CS 179i:
Project in Computer Science (Networks)

Jiasi Chen

Lectures: 3:10-4pm Watkins 2240

TA: Shahryar Afzal

Lab: Tuesday 4:10-7pm WCH 133

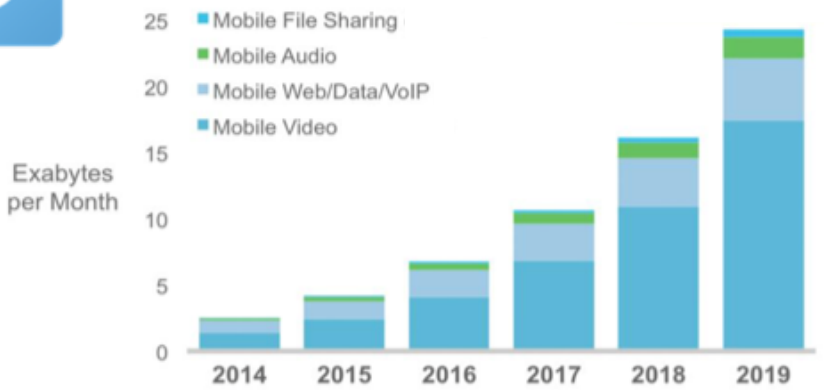
http://www.cs.ucr.edu/~jiasi/cs179i_winter18/

Why Networks?

Supports the applications that we use today…

Social media

Video streaming



Number of Internet users

- 97% of Americans between 18-29
- 40% of the world population → scope for more users

http://www.pewinternet.org/data-trend/internet-use/latest-stats/
https://en.wikipedia.org/wiki/List_of_countries_by_number_of_internet_users

Why Networks?

But also a source of conflict.

Cyber security

A Look Back at the Target Breach

Posted: 04/06/2015 10:30 am EDT | Updated: 06/06/2015 5:59 am EDT

Network neutrality

TECHNOLOGY

T-Mobile Video Plan Could Test F.C.C.'s New Net Neutrality Rules

By CECILIA KANG NOV. 11, 2015

Email
Share
Tweet
Save

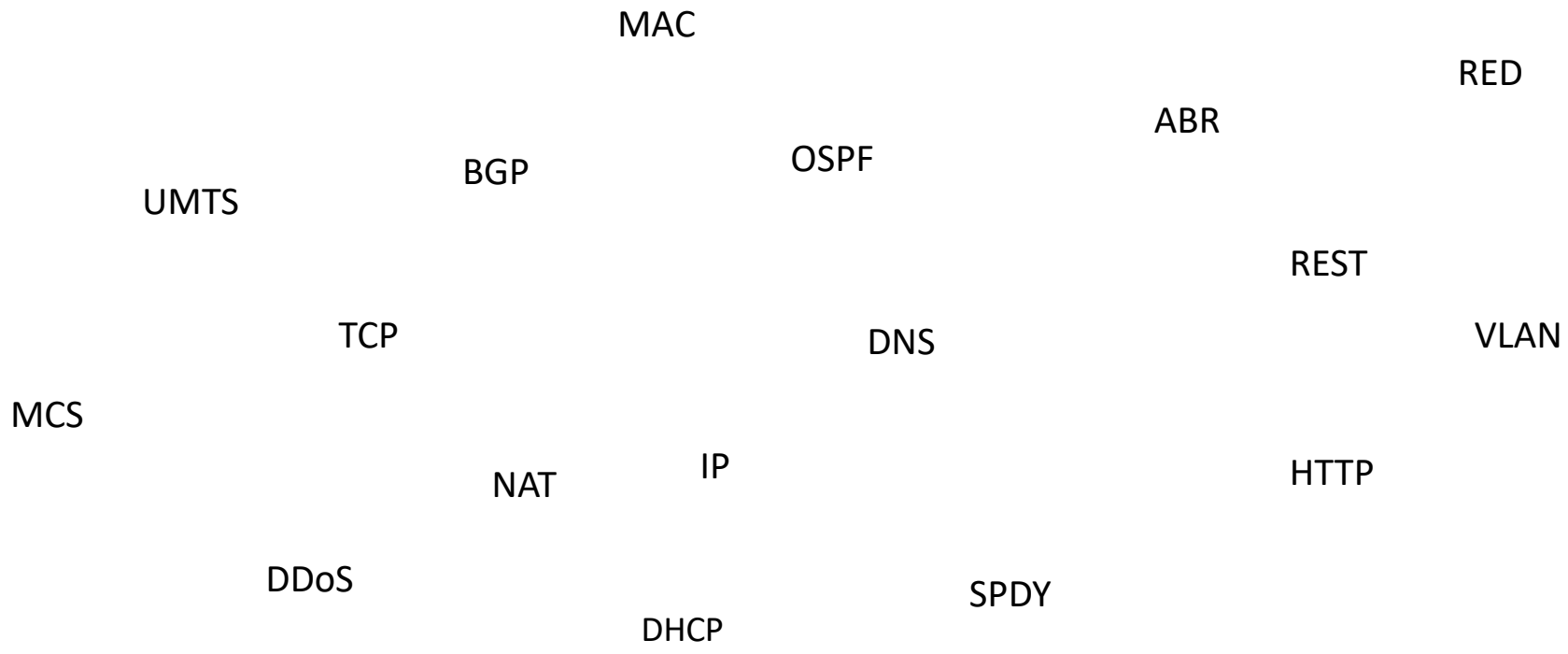
A new plan from T-Mobile USA to allow unlimited streaming of some video services may become the first test of the federal government's rules to prevent favoritism on the Internet.

On Tuesday, T-Mobile, the nation's third-largest wireless carrier, said customers could stream as many videos as they want — regardless of their data plan limits — from more than two dozen video providers, including Hulu and Netflix.

http://www.huffingtonpost.com/eric-dezenhall/a-look-back-at-the-target_b_7000816.html
http://www.nytimes.com/2015/11/12/technology/t-mobile-video-plan-could-test-fccs-new-net-neutrality-rules.html

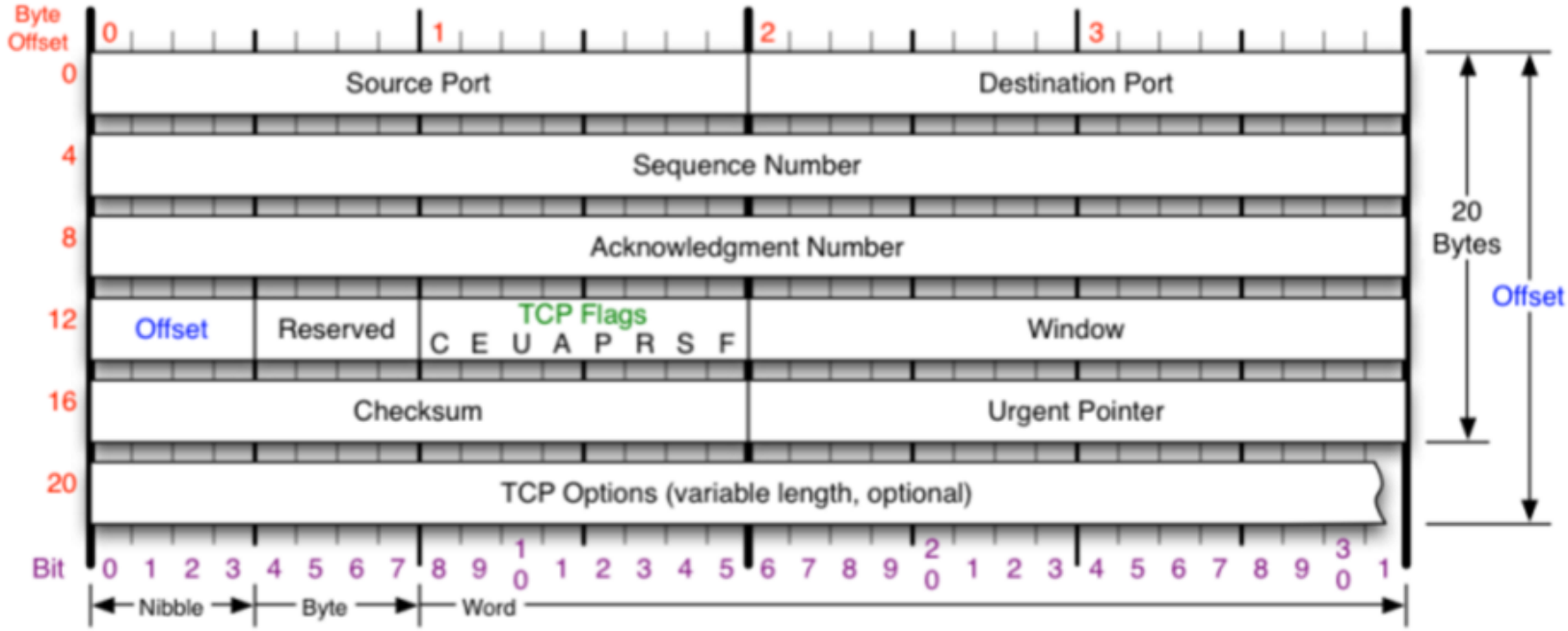
What is networking?

- Bunch of acronyms?



What is networking?

- Bunch of headers?



Source: https://nmap.org/book/tcpip-ref.html

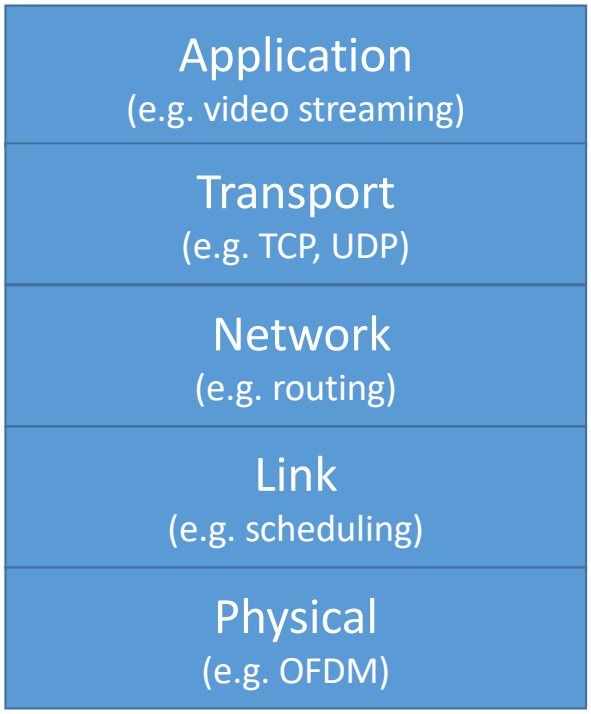
Networking is…

The search for general principles to guide communication

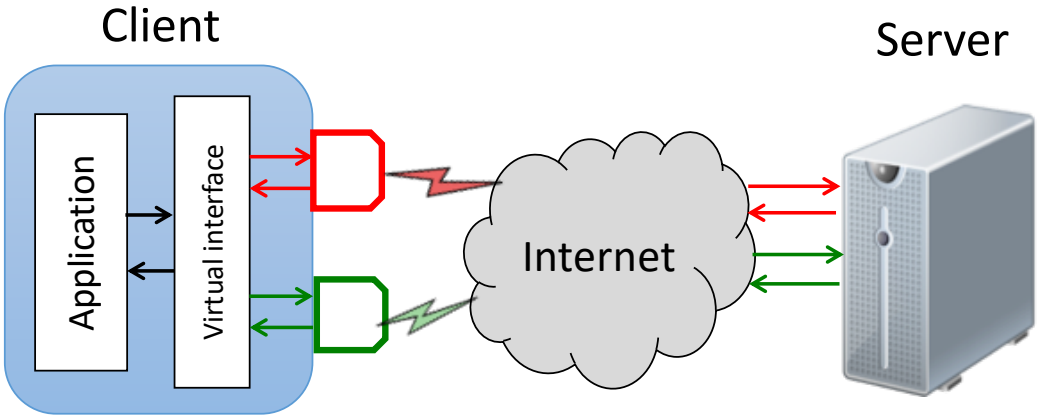
Major Areas in Networking

- Wireless
  - How to provide a one-to-one communication pipe in an inherently broadcast environment?
- Layering
  - How to modularize the design to enable easy innovation?
- Protocols
  - How to interact within each layer, and talk to other layers?
- Resource allocation
  - How to share limited resources between competing users?

OSI 5-layer model of the Internet



Download Booster Using Multiple Interfaces



- Speed up downloads by using multiple interfaces simultaneously (e.g., WiFi, 4G, Ethernet)
- Samsung introduced Download Booster, but it got blocked by major carriers
- Multipath-TCP is another major standardization effort

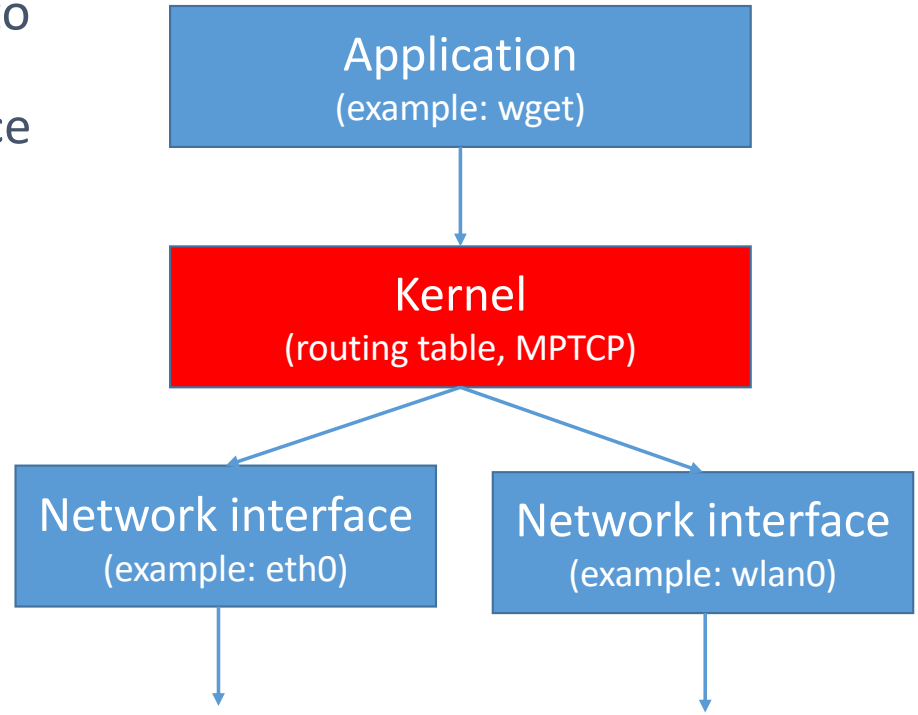
Transport layer multipath: multipath-TCP

- Extension of TCP to split a single flow into multiple subflows
- Each subflow can use a different interface

- Pros: works for TCP traffic
- Cons: kernel modification

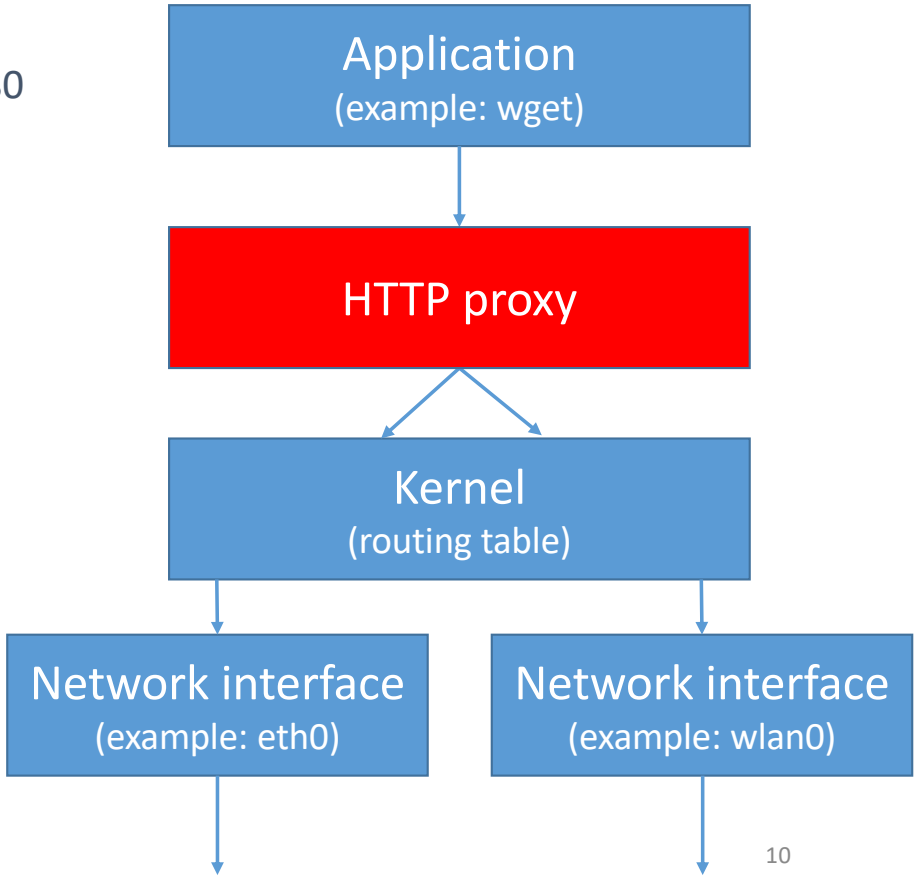
- Control knobs
  - Congestion control
  - Scheduler

- Resources
  - MPTCP kernel: http://www.multipath-tcp.org/



Application layer: HTTP proxy on client

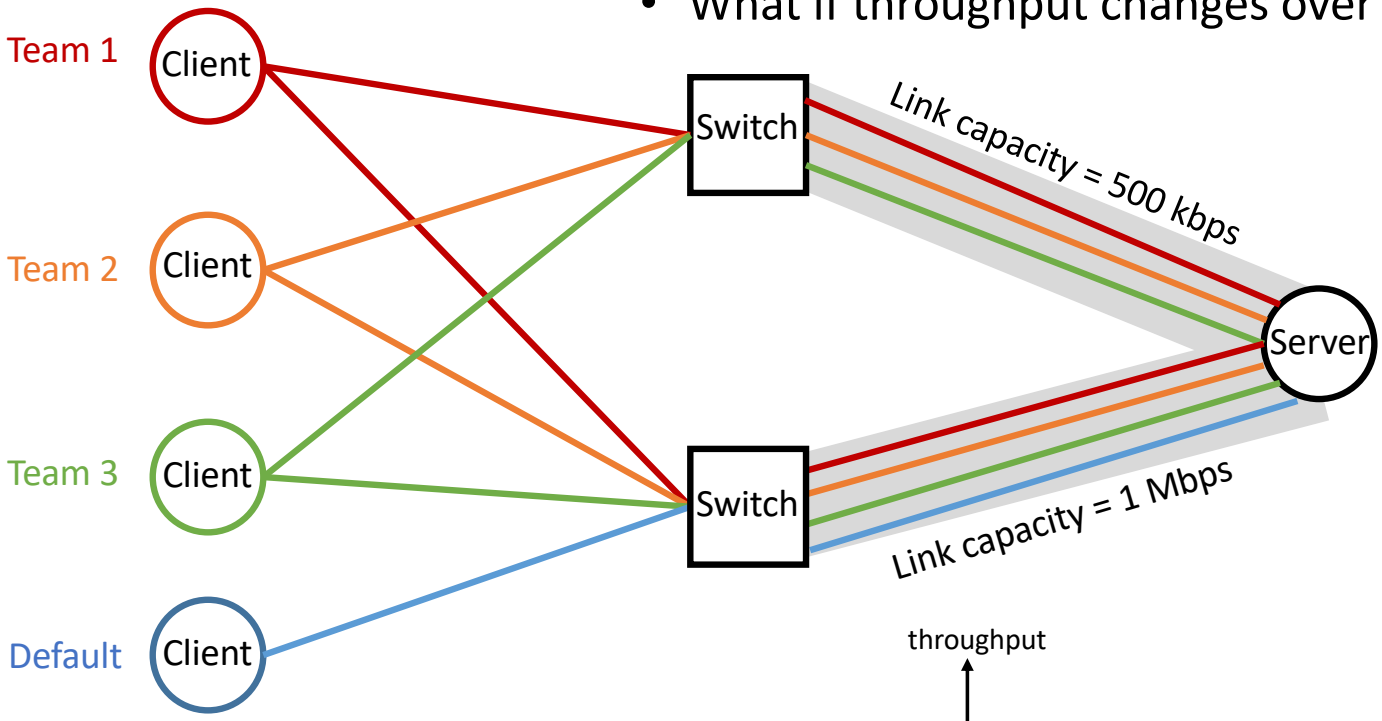
- Construct a local HTTP proxy that listens on port 8080
- Split GET requests
  - eth0: request bytes 1-50
  - wlan0: request bytes 51-100
- Pros: easy to run
- Cons: only works for HTTP traffic
- Control knobs
  - Split 50/50? 25/75? Depends on link bandwidth
  - What if link bandwidth changes over time?
- Resources
  - Simple Python local proxy will be provided



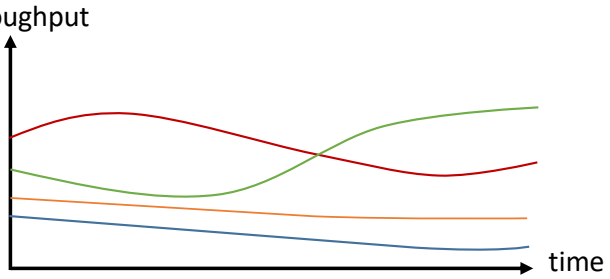
Competition

Metrics

- Throughput (how fast can I download?
- Fairness (how well do I share the link with others?)
- What if throughput changes over time?



Example run:



Your Tasks

1. Install and get familiar with Mininet (small assignment)
2. Install multipath-TCP as a baseline
   - Experiment with different congestion control and schedulers
3. Implement the HTTP proxy
   - Design an algorithms to splitting the traffic
4. Develop a GUI to visualize the results
5. Final demos: head-to-head comparison with your classmates
6. Bonus: Run the proxy in real life (e.g., WiFi + Ethernet)

What You Will Learn in this Course

- Knowledge: Common networking tools/protocols, depending on your choice of project
  - Software-defined networking
  - Multipath
  - Socket programming

- Skills
  - How to work in teams
  - How to lead your own project
  - How to learn on your own

Logistics

- Lecture: Jiasi Chen
  - Slides available on course website
  - Office hours: Thursdays 1-3pm, or by appointment
- Lab: Shahryar Afzal
- Submit assignments on iLearn
- Check class website for latest updates
  - http://www.cs.ucr.edu/~jiasi/cs179i_winter18/

Grading

- Project: 65% total
  - Mininet assignment: 5%
  - Project proposal: 5%
  - Progress update: 10%
  - Final report: 30%
  - Final presentation: 15%
- 4 essays: 20%
  - ABET requirement
  - 2 free late days
- Participation: 15%
  - Attending lecture and lab
  - Giving feedback during other teams' final presentations

Calendar

Week	Lecture	Assignment Due
1	Introduction	
2	MPTCP	Group formation
3	Proxy	Mininet mini-assignment
4	Visualization	New trends essay
5	Progress update / Q&A	Brief (10 minute) presentation per group
6	Ethics	
7	Guest lecture	
8	TBD	Ethics essay
9	Final presentations	
10	Final presentations	Presentation essay
Finals week		Teamwork essay, final report due

To do

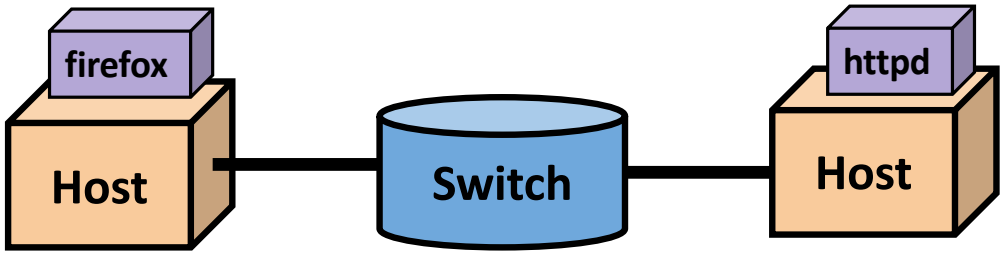
- Next lecture: Mininet
- To do by next class
  - Form groups (2+) and send one email per group to myself and TA
- Questions?

Source: https://conferences.sigcomm.org/sigcomm/2014/doc/slides/mininet-intro.pdf

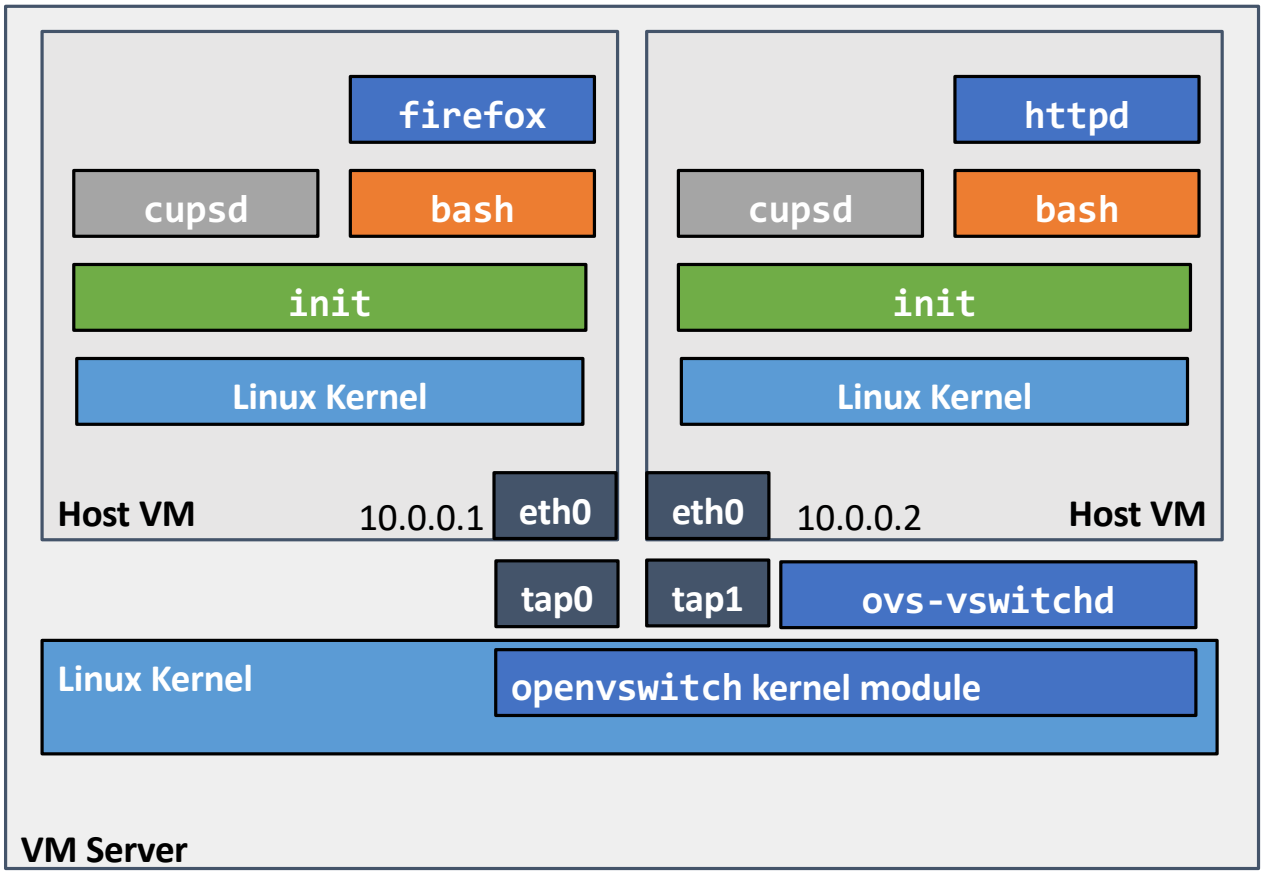
Platforms for Network/Systems Teaching

Platform	Advantages	Disadvantages
Hardware Testbed	fast accurate: "ground truth"	expensive shared resource? hard to reconfigure hard to change hard to download
Simulator	inexpensive, flexible detailed (or abstract!) easy to download virtual time (can be "faster" than reality)	may require app changes might not run OS code detail != accuracy may not be "believable" may be slow/non-interactive
Emulator	**inexpensive, flexible** **real code** **reasonably accurate** **easy to download** **fast/interactive usage**	slower than hardware experiments may not fit possible inaccuracy from multiplexing

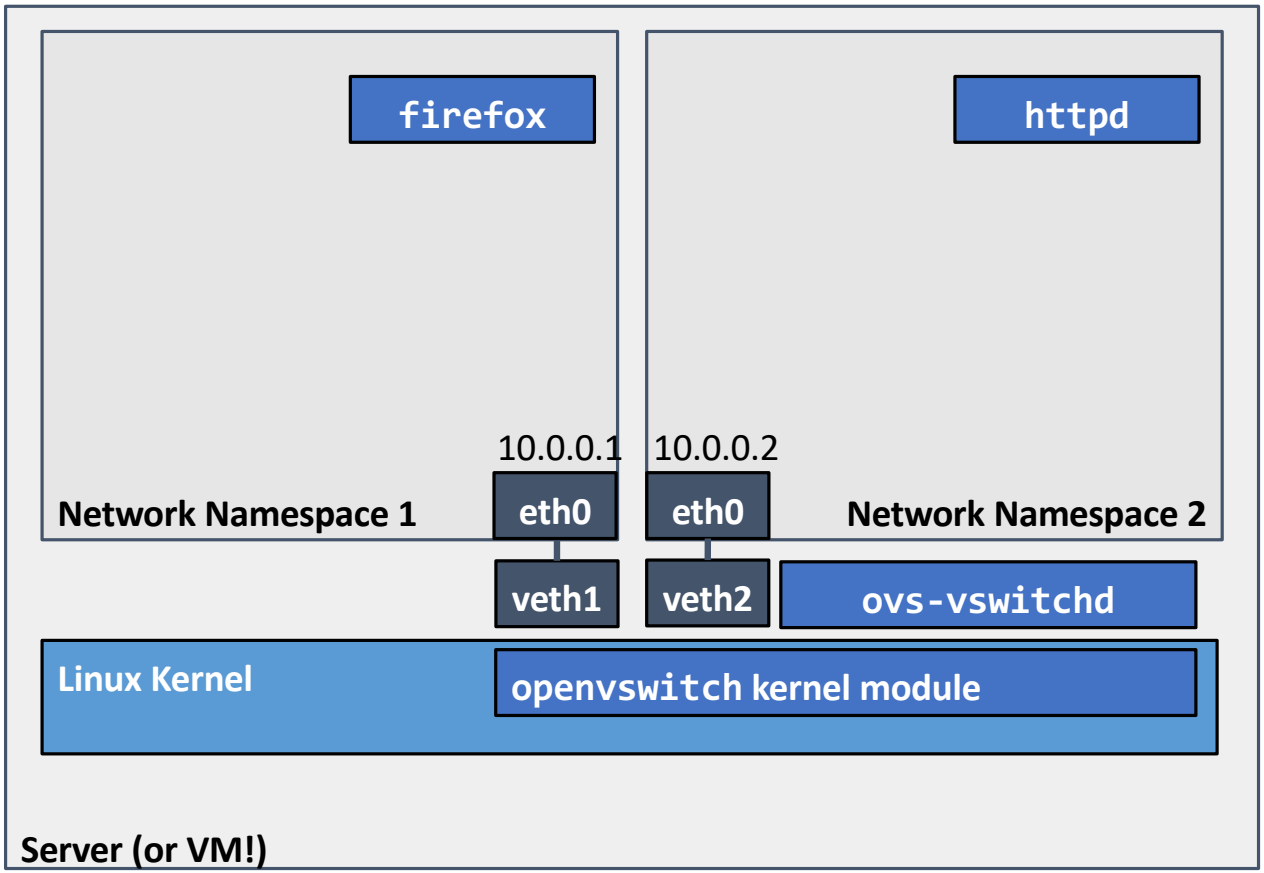
To start with,
a Very Simple Network



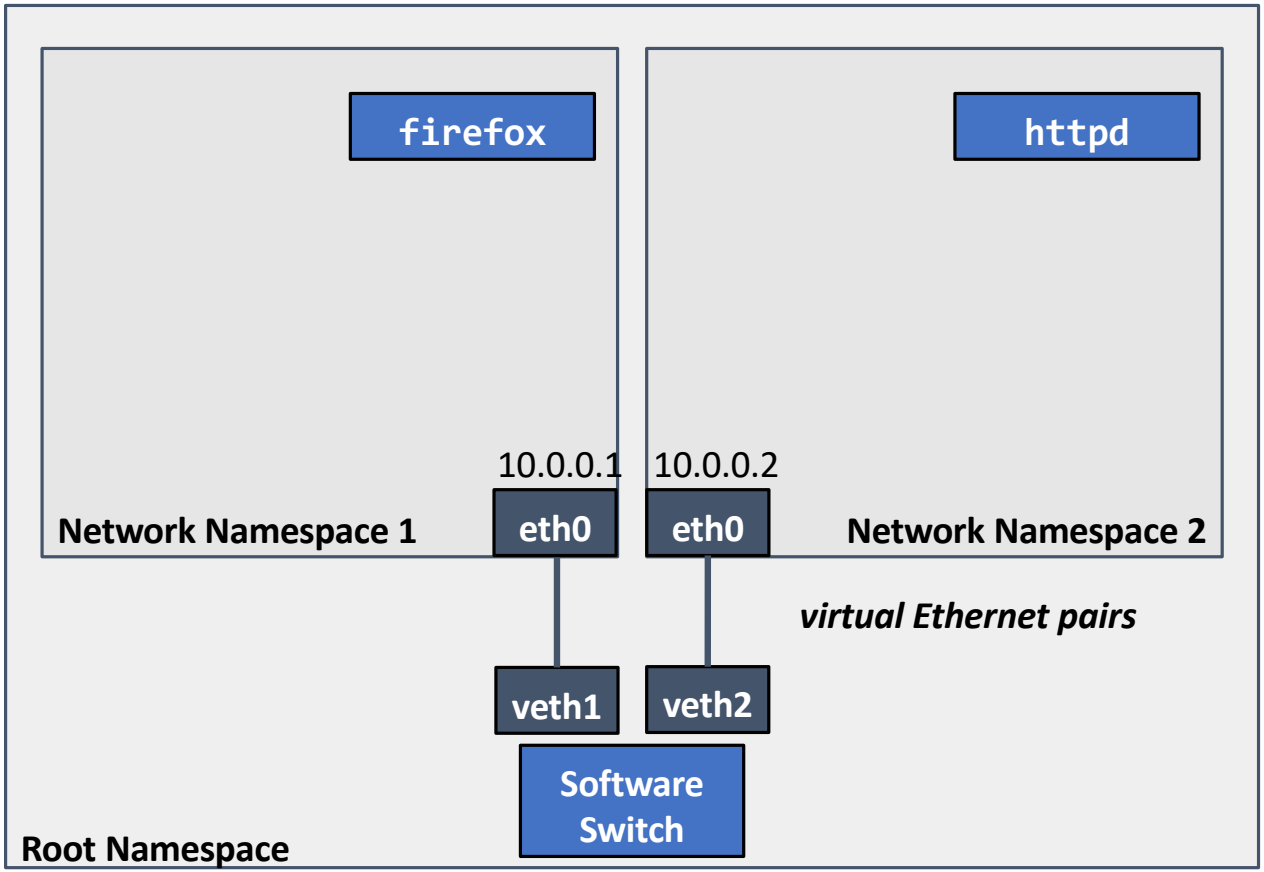
Very Simple Network using Full System Virtualization



Very Simple Network using Lightweight Virtualization



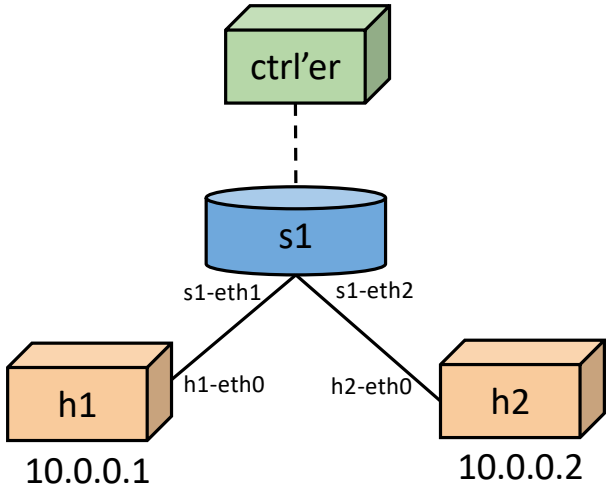
Mechanism: Network Namespaces and Virtual Ethernet Pairs



Creating it with Linux

```
sudo bash
# Create host namespaces
ip netns add h1
ip netns add h2
# Create switch
ovs-vsctl add-br s1
# Create links
ip link add h1-eth0 type veth peer name s1-eth1
ip link add h2-eth0 type veth peer name s1-eth2
ip link show
# Move host ports into namespaces
ip link set h1-eth0 netns h1
ip link set h2-eth0 netns h2
ip netns exec h1 ip link show
ip netns exec h2 ip link show
# Connect switch ports to OVS
ovs-vsctl add-port s1 s1-eth1
ovs-vsctl add-port s1 s1-eth2
ovs-vsctl show
# Set up OpenFlow controller
ovs-vsctl set-controller s1 tcp:127.0.0.1
ovs-controller ptcp: &
ovs-vsctl show
```

```
# Configure network
ip netns exec h1 ifconfig h1-eth0 10.1
ip netns exec h1 ifconfig lo up
ip netns exec h2 ifconfig h2-eth0 10.2
ip netns exec h1 ifconfig lo up
ifconfig s1-eth1 up
ifconfig s1-eth2 up
# Test network
ip netns exec h1 ping -c1 10.2
```



Wouldn’t it be great if…

- We had a simple command-line tool and/or API that did this for us automatically?
- It allowed us to easily create topologies of varying size, up to hundreds of nodes, and run tests on them?
- It was already included in Ubuntu?

Mininet: An Instant Virtual Network on your Laptop (or other PC) – Mininet

mininet.org

Reader

Mininet

An Instant Virtual Network on your Laptop (or other PC)

Mininet creates a **realistic virtual network**, running **real kernel, switch and application code**, on a single machine (VM, cloud or native), in seconds, with a single command:

Because you can easily interact with your network using the Mininet CLI (and API), customize it, share it with others, or deploy it on real hardware, Mininet is useful for development, teaching, and research.

Mininet is also a great way to develop, share, and experiment with OpenFlow and Software-Defined Networking systems.

Mininet is actively developed and supported, and is released under a permissive BSD Open Source license. We encourage you to contribute code, bug reports/fixes, documentation, and anything else that can improve the system!

Get Started

Download a Mininet VM, do the walkthrough and run the OpenFlow tutorial.

Support

Read the FAQ, read the documentation, and join our mailing list, mininet-discuss.

Contribute

File a bug, download the source, or submit a pull request - all on GitHub.

News

Mininet Tutorial at SIGCOMM

Announcing Mininet 2.1.0 !

Nick Feamster's SDN Course

Automating Controller Startup

Display a menu

Mininet command line tool and CLI demo

```
# mn
# mn --topo tree,depth=3,fanout=3 --
link=tc,bw=10
mininet> xterm h1 h2
h1# wireshark &
h2# python -m SimpleHTTPServer 80 &
h1# firefox &
# mn --topo linear,100
# mn --custom custom.py --topo mytopo
```

Mininet's Python API

Core of Mininet!! Everything is built on it.

Python >> JSON/XML/etc.

Easy and (hopefully) fun

Python is used for *orchestration*, but emulation is performed by compiled C code (Linux + switches + apps)

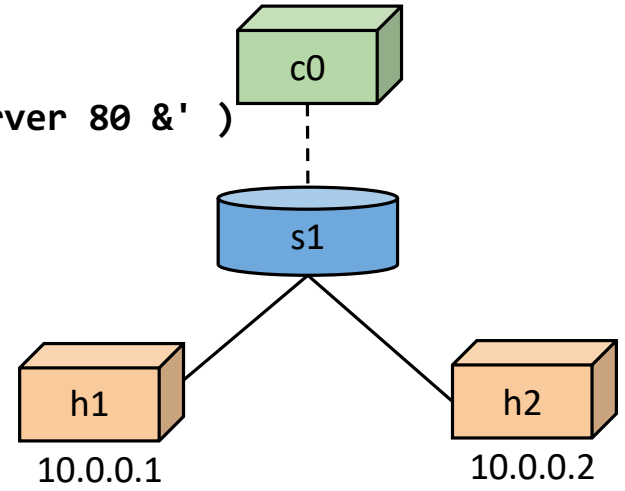
api.mininet.org

docs.mininet.org

Introduction to Mininet

Mininet API basics

```
net = Mininet()                          # net is a Mininet() object
h1 = net.addHost( 'h1' )                 # h1 is a Host() object
h2 = net.addHost( 'h2' )                 # h2 is a Host()
s1 = net.addSwitch( 's1' )               # s1 is a Switch() object
c0 = net.addController( 'c0' )           # c0 is a Controller()
net.addLink( h1, s1 )                           # creates a Link()
object
net.addLink( h2, s1 )
net.start()
h2.cmd( 'python -m SimpleHTTPServer 80 &' )
sleep( 2 )
h1.cmd( 'curl', h2.IP() )
CLI( net )
h2.cmd('kill %python')
net.stop()
```



Performance modeling in Mininet

```
# Use performance-modeling link and host classes
net = Mininet(link=TCLink, host=CPULimitedHost)
# Limit link bandwidth and add delay
net.addLink(h2, s1, bw=10, delay='50ms')
# Limit CPU bandwidth
net.addHost('h1', cpu=.2)
```