

Query: “T₁ AND T₂”

http://netdb.cis.upenn.edu/cis553projects/
PennSearch: Keyword-based Search over Chord

Query: “T₁ AND T₂”

http://netdb.cis.upenn.edu/cis553projects/
PennSearch: Keyword-based Search over Chord

Query: “T₁ AND T₂”
Networked Systems Project 2b: Declarative Networking

- Two volunteer groups implemented project 2 using declarative networking [SIGCOMM’05] framework
- Used RapidNet* declarative networking system [SIGCOMM’09 demo]
  - Compact and high-level representation of protocols
  - Orders of magnitude reduction in code size (add 50 rules)
- Implemented the Content Addressable Network (CAN), since a declarative Chord implementation [SOSP’05] exists

* http://netdb.cis.upenn.edu/rapidnet
Student Experiences

• What did students learn from ns-3?
  – Software engineering. Learning to integrate ideas from class into a 90K code-base.
  – Working in teams of 3-4 students
  – Open-source projects (ns-3 mailing list, online documentation)
  – Students are willing to put in the hours, if they see value in class projects (helps improve their CVs for job applications)

http://netdb.cis.upenn.edu/cis553projects/
Student Experiences

• What did students learn from ns-3?
  – Software engineering. Learning to integrate ideas from class into a 90K code-base.
  – Working in teams of 3-4 students
  – Open-source projects (ns-3 mailing list, online documentation)
  – Students are willing to put in the hours, if they see value in class projects (helps improve their CVs for job applications)

• Declarative networking?
  – Positive feedback from students for declarative networking
  – Declarative networking implementation as good as C++ implementation

http://netdb.cis.upenn.edu/cis553projects/
Student Experiences

• What did students learn from ns-3?
  – Software engineering. Learning to integrate ideas from class into a 90K code-base.
  – Working in teams of 3-4 students
  – Open-source projects (ns-3 mailing list, online documentation)
  – Students are willing to put in the hours, if they see value in class projects (helps improve their CVs for job applications)

• Declarative networking?
  – Positive feedback from students for declarative networking
  – Declarative networking implementation as good as C++ implementation

• How did students rate the class?
  – Highest course ratings (18% jump from previous year); record enrollment (81 students)
  – Highest rated course in Fall 2010 among TCOM classes

http://netdb.cis.upenn.edu/cis553projects/
Instructor Experiences

• What did the instructors learn?
  – ns-3 provides a good balance between realism (e.g. sockets, PCAP, IP addresses), and feasibility
  – Have to make sure that projects do not become widely available on the internet
  – No signs of cheating (ran MOSS on all projects, including code in open-source)
  – Group evaluations helped ensure student participation. However, learning curve of ns-3 sometimes result in imbalanced contribution
  – Exploring full-fledged use of declarative networking in future
Thank You!

• Projects: http://netdb.cis.upenn.edu/cis553projects/
  – Project write-up is available
  – Links to class syllabus and timeline
  – Project code (driver/skeleton, sample solution) is available on request

• RapidNet: http://netdb.cis.upenn.edu/rapidnet

Any Questions?