Teaching Computer Networking with Mininet

Session 2: Hands-on Lab -- BufferBloat

Te-Yuan (TY) Huang
Stanford/Netflix
Aug. 18th, 2014
Goals of the Assignment

• Understand the dynamics between TCP CWND and router’s buffer occupancy
• Understand why large router buffer can lead to poor TCP performance
  – I.e., the buffer bloat problem
Outline of the Session

• Why is Mininet helpful?
• Assignment Overview
• Behind the scene – part 1
  – Architecture
• Time to try it out!
• Behind the scene – part 2
  – The actual code
Why Mininet is Helpful?

• Easy to setup the environment
• Easy to access the information
  – Use `tcpprobe` to monitor TCP CWND at sender
  – Use `tc` to:
    • control the buffer size
    • monitor the buffer occupancy
Assignment Overview

• Part 1: The setup

RTT of h1 to h2
20ms

End Host
Bottleneck Router
Buffer Size: 150 kB
(100 pkt)
Server
Assignment Overview

• Part 2: CWND evolution of a short TCP flow
  – The TCP flow is created through a web request
  – No competing flow on the network
  – Observe the RTT and flow completion time
  – Think about how the CWND is evolved
Assignment Overview

• Part 3: CWND evolution of a long TCP flow
  – TCP flow is created through iperf
  – Observe the RTT and throughput
  – Think about how the CWND is evolved
  – Observe how the long flow affects the short flow
Assignment Overview

• Part 4: Verify the evolution of CWND through Mininet
• Part 5: Explore a solution: smaller buffer
• Part 6: Explore a solution: separate queue for each flows
Behind the Scene – Part 1

Each runs in a container

Open VSwitch
Use ‘tc’ to control the delay, link rate and queue length
Time to try it out!
Topology

• Defined in bufferbloat.py

```python
self.addHost('h1')
self.addHost('h2')
self.addSwitch('s0')
self.addLink('h1', 's0')
self.addLink('h2', 's0')
```
- Defined in bufferbloat.py

```python
self.addLink('h1','s0', bw=1000)
sself.addLink('h2','s0', bw=1000)
```
Link Speed

• Add asymmetry into the picture
• Command can be found in tc_cmd.sh

Bottleneck link from s0 -> h2

1.5Mb/s, 10ms

s0-eth2
Control Buffer Size

- Command can be found in tc_cmd.sh

Buffer size = 100 pkt

h2

s0-eth2

s0-eth1

h1
Services on Hosts

- Command can be found in bufferbloat.py

h2.cmd('iperf -s')
h1.cmd('python2.7 webserver.py')
Monitor

• Code can be found in exp_monitor.py

Buffer Occupancy  TCP CWND

h2  tc cat /proc/net/tcpProbe

h1