SWINBURNE UNIVERSITY OF TECHNOLOGY

TCP Experiment Automation Controlled Using Python (TEACUP)

Sebastian Zander, Grenville Armitage

Centre for Advanced Internet Architectures (CAIA)
Swinburne University of Technology

Overview

- TCP Experiments
- CAIA TCP Testbed
- TEACUP – Experiment Setup
- TEACUP – Data Analysis
- Future Work & Summary

Why TCP Experiments?

- TCP is more than three decades old
- Still lots of ongoing work to improve its performance
  - Short flows
  - Congestion control
  - Loss recovery
  - Incast problem
  - AQM mechanisms
  - …
- Who will write last PhD thesis on optimising TCP?

TCP Experiments Environment

- Simulator (ns-2, ns-3, …)
  - Easy to use, nice repeatable results
  - Large complex networks
  - No real TCP stacks or applications
  - Can get close to real stacks, e.g. TCP-Linux for ns-2 http://heim.ifi.uio.no/michawe/research/tools/ns/index.html
- Real networks (e.g. Internet)
  - The real thing
  - Uncontrollable and unpredictable

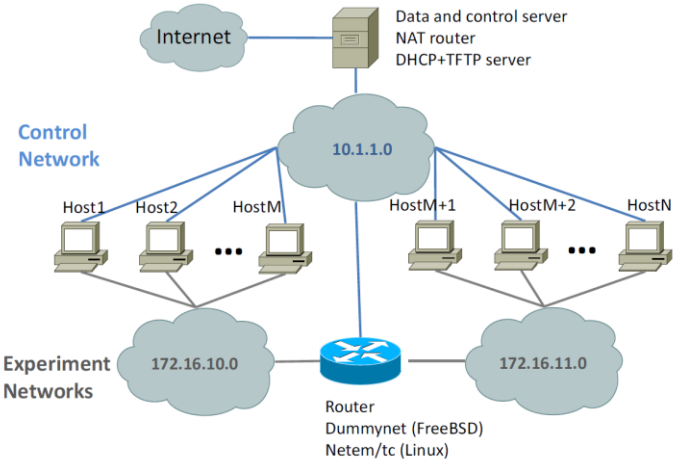
TCP Experiments Environment

- Testbed
  - Real TCP stacks of different OS
  - Real or emulated applications
  - Controllable environment
    - Delay
    - Loss
    - Bandwidth
  - Limited in scale
- **TEACUP is for controlling testbed experiments**

CAIA TCP Testbed

- Traffic sources/sinks
  - 21 PCs (Linux, FreeBSD, Windows)
  - 2 Mac Minis
  - 2 WD Media Players
- 3 PC Routers (Linux, FreeBSD)
- Network
  - 2 Dell Managed Gig Ethernet Switches
  - 1 Pica Gig Ethernet SDN Switch
- Management
  - 3 IP KVMs
  - 5 Power controllers
- 2 Control PCs
  - Experiment control
  - Data storage & analysis

CAIA TCP Testbed Logical Topology



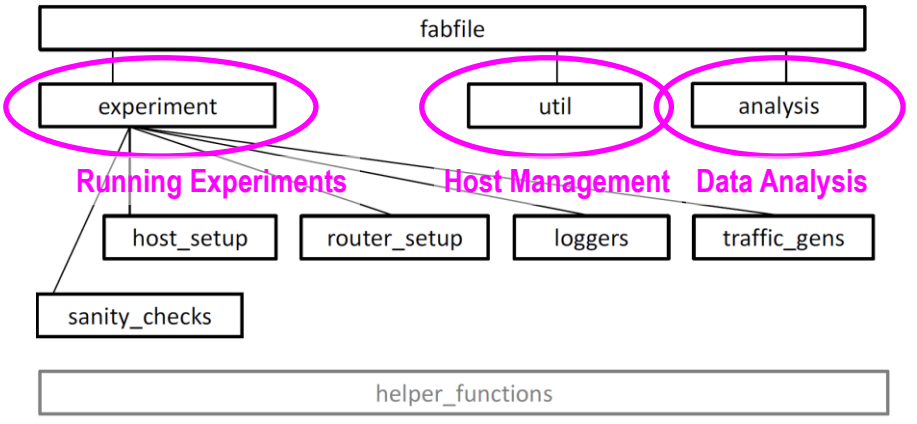
- TEACUP can automatically move hosts between subnets
- Overall topology can be changed through VLAN reconfiguration

CAIA TCP Testbed Picture

TEACUP Architecture

- Implemented in Python + shell script + R
- Based on Python Fabric
  - Fabric is module for remotely controlling many hosts
- Small tasks than can be
  - Executed from command line
  - Combined into larger tasks
- Central config file (Python file)

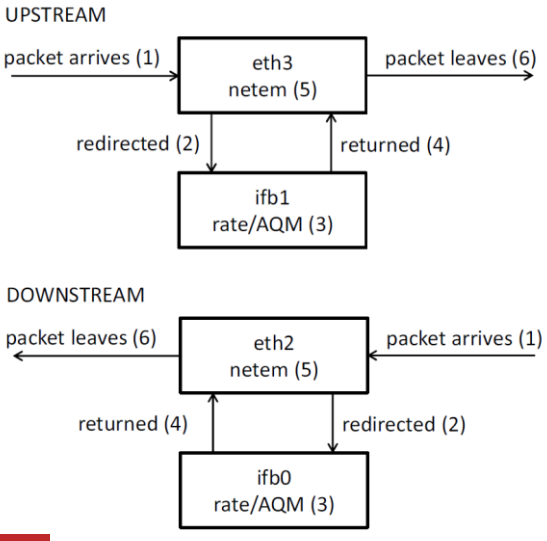
TEACUP Architecture



TEACUP Host Setup

- Operating System (OS)
- Connected subnet
- Disable various hardware offloading, e.g. TSO
- Set socket buffer sizes
- Set TCP congestion control
- Enable/disable Explicit Congestion Notification (ECN)
- Execute user custom commands

TEACUP Router Setup (Linux)



- Based on netfilter and advanced routing (tc)
- Use HTB for shaping
- Separate bottleneck queue (AQM) and bandwidth shaping from delay, loss, etc. emulation (netem) with pseudo interfaces (ifb)

TEACUP Data Logging

- Various host information (e.g. uname, ifconfig, sysctl)
- Router queue information (after experiment)
- TEACUP configuration and per-experiment settings
- tcpdump
- TCP stack information
  - SIFTR (FreeBSD)
  - Web10G (Linux)
  - CAIA-developed tools for Windows and Mac OS X

TEACUP Traffic Generators

- TCP bulk transfer (iperf)
- UDP fixed rate (iperf)
- HTTP (httperf)
- TCP streaming-like (httperf)
- Incast scenario (httperf)
- VoIP-like (nttcp)
- FPS game-like (BITSS)

TEACUP Variable Parameters

- TCP parameters exposed including congestion control
- ECN on/off
- Network delay
- Packet loss
- Bandwidth (upstream/downstream)
- AQM technique
- AQM buffer size
- Traffic generator parameters
- User defined parameters

TEACUP Config File

- Define testbed settings (hosts, router)
- Define network traffic
- Define parameters to vary in experiments

```
# Path to teacup scripts
TPCONF_script_path = '/home/teacup/teacup-0.9'

# Host lists
TPCONF_router = [ 'newtcprt3', ]
TPCONF_hosts = [ 'newtcp20', 'newtcp27' ]

# Map external IPs to internal IPs
TPCONF_host_internal_ip = {
    'newtcprt3': ['172.16.10.1', '172.16.11.1'],
    'newtcp20': ['172.16.10.60'],
    'newtcp27': ['172.16.11.67']
}
```

TEACUP Running Experiments

- Copy fabfile.py and run.sh
- Create config file
- Run single experiment

```
> fab run_experiment_single
```

- Run series of experiments

```
> fab run_experiment_multiple
OR
> ./run.sh
```


TEACUP Host Control

- Reboot

```
> fab -H newtcp20,newtcp21 init_os:os_list="Linux"
```

- Power cycle

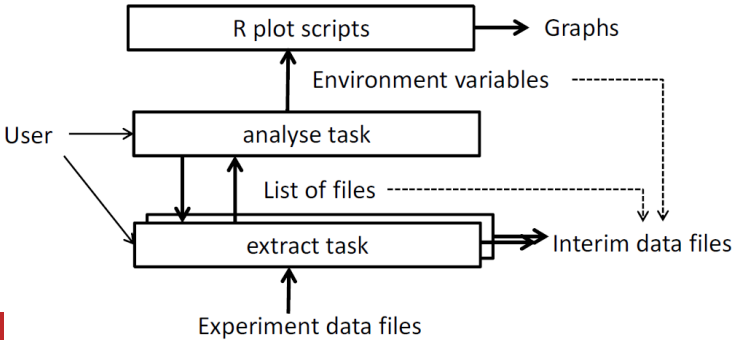
```
> fab -H newtcp20,newtcp21 power_cycle
```

TEACUP Clock Offset Correction

- Clocks in testbed are synchronised with NTP
  - Not very accurate (especially on Windows)
- TEACUP has own clock offset correction
  - During experiment control host sends regular heartbeats on control network to multicast/broadcast destination
  - All hosts receive each heartbeat broadcasted by switch
  - Compute offsets between each host's clock and reference clock and use to correct timestamps for graphs
  - Switch and receiving PCs add noise, but it is in order of 10s of microseconds whereas NTP's noise is up to 10s of milliseconds

TEACUP Data Analysis

- Basic analysis tasks (modular design)
- Analysis is separate from experiment setup
- Can write your own analysis scripts or modify existing functions

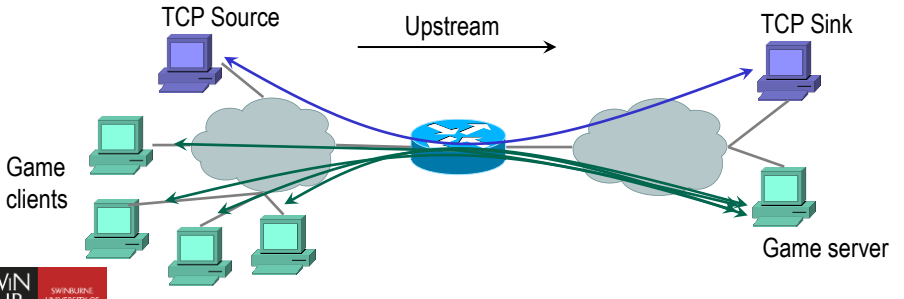


TEACUP Analysis Functions

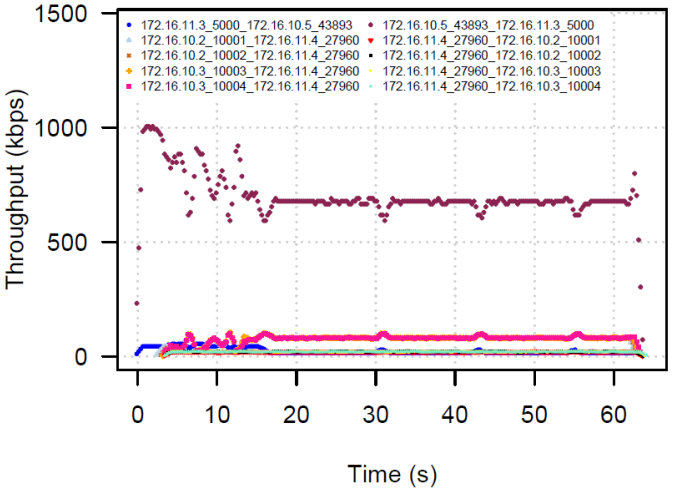
- SPP RTT or TCP RTT over time
- CWND over time
- Throughput over time
- Selected TCP statistic over time
- Incast response time over time
- Acknowledged bytes, dupACKs over time (incast)
- Comparison of one metric for different parameter combos (boxplot, mean, median)
- Plot of two metrics for different parameter combos (2d density or ellipse plot)

TEACUP Experiment

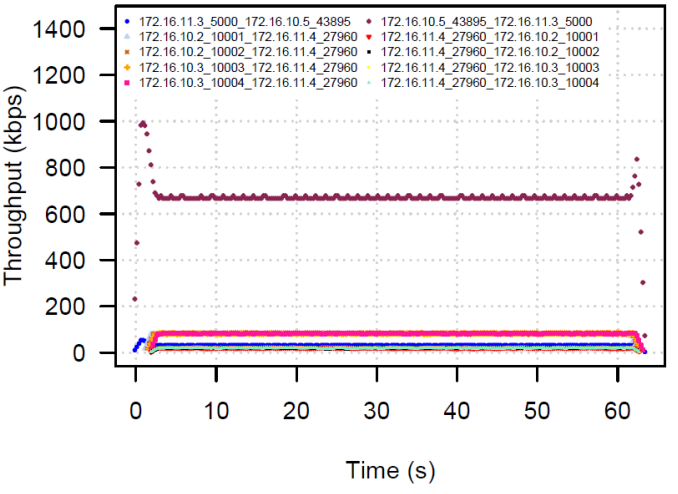
- Bandwidth: 20mbps down, 1mpbs up
- Intrinsic RTT: 40ms
- Traffic: game traffic (4 clients) + TCP bulk transfer
- AQMs: FIFO, PIE, Codel, FQ-Codel (200 packet queue limit)
- Linux Newreno, no ECN



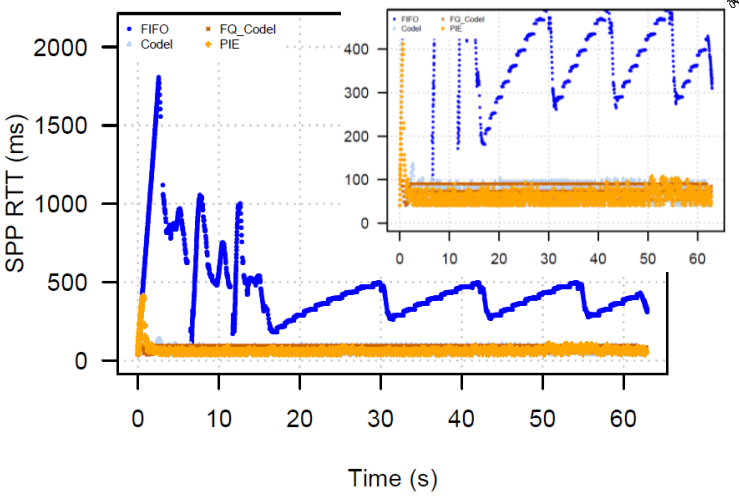
Upstream Throughput FIFO



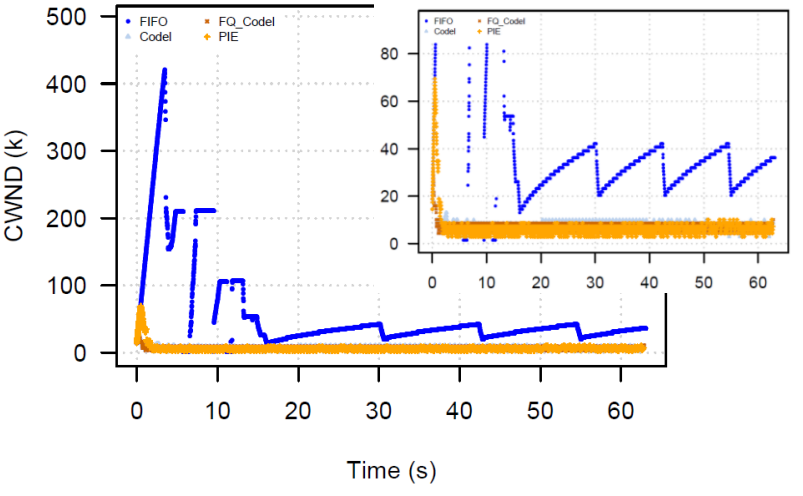
Upstream Throughput FQ-Codel



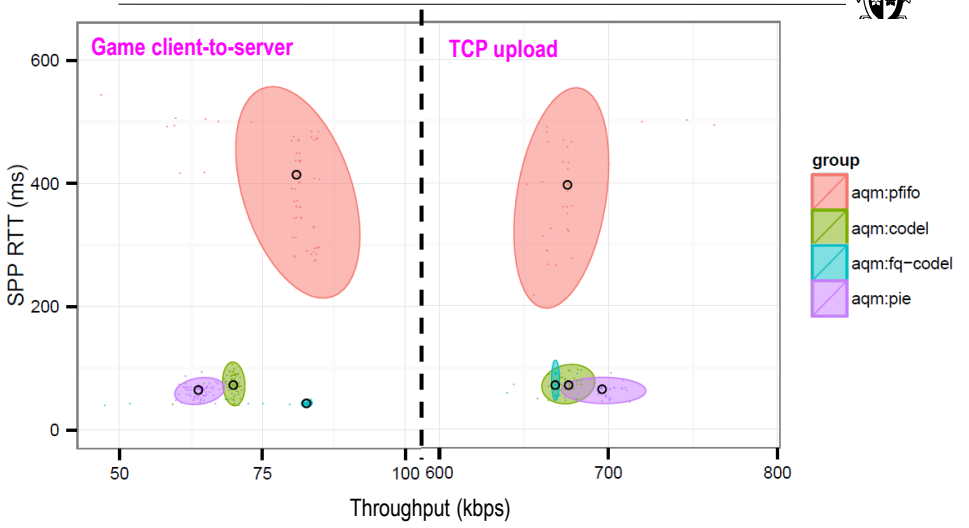
Round Trip Time (RTT)



TCP CWND



AQM Comparison



Future Work

- More traffic generators
- More analysis functions
- Improve TEACUP
  - Better architecture
  - Better user-friendliness

TEACUP Summary

- Greatly simplifies TCP experiments in testbed
- Basic analysis/plotting functions
- Can use in a VM testbed
- Not limited to just TCP
- Could be used for teaching, after we make it more user-friendly
- Code and tech reports
  - http://caia.swin.edu.au/tools/teacup/

TEACUP v0.9 (April 2015) was developed as part of a project funded by Cisco Systems and titled "Study in TCP Congestion Control Performance In A Data Centre". This is a collaborative effort between CAIA and Fred Baker of Cisco Systems.

Questions?