SDN Enables New Functionality

- Application
- Network OS
- Switch
- OpenFlow

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

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

... at the Risk of Bugs

Network Operating System

Press any key to terminate the current application
This will close all windows and free up your resources.
Press any key to continue
Challenge #2: Control-Plane Delays

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

Challenge #3: Timing and Ordering

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

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

Challenge #5: Conflicting Modules

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

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

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
Discussing the Papers

NICE and ndb

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

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?

MAC-Learning Application

Packet-In Event Handler

```python
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
```