

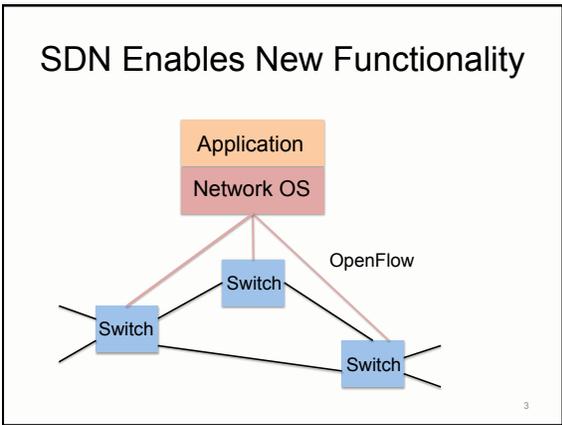
Testing and Debugging

COS 597E: Software Defined Networking

Jennifer Rexford
Princeton University
MW 11:00am-12:20pm

Challenges for SDN Programmers

2



... at the Risk of Bugs

Network Operating System

A fatal exception has occurred at 10.3.0.5/C0011E36 in OF(01) + 00010E36. The current OpenFlow application will be terminated.

* Press any key to terminate the current OpenFlow application
* Press CTRL+ALT+DEL again to restart your network. Your users will lose all network connectivity.

Press any key to continue

4

MAC-Learning Application

Packet-In Event Handler

```

mactable[srcmac] = inport
if (dstmac is broadcast address)
  flood packet
else if (dstmac is in mactable)
  outport = mactable[dstmac]
  install rule matching (inport, srcmac, dstmac)
  with action of forwarding packet to outport
else
  flood packet
    
```

5

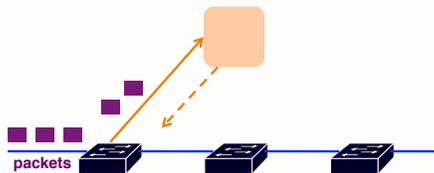
Challenge #1: Two-Tiered Program

- Two-tiered application
 - Controller
 - Switches
- Rules in the switches
 - Crucial for performance
 - ... but limits visibility
- Example
 - MAC learning bug from Monday's class
 - Matching on dstmac, without inport and srcmac

6

Challenge #2: Control-Plane Delays

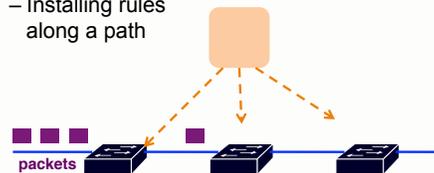
- Common programming idiom
 - First packet goes to the controller
 - Controller application installs rules



7

Challenge #3: Timing and Ordering

- Multiple switches
 - Different delays
- Example
 - Installing rules along a path



8

Challenge #4: Overlapping Rules

- Semantics of a rule depend on context
 - Overlapping patterns
 - Disambiguated by priorities
- Example
 - Initial rule matching `srcip==12.1.0.0/16`
 - Add a rule matching `srcip==12.0.0.0/8`
- Two scenarios
 - Overlap: `12.1.0.0/16` with higher priority
 - Shadowing: `12.0.0.0/8` with higher priority

9

Challenge #5: Conflicting Modules

- Modular programs
 - Multiple modules for different tasks
 - E.g., firewall and routing
- Routing
 - `Match(dstip=12.0.0.0/8) → forward(3)`
- Firewall
 - `Match(srcip=1.2.3.4, dstip=12.1.1.1) → drop`
- One rule may conflict with another

10

Challenge #6: Topologies

- Many different network topologies
 - Chain
 - Tree
 - Arbitrary graph
- Program should work for all topologies
- Example
 - Program that (implicitly) assumes the graph has no cycles

11

Challenge #7: End-to-End Protocols

- Internet applications are robust to errors
 - Retransmission of lost packets
 - Reordering of out-of-order packets
- This can mask some bugs
 - E.g., forgetting to handle the packet that triggered an event
 - E.g., forgetting to handle packets that arrive before you install rules in the switches

12

Discussing the Papers

NICE and ndb

13

Testing vs. Debugging

- Debugging
 - Fixing a known problem in your program
 - Locating and fixing bugs
 - E.g., using tools like gdb
- Testing
 - Convincing yourself that your program (probably) works
 - Systematically finding inputs that lead to incorrect outputs

14

Discussion

- Debugging
 - What debugging features are useful for SDN programmers?
 - How can we exploit OpenFlow capabilities to support debugging?
- Testing
 - How to overcome the scalability challenges?
 - How to detect performance bugs?
- Preventing bugs
 - How can we change the programming environment to prevent bugs?
 - How can we change the programming environment to make testing easier?

15

MAC-Learning Application

Packet-In Event Handler

```
mactable[srcmac] = inport
if (dstmac is broadcast address)
  flood packet
else if (dstmac is in mactable)
  outport = mactable[dstmac]
  install rule matching (inport, srcmac, dstmac)
  with action of forwarding packet to outport
else
  flood packet
```

16