Software-Defined Networks

Jennifer Rexford
Princeton University
Traditional Networks

control plane: distributed algorithms
data plane: packet processing
Software Defined Networks

decouple control and data planes
Software Defined Networks

decouple control and data planes by providing open standard API
Simple Data-Plane API

• Prioritized list of rules
  – Pattern: match packet header bits
  – Actions: drop, forward, modify, send to controller
  – Priority: disambiguate overlapping patterns
  – Counters: #bytes and #packets

1. srcip=1.2.*.*, dstip=3.4.5.* → drop
2. srcip=.*.*.*, dstip=3.4.*.* → forward(2)
3. srcip=10.1.2.3, dstip=.*.*.*.* → send to controller
(Logically) Centralized Controller
Protocols ➔ Applications

Controller Application
Controller Platform
Seamless Mobility

• See host sending traffic at new location
• Modify rules to reroute the traffic
Server Load Balancing

- Pre-install load-balancing policy
- Split traffic based on source IP

```plaintext
src=0*,
dst=1.2.3.4
src=1*,
dst=1.2.3.4
```
Middlebox Traffic Steering

• Direct selected traffic (e.g., port 80)
• … through a chain of middleboxes

\[
\text{dstip} = 1.2.3.4 \\
\text{dstport} = 80 \\
\text{dstip}=1.2.3.4
\]
Example SDN Applications

- Seamless mobility and migration
- Server load balancing
- Steering traffic through middleboxes
- Dynamic access control
- Using multiple wireless access points
- Energy-efficient networking
- Blocking denial-of-service attacks
- Adaptive traffic monitoring
- Network virtualization
- <Your app here!>
A Major Trend in Networking

• SDN components
  – Switches: Open vSwitch, hardware switches, etc.
  – Controllers: ONOS, Floodlight, Ryu, Frenetic, ⋯

• Commercial successes
  – Google’s private backbone
  – Nicira’s network virtualization platform

• Industry consortia
  – Open Networking Foundation (ONF)
  – Open DayLight (ODL)
  – Open Compute Project (OCP)
Example Research Areas
Languages and Verification

- **Languages**
  - Abstractions for apps
  - Compilation to switches

- **Verification**
  - Data-plane invariants
  - Control-plane correctness
Distributed Controllers

- Scalability, reliability, and performance
- Managing controller state or replicas
- Aggregating information about the network
More Sophisticated Switches

- OpenFlow 1.0
  - Single rule table and twelve header fields
- OpenFlow 1.3/1.4
  - Multiple match-action stages on different headers
- OpenFlow 2.0 (?)
  - Reconfigurable parsing and match-action tables
- White-box/bare-metal switches
  - Program the switch directly
Network Function Virtualization

- Network functions
  - Firewall, intrusion detection, NAT, transcoder, compression, proxy cache, monitoring, ...
- Virtualized
  - Virtual machines that can run anywhere
- Challenges
  - Optimization (placement, steering, routing)
  - Platforms for hosting virtualized functions
  - Control protocols for managing the functions
SDN Security

• Securing the entire stack
  – Switches
  – Control protocol
  – Controller platform
  – Controller apps

• Example attacks/vulnerabilities
  – Worst-case traffic to DoS the controller
  – Rogue apps that violate user privacy
  – Compromising the controller platform
New Applications of SDN

- Cloud
  - Data centers
  - Private backbones
- Other networks
  - Enterprise
  - Cellular
  - Home
  - Exchange points
  - Optical networks
- Hybrid deployments
  - Overlay (SDN edge, legacy core)
  - Mix of SDN and legacy devices
- Beyond networking
  - Software Defined Infrastructure
  - Network, middleboxes, storage, compute, …
Conclusions

• SDN is two main ideas
  – Logically centralized controller
  – Standard APIs to the data plane
• SDN is happening in practice
  – Protocol standards and white-box networking
  – Wide variety of switch and controller platforms
  – Real operational deployments
• Clean-slate research opportunity
  – … while still influencing the practice