Programmable Networking: P4→NetFPGA Workflow

Stephen Ibanez, Nick McKeown
Electrical Engineering, Stanford University, sibanez@stanford.edu

P4 Background

- Programming language to describe packet processing
- First appeared in SIGCOMM CCR 2014
- Three goals: (1) Reconfigurability in the field, (2) Protocol independence, (3) Target independence
- Maintained by nonprofit organization: P4.org
- Emerged as *de facto* standard for packet processing

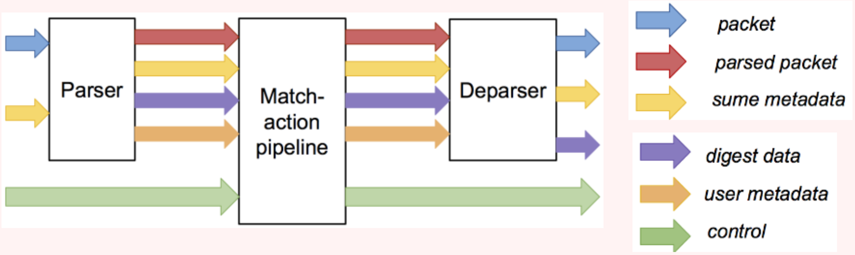
NetFPGA Background

- Line-rate, flexible, open networking platform for teaching and research
- Community began with Stanford and Xilinx Labs, now anchored at Cambridge
- NetFPGA systems deployed at over 150 institutions in over 40 countries
- Four elements: (1) Community: NETFPGA.org, (2) Low-cost board family, (3) Tools & reference designs, (4) Contributed projects

NetFPGA SUME 4x10G ports

P4→NetFPGA Workflow

- Makes it easy to run P4 programs on NetFPGA SUME
- No need for prior experience with HDL
- Built on top of Xilinx P4-SDNet
- Free for academic users
- Active community of developers and people willing to help
- Self guided tutorials, library of extern functions, runtime control API
- Example applications that have been prototyped:
    - Distributed congestion control
    - In-band Network Telemetry
    - In-network compression; In-network key-value cache
    - IPv4 router
    - Network-accelerated sorting; Network-accelerated consensus
- Documentation: https://github.com/NetFPGA/P4-NetFPGA-public/wiki

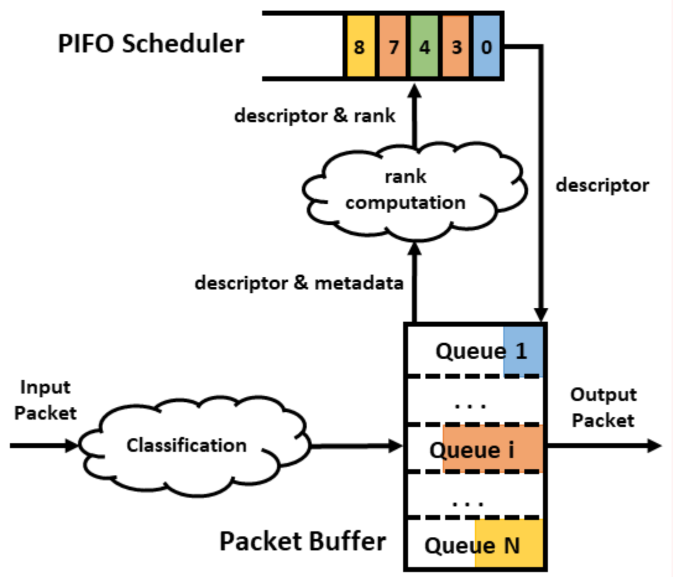


Course CS344: Build an Internet Router

- Students work in teams of two to build a fully functioning Internet router on NetFPGA SUME
- Python Control Plane: ARP, error handling, OSPF-lite routing protocol
- P4 Data Plane: IPv4 forwarding, ARP cache, statistics
- Week 6 – Interoperability Testing
- Week 10 – Open-ended design challenge demos
- Course website: https://build-a-router-instructors.github.io/
- Entire course available for copy/reuse on GitHub

Programmable Traffic Management

- Traffic management: packet scheduling, shaping, policing, drop policy, replication, buffering
- Very important for dealing with diverse traffic requirements and network operator objectives
- Scheduling and shaping cannot be described in P4 today
- Propose programmable TM architecture and P4 extensions based on PIFO model [1]
- Implemented prototype on NetFPGA SUME:



References

[1] Sivaraman, Anirudh, et al. "Programmable packet scheduling at line rate." Proceedings of the 2016 ACM SIGCOMM Conference. ACM, 2016.