A Scalable Simulation and Modeling Framework for Evaluation of Software-Defined Networking Design and Applications

Dong (Kevin) Jin
CODES Summer Workshop
Argonne National Lab
July 17, 2018
Acknowledgement

• Ph.D. Students

Jiaqi Yan

Christopher Hannon

Xin Liu

Yanfeng Qu

Neil Getty

Xiaoliang Wu
Our Team - Research Interests

- Modeling & Simulation
- Cyber-Security
- Smart Grid Security and Resilience
- Networks
- High Performance Computing
Background: Software Defined Networking (SDN)

Vertically integrated
Closed, proprietary
Slow innovation

Horizontally integrated
Open interfaces
Rapid innovation

Specialized Features
Specialized Control Plane
Specialized Hardware

Merchant Switching Chips

Control Plane
or
Control Plane
or
Control Plane

Open Interface

Source: Nick McKeown, Open Networking Summit 2012
SDN Architecture

Applications
- QoS
- Access Control
- VPN

Control Plane
- OpenFlow Controller

Data Plane
- OpenFlow Protocol
- OpenFlow Switches
- Net 1
- Net 2
- Net 3
- Net 4
- Net 5
- Net 6
SDN Simulation/Emulation Framework

Our Framework
• Parallel simulation kernel
• Light weighted container based emulation
• OpenFlow protocol and device models

Challenges
• Performance fidelity
• Synchronization between two sub-systems
  o Emulation – executing “native” software to produce behavior in wall-clock time
  o Simulation – executing model software to produce behavior in virtual time
Problems of Container Based Emulation

• Performance Fidelity & Scalability
  – A target experiment’s aggregate resource must fit within a single multi-core server [Handigol, 2012]
    • Physical resources: 2.98 GHz of CPU, 4 GB RAM, 3 Gbps of internal packet bandwidth
    • Best case emulation: 30 nodes (100 MHz CPU, 100 MB RAM each) connected by 100 Mbps links

• Temporal fidelity
  – Emulation: VM executions are serialized
  – Physical testbed: distributed hosts running in parallel
Performance Fidelity Issues in Mininet

![Diagram of a network topology with a server and client connected by a series of routers.]

- Bar graph showing average TCP throughput (Gbps) vs. link bandwidth (Gbps) for Mininet and Physical Testbed.
  - Link Bandwidth: 1, 2, 4, 8, 10 Gbps
  - Mininet: 1.0, 1.9, 3.8, 5.9, 9.2 Gbps
  - Physical Testbed: 1.0, 1.9, 3.8, 5.7, 8.2 Gbps

- Line graph showing average TCP throughput (Gbps) vs. number of switches (#Switches), with a line indicated as 'Line Rate'.
  - #Switches: 10, 20, 40, 60, 80
  - Throughput: 3.8, 3.8, 3.7, 2.2, 1.3 Gbps
Our approach: Virtual Time

• Key idea: trade execution time with fidelity
• Time dilation factor (TDF) [Gupta, 2011]

\[
TDF = \frac{\text{time passing rate in the physical world}}{\text{time passing rate in a VM's perception of time}}
\]

• TDF = 10
  – 10 seconds in real time = 1 second in a time-dilated emulated host
  – a 100 Mbps link is scaled to a 1 Gbps link

Virtual Time System Architecture for a Container-based Network Emulator

Source code: https://github.com/littlepretty/VirtualTimeForMininet
Virtual Time is Useful

1. Emulation Fidelity Enhancement
2. Simulation/Emulation Synchronization
Use Case: ECMP Evaluation

- Equal-cost multi-path (ECMP)
  - balances traffic load over multiple paths
  - By performing a modulo-N hash over the selected fields of a packet's header

- Communication bottleneck
  - when several large and long-lived flows collide on their hash

Use Case: ECMP Evaluation

- Experiment Setup
  - Fat-tree topology with degree 4
  - SDN open vSwitches connected to Ripl-POX controller (hash-based ECMP traffic routing)
  - Generate collisions with stride traffic pattern
    - host_x sends to host_(x+i mod n), i = {1,4}
Use Case: ECMP Evaluation

- **Experiment Setup**
  - Fat-tree topology with degree-4
  - SDN open vSwitches connected to RipL-POX controller (hash-based ECMP traffic routing)
  - Generate collisions with stride traffic pattern
    - `host_x` sends to `host_(x+i mod n)`, `i = {1,4}`

**stride step = 1**
Use Case: ECMP Evaluation

• Experiment Setup
  – Fat-tree topology with degree-4
  – SDN open vSwitches connected to RipL-POX controller (hash-based ECMP traffic routing)
  – Generate collisions with stride traffic pattern
    • host_x sends to host_(x+i mod n), i = {1,4}

stride step = 4
Use Case: ECMP Evaluation

100 Mbps link

10 Gbps link
Conclusions and Future Work

• SDN testing framework
  – virtual time system
  – improve scalability and performance fidelity
  – github.com/littlepretty/VirtualTimeForMininet

• Applications
  – Energy systems
    • combined with power system simulator [DSSNet]
    • load shifting, self-healing PMU networks
  – Blockchain testing system
2019 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (PADS)

• Illinois Institute of Technology (IIT), Chicago
• June 2–5, 2019
  – PhD Colloquium: June 2 (Sunday afternoon)
  – Main conference: June 3-5 (Monday to Wednesday noon)
    • Paper deadline: January 7, 2019
    • Regular and short papers
    • Special issue in TOMACS
  – Dinner on a CRUISE and enjoy sunset