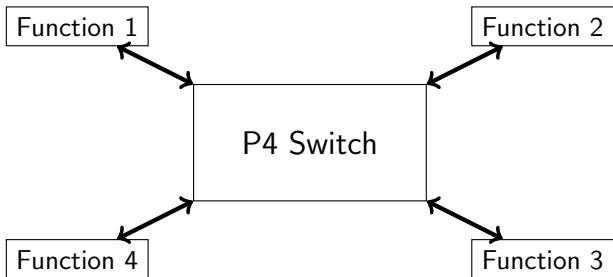# Hybrid P4 Switch

Ashkan Aghdai
Yang Xu
Jonathan Chao

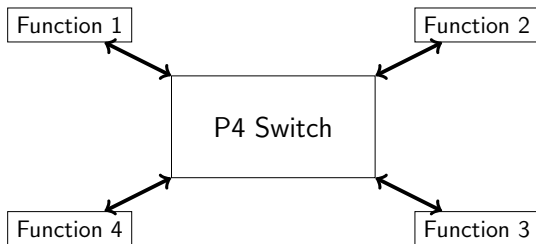**NEW YORK UNIVERSITY**

Tandon School of Engineering

May 17, 2017

NFV using P4

- Offload complex functions to X86 servers
  - P4 Switch: L2, L3
  - Servers: L4 and up

- QoS
  - Priority classes
  - SLAs for different types of traffic
- Stateful packet processing
- Can we bring processing to the data instead?

The Wedge model [FP15]

- P4-compatible Wedge Switch [FP15]
  - + P4 compatibility
  - + Add more microservers to virtualize network functions?
  - + Modify scheduling?
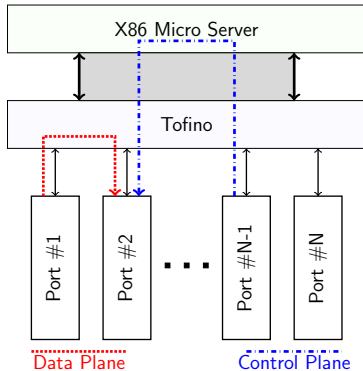
A Better Solution for NFV at Network' Edge

	Wedge w/Tofino	X86 NetVM
Programmability	P4	DPDK
Implementation	Hardware ASIC	Software Commodity Server
Throughput	O(1Tbps)	O(100Gbps)

Can we get the best of both worlds?

- P4 as DSL
- DPDK for Network Functions
- Modularity

Compromise on Throughput

- O(100Gbps) for P4 and DPDK paths

Programmable Packet Forwarding
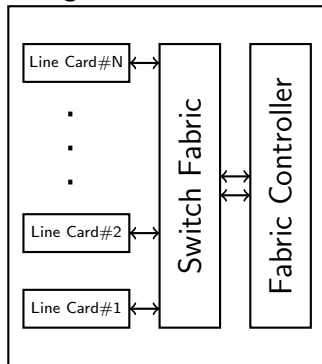
Let's make a programmable input-buffered switch

- Packet Processing
  Table look-ups and header updates
  Programmable Match+Action Tables

- Packet Switching
  Copy from ingress to egress port/s

- Packet Scheduling
  Orchestrate packet transfers

Programmable Switch

Proposed Architecture
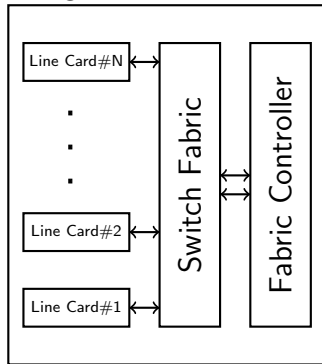
Let's make a programmable
input-buffered switch

- Line Cards
  Smart NICs
- Switch Fabric
  PCI Express
- Fabric Controller
  CPU orchestrates packet transfers
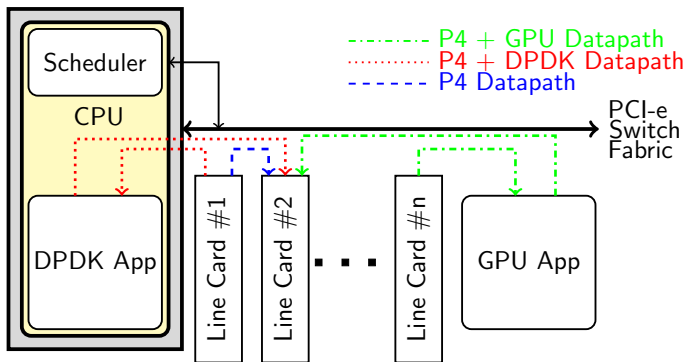 Small bi-partite matching problem

Programmable Switch

Flexible Datapaths

Introduction
  Motivation
Proposed
Design
  Ingredients
Switch
Architecture
  Challenges
  Conclusion
References