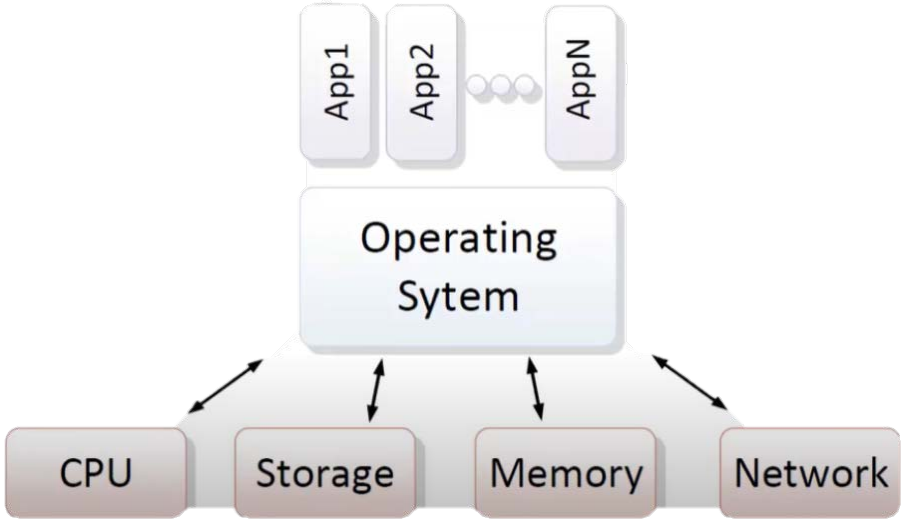
Evaluating Webpage Performance and DNS latency during a DoS attack on SDN Controller

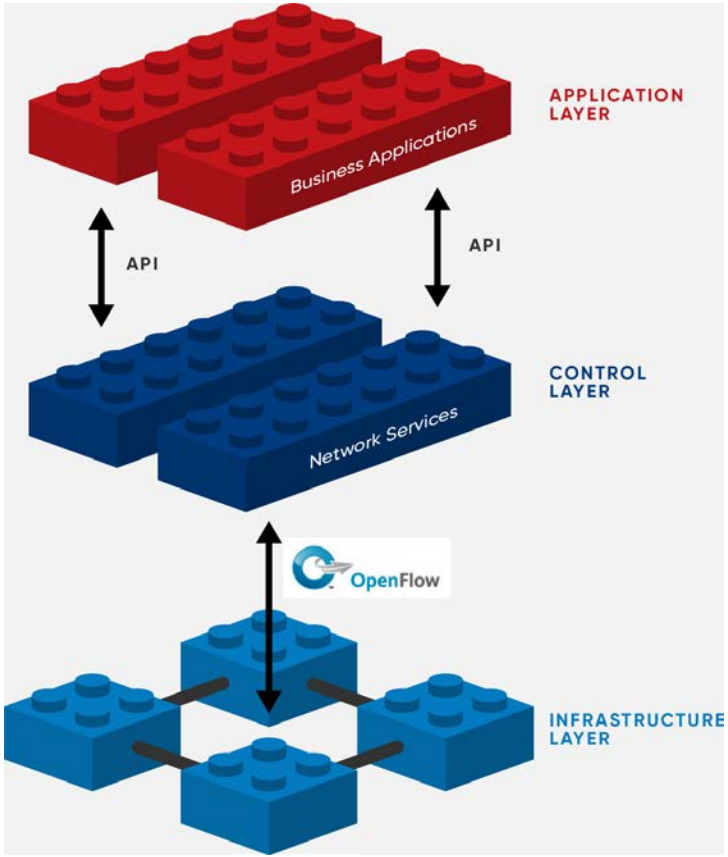
Kevin Zhao | Tandon School of Engineering | CS-GY 6233

Software Defined Networking is…

- Separation of control and data planes
- Centralization at the controller → global view of topology
- Northbound API
  - E.g. RESTConf
- Southbound API
  - We use OpenFlow for updating flows on switching devices/querying flow tables/maintaining communication with controller
- Flows
  - Match (input port, ethernet layer, IP layer, transport layer) + action (forward, drop, modify)
  - Counters
    - # of packets, byte count

An Analogy

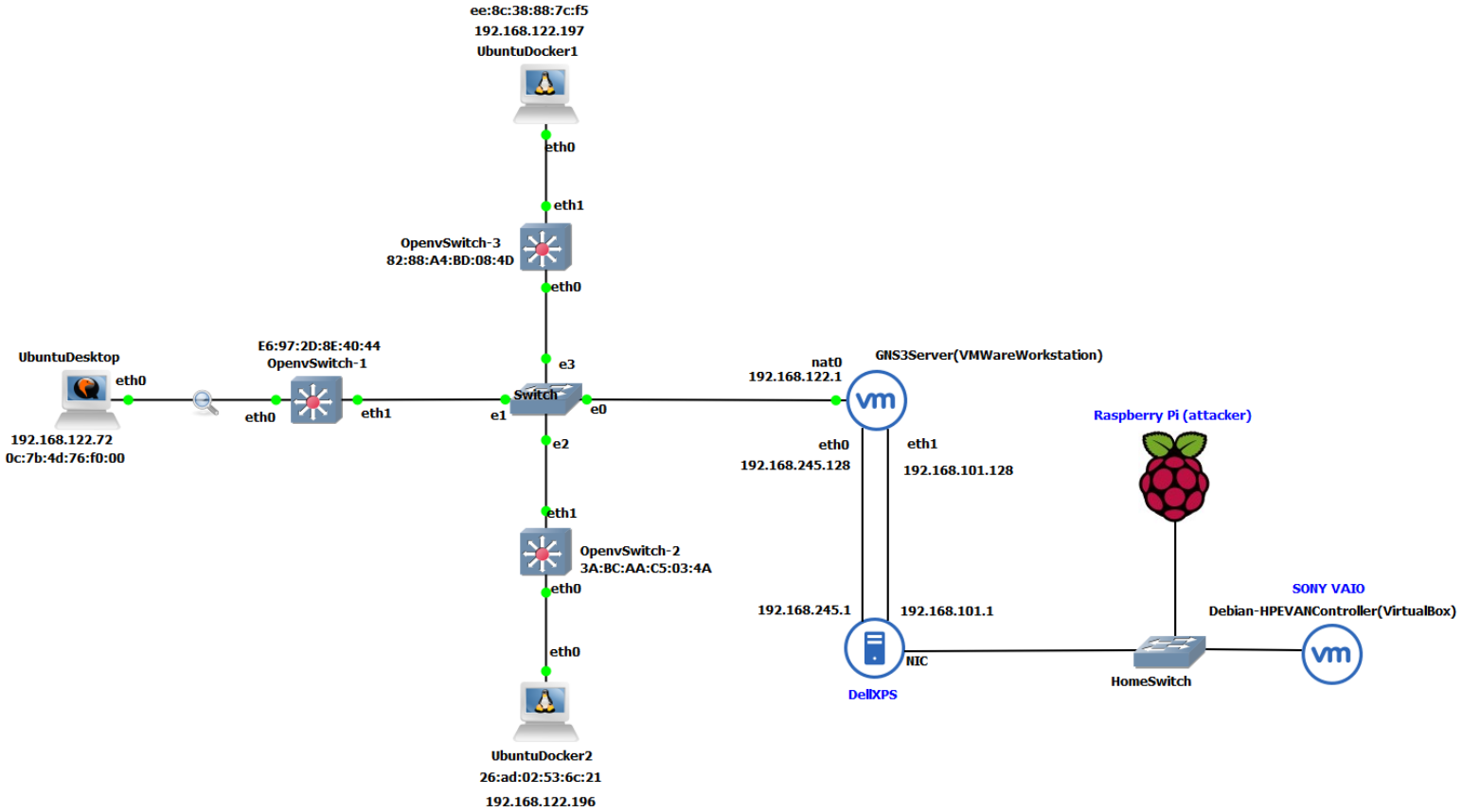

App1
App2
AppN
Operating Sytem
CPU
Storage
Memory
Network


Business Applications
APPLICATION LAYER
API
API
Network Services
CONTROL LAYER
OpenFlow
INFRASTRUCTURE LAYER

Key Contributions

- architecting a basic software-defined network by leveraging GNS3 and VMWare Workstation, a separate host for our controller (HPE VAN version 2.8.8) with Network Protector App, and a Raspberry Pi 4 Model B (primarily used for executing the DoS attack) on a residential network
- evaluating **DNS latency** and impact to **page load times** when browsing to each of the top 50 Alexa site by U.S. Region using Google Chrome before and during OpenFlow flooding attack

Testbed


ee:8c:38:88:7c:f5
192.168.122.197
UbuntuDocker1
eth0
eth1
OpenvSwitch-3
82:88:A4:BD:08:4D
eth0
UbuntuDesktop
eth0
192.168.122.72
0c:7b:4d:76:f0:00
E6:97:2D:8E:40:44
OpenvSwitch-1
eth0
eth1
e3
Switch
e1
e0
e2
nat0
192.168.122.1
GNS3Server(VMWareWorkstation)
eth0
192.168.245.128
eth1
192.168.101.128
Raspberry Pi (attacker)
eth1
OpenvSwitch-2
3A:BC:AA:C5:03:4A
eth0
192.168.245.1
192.168.101.1
SONY VAIO
Debian-HPEVANController(VirtualBox)
eth0
NIC
DellXPS
HomeSwitch
UbuntuDocker2
26:ad:02:53:6c:21
192.168.122.196

NYU

Hypothesis

- DNS resolution will be unaffected unless the flooding throughput can meet—and exceed—all available bandwidth (as opposed to theoretical maximum)
  - Available bandwidth measured using iPerf

Studies on DDoS Mitigation

- Proactive:
  - DELTA blackbox pen-testing framework using “fuzzing” technique to detect vulnerabilities in the SDN stack (Lee et al., 2017)
- Reactive:
  - query flows and/or statistics from each switch across the network and conduct anomaly analysis (Wang, Jia, and Zhu (2015))
  - Module installed on top of controller where traffic samples is sent to a load-balanced, resource scaled pool of VMS and statistics are extrapolated (Miao, R., Yu, M., & Jain, N. (2014))

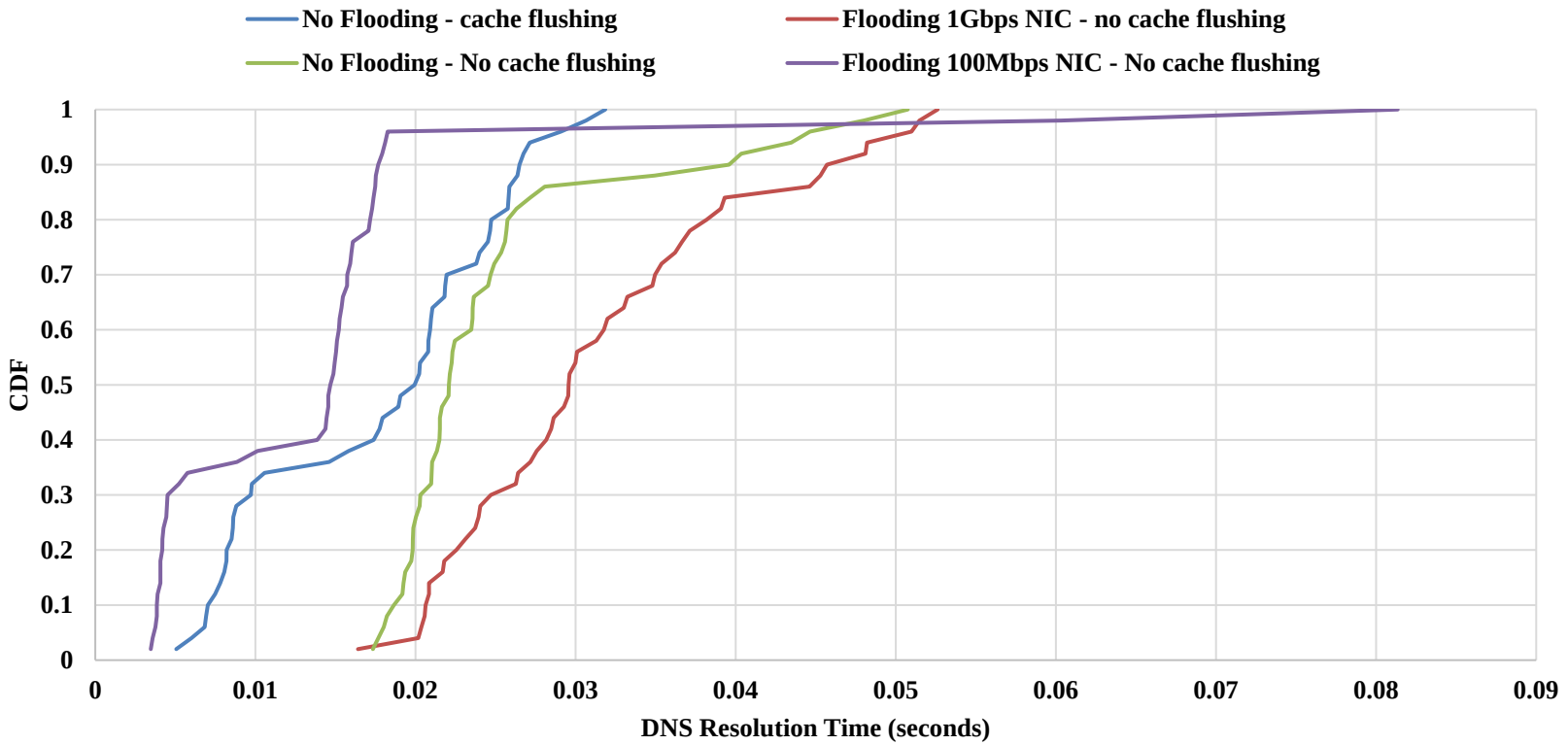
Study on feasibility of SDN in residential network (Taylor, Guo, Shue, and Najd (2017))

- Sourced participants (U.S. based, on residential network) over Amazon Mechanical Turk to take a speed test, which connects to custom AWS, GCP, Azure, and Digital Ocean VMs
- 90% fell within 50ms RTT of two VMs
- PLT measured for top 100 Alexa sites where each fetch for a remote resource must be approved by the controller
- 50 percentile: median page load time increases by ~2s
  - 4s (plain routing/switching) → 6s (OpenFlow+50ms artificially induced latency to controller)
- our study measures page load time, but within the context of a DoS attack and DNS filtering by an SDN application

Papers on Evaluating Controller Performance

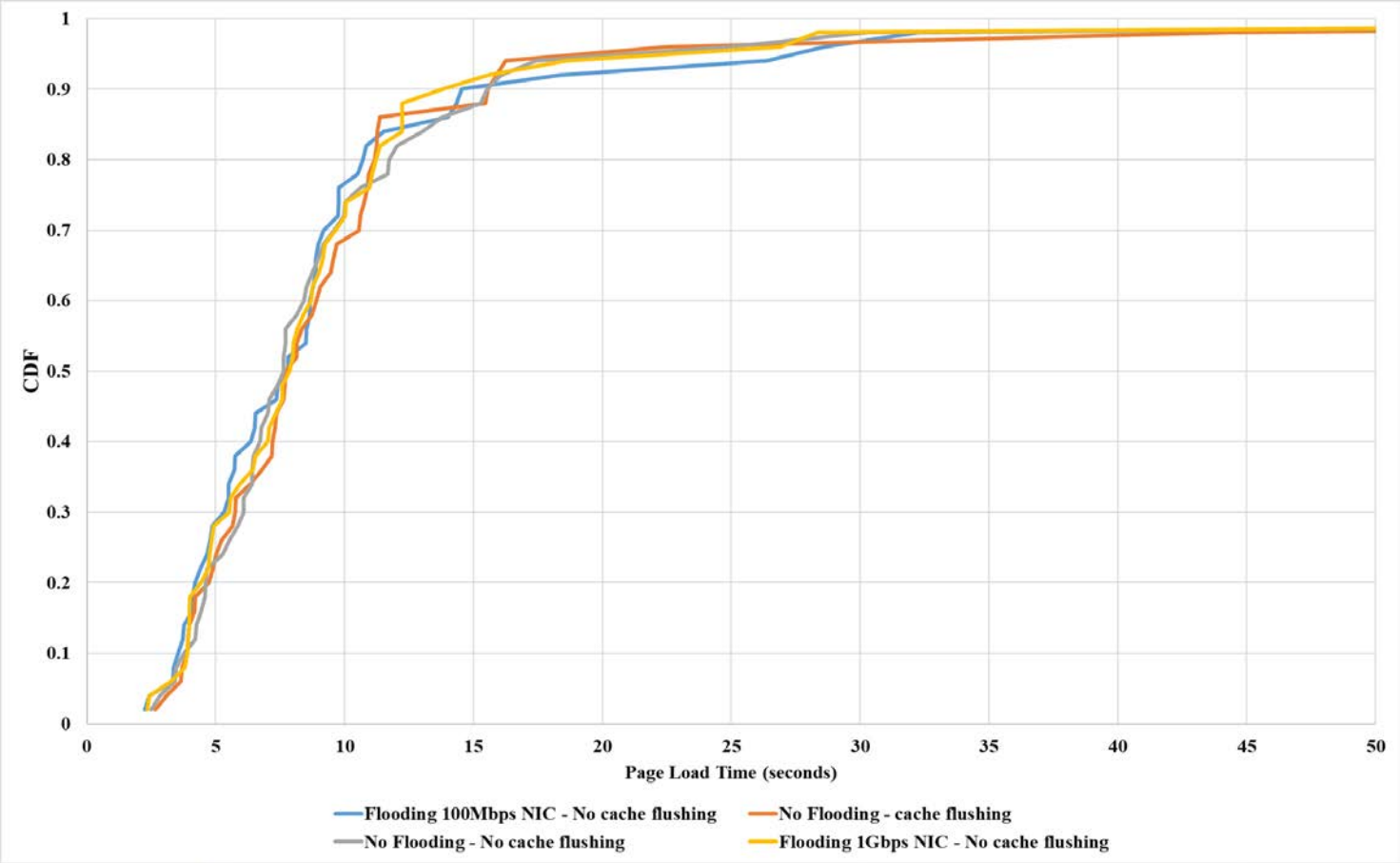
- Badotra and Panda (2019)
  - simulate a hierarchical network topology of 27 hosts and 13 OpenvSwitch switches using Mininet in one Ubuntu VM
  - Two other VMs used to host OpenDayLight and ONOS controllers
- Zhu et al. (2019) – provide a list of benchmarking metrics & tools used to evaluate 9 controllers in different network scenarios
  - E.g. Throughput, latency, and flow installation rate

Results/Discussion



NYU

Results/Discussion



NYU

Conclusion / Future Work

- *Test more than once*
- Attempting a different form of DoS
- More apps; programming flow paths
- Controller failover: a practical deployment will have several distributed ones
- Securing SBI with TLS and testing overhead
- Virtualization/hardware limitations
  - Switching in software: (2 SW, 1 HW in line)
  - Type 2 Hypervisor (VirtualBox)
    - Available bandwidth disparity of 390 Mbps to 930 Mbps