Innovation in the P4 “Stack”

Jennifer Rexford
Princeton University
"Classic" OpenFlow (1.x)

SDN Control Plane

Installing and querying rules

Target Switch
Simple API as a Magnet

• Simple, open interface
  – OpenFlow 1.0 table of match-action rules
  – … with a small set of headers and actions

• Useful recruiting aid
  – Programming languages
  – Verification

• Many success stories
  – Higher-level control abstractions
  – … and efficient analysis and compilation
Examples: Static Policies

• Data-plane policy as a function
  – Input: located packet (header fields and location)
  – Output: a set of located packets

• Data-plane verification
  – Checking that the function satisfies invariants
  – E.g., no loops, no blackholes, access control, …
Examples: Combining Apps

- **Slicing**
  - Multiple tenants
  - Traffic isolation

- **Composition**
  - Modular applications
  - On the same traffic
Examples: Network Topology

• Abstract topology
  – Controller apps see a virtual topology
  – E.g., one big switch

• Network Info Base
  – Network data model
  – Distributed key-value store

Firewall, Load balancer

C1  C2  C3
Examples: Reading/Writing State

• Queries
  – Traffic and topology
  – Declarative
  – Modular

• Consistent updates
  – Transition from one policy to another
  – …while preserving key invariants
All taking OpenFlow 1.x “as is”…

… but now we can take these lessons and design a *better* interface
P4

SDN Control Plane

Configuring: Parser, tables, and control flow

Populating: Installing and querying rules

Compiler

Parser & Table Configuration → Rule Translator

Target Switch
Many New Opportunities

- **Applications**
  - Compelling P4 apps
  - To stress-test the P4 design
  - … and identify gaps

- **Language**
  - Expression language for primitive actions
  - Support for modular programs
  - More flexible control flow
  - Unambiguous specification
  - QoS mechanisms, monitoring
Many New Opportunities

- Compiler
  - Exploiting opportunities for concurrency
  - Techniques for proving program equivalence
  - Making effective use of the target switch’s resources
  - Updating a switch from one P4 program to another
  - Supporting a wide range of target switches
Collaboration Opportunities

• Crossing the divide
  – Industry with research
  – Computer networking with other fields

• Building a community
  – Open language
  – Open-source software
  – Repository of example P4 applications
  – Tutorials, workshops, hackathons, …
  – Summer internships for grad students
The Day Ahead

- 9:00-11:00: P4 community
- 11:00-11:30: Break/demos
- 11:30-12:30: Panel on user perspectives
- 12:30-1:30: Lunch/demos
- 1:30-3:00: Language evolution & research
- 3:00-3:30: Break/demos
- 3:30-5:00: Applications & targets
- 5:00-5:30: Wrap-up
- 5:30-6:30: Reception/demos
Demos

- **Xilinx**: P4 for an FPGA target
- **Barefoot**: Inband Network Telemetry in P4
- **Dell**: Tunnel splicing with visibility and monitoring
- **Intel**: P4 Applied to a vSwitch Data Plane
- **Microsoft**: Verification of P4 programs
- **Netronome**: P4 for Network Flow Processors
- **Yale**: Magellan: Compiling Datapath-Oblivious Packet Processors to P4
- **Rocker**: P4 switch support in Rocker
- **USC**: In-network layer-4 load balancing in P4
- **ON.Lab**: An ONOS controller for a P4 switch
- **Princeton**: Utilization Aware Load Balancing using P4
Thank you!