Princeton P4 Campus:
Building and Running Novel Network Applications on Campus
https://p4campus.cs.princeton.edu

Hyojoon (Joon) Kim
Princeton University

Programmable Switches Workshop
February 23, 2022
P4 Campus

An initiative to create and deploy experimental but useful network applications on a production campus network

We primarily use programmable data planes and P4

Why?  How?
The “Gap”

Bad idea

New idea

Small, unrealistic experiments

Outdated tools & practices

Research

Production Network
Positive Feedback Loop Missing

- Research
- Production Network
Network Research Pipeline

Idea and Design → Software w/ simulations → Hardware w/ synthetic traffic → Hardware w/ real traces (lab) → Real deployment w/ live traffic

NS2, Mininet, P4 BMv2
Network Research Pipeline

- **Idea and Design**
- **Software w/ simulations**
- **Hardware w/ synthetic traffic**
- **Hardware w/ real traces (lab)**
- **Real deployment w/ live traffic**

- NS2
- Mininet
- P4 BMv2

Programmable hardware

No feedback loop!
Network Research Pipeline

Idea and Design → Software w/ simulations → Hardware w/ synthetic traffic → Hardware w/ real traces (lab) → Real deployment w/ live traffic

Without the last two stages, new ideas barely see light outside of a lab
Real Traces & Deployment is Hard

- Disruptive
- User privacy (PII in production traffic)
- Lack of collaboration and communication
Alternatives: Dedicated Testbed

Real network, WAN-scale

Limited access to production traffic
Campus Network as Lab

Variety of traffic
- Science
- Data center
- Residential
- Business

Open and Dynamic
- Open to public
- Many visitors & events
- BYOD devices
- Closer to user

Enterprise vs Cloud
- Still has value as an access network
- Enterprise solutions are applicable to cloud networks
Campus Network as Lab

Research-friendly

Existing mechanisms

OIT is committed to providing technology services that enable Princeton to achieve its mission to advance learning through scholarship, research, and teaching of unsurpassed quality.
Jumping The Research Chasm

Disruptive Privacy Lack of collaboration

Research Idea

Real Campus Deployment

Hardware w/ real traces

Real deployment w/ live traffic

feedback
1. Sharing Our Experience

<table>
<thead>
<tr>
<th>Less Disruptive</th>
<th>Preserve Privacy</th>
<th>More Collaboration with OIT</th>
</tr>
</thead>
</table>

13
2. Successful Deployments

**ONTAS, *Flow**

*live traffic anonymization*

**enablers**

**Precision, ConQuest, P4-RTT, P4-DNS**

*passive traffic analytics*

**PINOT**

*active traffic experiment*
# Becoming Less Disruptive

<table>
<thead>
<tr>
<th>Less Disruptive</th>
<th>Preserve Privacy</th>
<th>More Collaboration with OIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Work w/ mirrored traffic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>● Passive monitoring as a “gateway drug”</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Traffic Mirroring

Network TAP

ISP NY

ISP Philly

Princeton Network

Borders

Firewalls

Cores

... buildings

... buildings
Working with Mirrored Traffic

- Low risk
  - Little or no disruption

- High return
  - Real-time traffic analysis is a nice “gateway drug”
Some Tips on Mirrored Traffic

- TAP is better than port mirroring (SPAN)
  - SPAN burns a port and resources on a switch

- Modern *packet broker systems* can
  - Apply filtering policies
  - Remove/mask payload
  - Remove duplicate packets
Preserving User Privacy

<table>
<thead>
<tr>
<th>Less Disruptive</th>
<th>Preserve Privacy</th>
<th>More Collaboration with OIT</th>
</tr>
</thead>
</table>
| ● Work w/ mirrored traffic  
● Passive monitoring as a “gateway drug” | ● IRB  
● Data anonymization |                           |
Navigating Campus Traffic Data Access

Institutional Review Board (IRB)

- Rights, Privacy, Welfare of Human Subjects

Institutional Review Panel for the use of Administrative Data in Research (PADR)

- Feasibility, Value, Risk, and Compliance
Prepping IRB Applications

State that you will remove/anonymize PII

- MAC and <Your Institute> IP addresses will be anonymized
- Payload will be removed
- If not, operator will run scripts/programs and provide aggregated results

Show you will take good care of the data

- Data will be stored and processed at machines managed by IT staff
Offline Data Anonymization

- CAIDA’s best practices and offline tools
  - [https://www.caida.org/projects/predict/anonymization/](https://www.caida.org/projects/predict/anonymization/)
Anonymize Live Traffic

ONTAS
P4 program

controller

Configure which fields to hash

Hash relevant fields (e.g., IP, MAC)

~100 Gbps

Anonymized traffic

P4 switch or server

Hyojoon Kim et. al., "ONTAS: Flexible and Scalable Online Network Traffic Anonymization System."
2019 SIGCOMM Workshop on Network Meets AI & ML
ONTAS P4 program hash function + salt

- Online (not offline)
- Line-rate
- Customizable (e.g., select IPs, preserve prefix, etc)
Collaboration With The IT Group

<table>
<thead>
<tr>
<th>Less Disruptive</th>
<th>Preserve Privacy</th>
<th>More Collaboration with OIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work w/ mirrored traffic</td>
<td>IRB</td>
<td>Tackle problems that matter</td>
</tr>
<tr>
<td>Passive monitoring as a</td>
<td>Data anonymization</td>
<td>Joint positions</td>
</tr>
<tr>
<td>“gateway drug”</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Collaboration with OIT

Find problems that matter

- Operator is tired of anonymizing traffic for researchers. Harder for live traffic.
  - ONTAS: traffic anonymization

- Occasional packet drops at switch, but don’t know why.
  - ConQuest: Queue monitoring

- Having latency problems. Where is the bottleneck?
  - P4-RTT: Measure RTT at different vantage points
Successful Deployments

- Real-time OS fingerprinting
- Continuous RTT monitoring
P40f: OS Fingerprinting with P4

- Fingerprint OS type in the data plane
- Higher abstraction than IP addresses

Write policies based on OS type
- “Block all traffic from Windows XPs”
- “Rate-limit traffic to/from Echo Dot”
- “Monitor OS distribution in real time”
- …
The *p0f* Tool

- Each OS uses a unique combination of IP/TCP header values

- *p0f* signature

- Example: Linux v3.11 or higher
  - *: 64: 0: *: (20, 10): (mss, sok, ts, nop, ws): (df, id+): 0

---

The *p0f* tool cannot run against live traffic with high data rate
P40f: Let’s Do This in the Switch

[Diagram showing the process of packet handling in a switch, with stages for parsing, control pipeline, and deparsing.]

1. p0f signatures
2. actions

Match-action table

<table>
<thead>
<tr>
<th>p0f_metadata</th>
<th>OS label + action</th>
</tr>
</thead>
<tbody>
<tr>
<td>p0f signature for Linux 3.1</td>
<td>Linux 3.1, drop</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Packet in

Standard IP/TCP header

TCP option stage 1

TCP option stage 2

TCP option stage 10

EOL

NOP

TS

p0f_metadata

Parsing

Control pipeline

Deparsing

Packet out

Deparse packet

Rules & action compiler

PRINCETON UNIVERSITY
Against 3-hour campus trace

<table>
<thead>
<tr>
<th>Internal hosts</th>
<th>External hosts</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OS Label</strong></td>
<td><strong>p0f-v3.09b</strong></td>
</tr>
<tr>
<td>Linux</td>
<td>Linux</td>
</tr>
<tr>
<td>2.2.x-3.x</td>
<td>11412</td>
</tr>
<tr>
<td>3.11+</td>
<td>9558</td>
</tr>
<tr>
<td>3.1-3.1.10</td>
<td>1406</td>
</tr>
<tr>
<td>3.x</td>
<td>332</td>
</tr>
<tr>
<td>Android</td>
<td>21</td>
</tr>
<tr>
<td>2.4.x</td>
<td>20</td>
</tr>
<tr>
<td>2.2.x-3.x (barebone)</td>
<td>15</td>
</tr>
<tr>
<td>2.2.x-3.x (no timestamps)</td>
<td>11</td>
</tr>
<tr>
<td>2.6.x</td>
<td>5</td>
</tr>
<tr>
<td>2.4.x-2.6.x</td>
<td>5</td>
</tr>
<tr>
<td>Windows</td>
<td>Windows</td>
</tr>
<tr>
<td>NT kernel</td>
<td>11753</td>
</tr>
<tr>
<td>NT kernel 5.x</td>
<td>10202</td>
</tr>
<tr>
<td>7 or 8</td>
<td>920</td>
</tr>
<tr>
<td>XP</td>
<td>560</td>
</tr>
<tr>
<td>XP</td>
<td>65</td>
</tr>
<tr>
<td>NT kernel 6.x</td>
<td>6</td>
</tr>
<tr>
<td>Mac</td>
<td>Mac</td>
</tr>
<tr>
<td>OS X</td>
<td>23917</td>
</tr>
<tr>
<td>OS X 10.x</td>
<td>23634</td>
</tr>
<tr>
<td>OS X 10.9+ (iPhone/iPad)</td>
<td>171</td>
</tr>
<tr>
<td>Other</td>
<td>Other</td>
</tr>
<tr>
<td>FreeBSD</td>
<td>47</td>
</tr>
<tr>
<td>FreeBSD 9.x+</td>
<td>37</td>
</tr>
<tr>
<td>NMap SYN scan</td>
<td>9</td>
</tr>
<tr>
<td>Unclassified</td>
<td>Unclassified</td>
</tr>
<tr>
<td>Total</td>
<td>374047</td>
</tr>
</tbody>
</table>
P4-RTT

Continuous round trip time monitoring *beyond* the TCP handshake
How P4-RTT Operates

- **External Leg**: TAP traffic from the Internet to the Home Network.
- **Internal Leg**: Stream of RTT measurements from the Home Network to the Collection & Analysis server.

Diagram: A network diagram showing the flow of traffic from the Internet to the Home Network, with TAP traffic and a stream of RTT measurements.
JOIN of outgoing and incoming packet streams in the data plane
Many TCP packets don’t receive a corresponding ACK

Lazily expire entries with a threshold

Threshold: 99th percentile RTT (500ms)
Multi-stage Hash Table (Registers)

Overcomes the memory limit per register

Overcomes RTT sample loss due to hash collision
Per Application RTT

RTT samples for Youtube
Mean RTT = 20.04 ms
Median RTT = 15.0 ms

RTT samples for Netflix
Mean RTT = 49.52 ms
Median RTT = 37.15 ms

RTT samples for Zoom
Mean RTT = 52.22 ms
Median RTT = 41.11 ms
Impact of wired vs. wireless infrastructure on 90\%ile RTT to YouTube
Summary

Jumping The Research Chasm

- **Less disruptive**
  - Passive traffic monitoring
- **Preserve user privacy**
  - IRB prep (and more)
  - Anonymization tools
- **More collaboration**
  - Joint position
  - problems that matter

Real Deployment Successes

- **ONTAS**
  - anonymized data collection
- **P40f**
  - Real-time OS fingerprinting
- **P4 RTT**
  - Continuous RTT monitoring
Experience-Driven Research

Experience-Driven Research on Programmable Networks
Hyojoon Kim, Xiaoqi Chen, Jack Brassil, and Jennifer Rexford

Github repo for our P4 projects:
https://github.com/Princeton-Cabernet/p4-projects
Please Join Our Effort!
More Campus Applications

Current P4 Applications

Measure microbursts
Catch the microburst culprits in the network.

Find heavy hitters
Find heavy hitters in the network.

Anonymize live traffic
Anonymize live network traffic in line-rate.

Fingerprint OS

<table>
<thead>
<tr>
<th>Host IP</th>
<th>OS type</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.2</td>
<td>Linux 3.1-3.10</td>
</tr>
<tr>
<td>172.17.2.10</td>
<td>Windows XP</td>
</tr>
<tr>
<td>10.0.0.3</td>
<td>Mac OS X 10.9 or newer</td>
</tr>
<tr>
<td>192.168.2.10</td>
<td>Windows NT kernel 5.x</td>
</tr>
</tbody>
</table>

Measure Flow RTT

Median RTT per flow

Protect against surveillance

...More from you!

https://p4campus.cs.princeton.edu
Thank You!

P4 Campus Website:  
https://p4campus.cs.princeton.edu

Reach me at:  
hyojoonk@cs.princeton.edu