

# A Hands-on Tutorial on P4 Programmable Data Planes

Jorge Crichigno  
University of South Carolina  
<http://ce.sc.edu/cyberinfra>  
[jcrichigno@cec.sc.edu](mailto:jcrichigno@cec.sc.edu)

Internet2 Technology Exchange

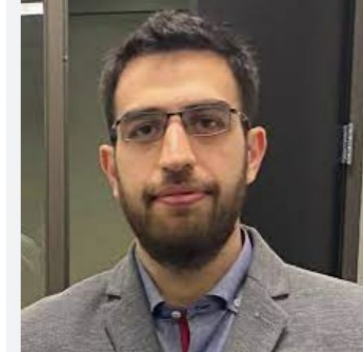
Monday December 5<sup>th</sup>, 2022  
Denver, Colorado

# Organizers

---



Jorge Crichigno  
Faculty



Elie Kfoury  
PhD Student



Ali AlSabeh  
PhD Student



# Workshop Website

---

- All material is posted on the website of the tutorial  
[http://ce.sc.edu/cyberinfra/workshop\\_2022\\_ie2\\_p4.html](http://ce.sc.edu/cyberinfra/workshop_2022_ie2_p4.html)

# Agenda

---

Time	Topic	Presenter
01:30 - 01:55	Overview of P4 and programmable data plane switches Introduction to packet parsing	Jorge Crichigno
01:55 - 02:30	Hands-on Session 1: intro to P4 and BMv2; writing a parser, compiling, and downloading P4 code to the switch	Elie Kfoury
02:30 - 2:45	Break	
02:45 - 03:05	Using registers to store arbitrary data	Jorge Crichigno
03:05 - 03:40	Hands-on Session 2: Calculating packets interarrival times using hashes and registers	Elie Kfoury
03:40 - 03:55	Break	
03:55 - 04:30	Hands-on Session 2 (continue): Calculating packets interarrival times using hashes and registers	Elie Kfoury

# Overview of P4 and Programmable Data Plane Switches

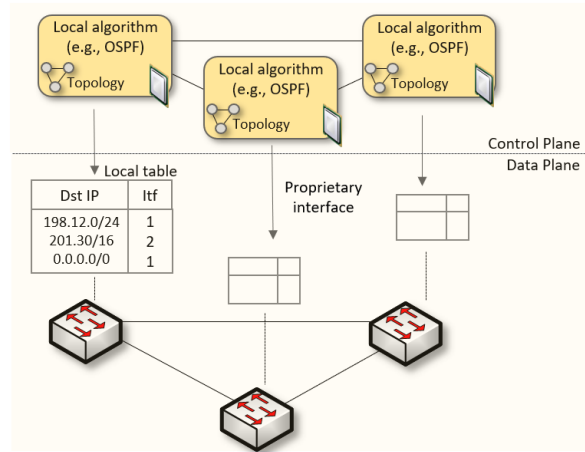
Jorge Crichigno  
University of South Carolina  
<http://ce.sc.edu/cyberinfra>  
[jcrichigno@cec.sc.edu](mailto:jcrichigno@cec.sc.edu)

Internet2 Technology Exchange

Monday December 5<sup>th</sup>, 2022  
Denver, Colorado

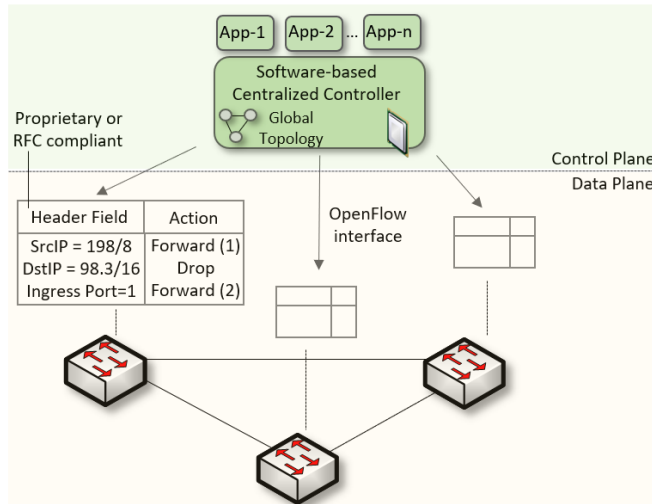
# Traditional (Legacy) Networking

- Since the explosive growth of the Internet in the 1990s, the networking industry has been dominated by closed and proprietary hardware and software
- The interface between control and data planes has been historically proprietary
  - Vendor dependence: slow product cycles of vendor equipment, no innovation from network owners
  - A router is a monolithic unit built and internally accessed by the manufacturer only



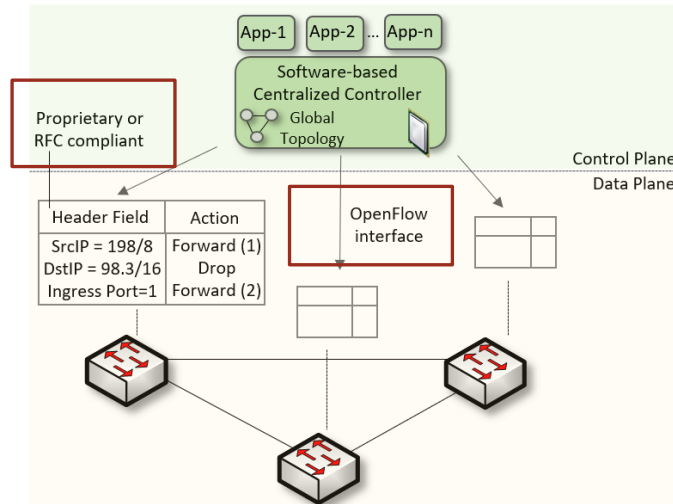
# SDN

- Protocol ossification has been challenged first by SDN
- SDN (1) explicitly separates the control and data planes, and (2) enables the control plane intelligence to be implemented as a software outside the switches
- The function of populating the forwarding table is now performed by the controller



# SDN Limitation

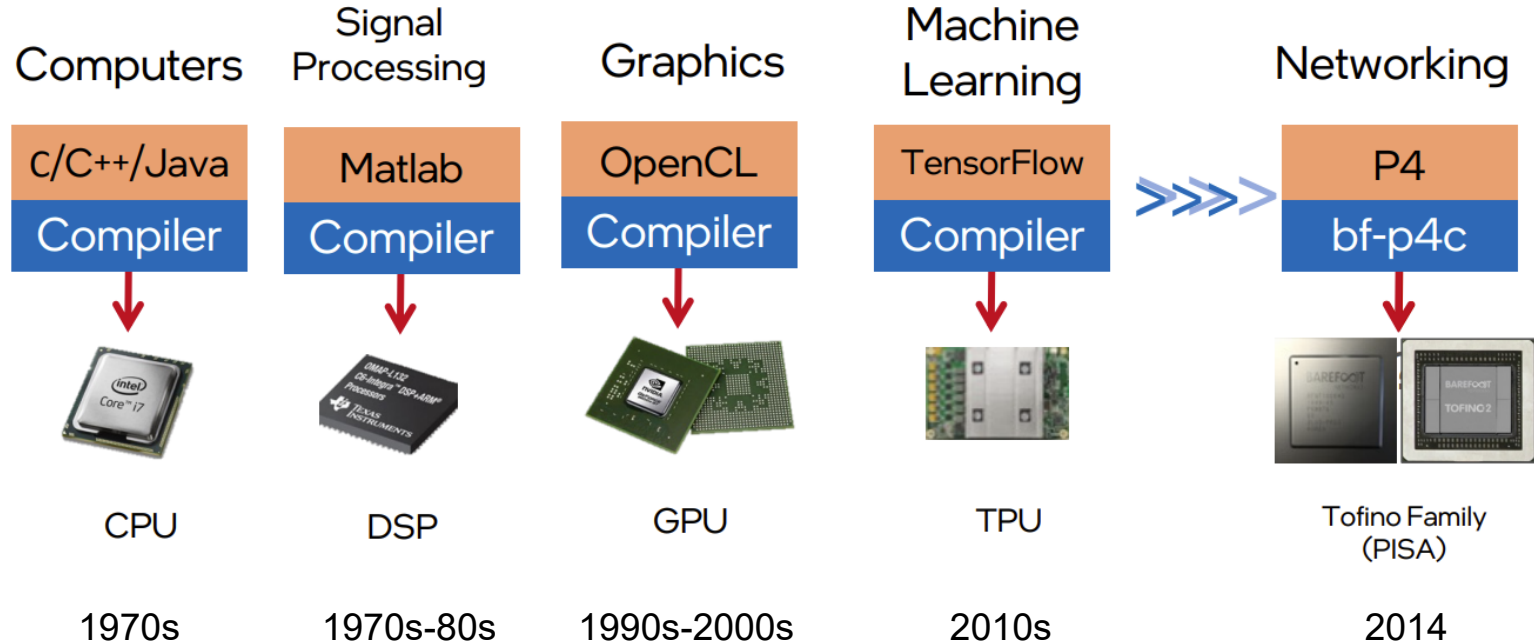
- SDN is limited to the OpenFlow specifications
  - Forwarding rules are based on a fixed number of protocols / header fields (e.g., IP, Ethernet)
- The data plane is designed with fixed functions (hard-coded)
  - Functions are implemented by the chip designer





# Can the Data Plane be Programmable?

- Evolution of the computing industry



1. Vladimir Gurevich, "Introduction to P4 and Data Plane Programmability," <https://tinyurl.com/2p978tm9>.

# Can the Data Plane be Programmable?

- Data plane comparison: fixed-function vs P4 programmable



64 x 100GE  
Legacy,  
Fixed Function ASIC

Parameter	Measurement Unit	Comparison
Throughput	Packets/s	21% higher
Power Consumption	Switching Troughput/W (pps/W)	53% lower
Table Scale	ACL, NAT, tunnels	20x
	Routes (IPv4/IPv6)	10x
	ECMP	2x
Non-standard Application Support	Smart Load balancing	∞
	Segment routing	∞
	In-band Telemetry	1000x

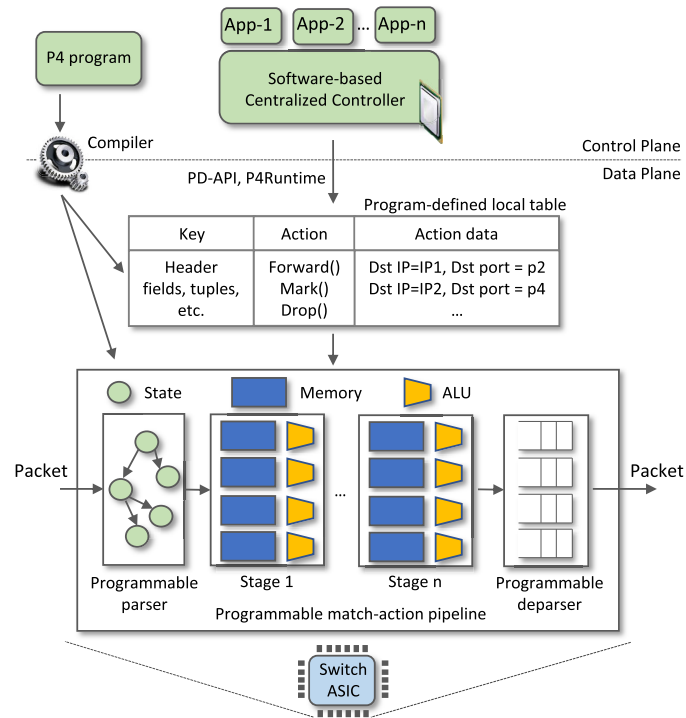


64x100GE  
Barefoot Tofino

1. Vladimir Gurevich, "Introduction to P4 and Data Plane Programmability," <https://tinyurl.com/2p978tm9>.

# P4 Programmable Switches

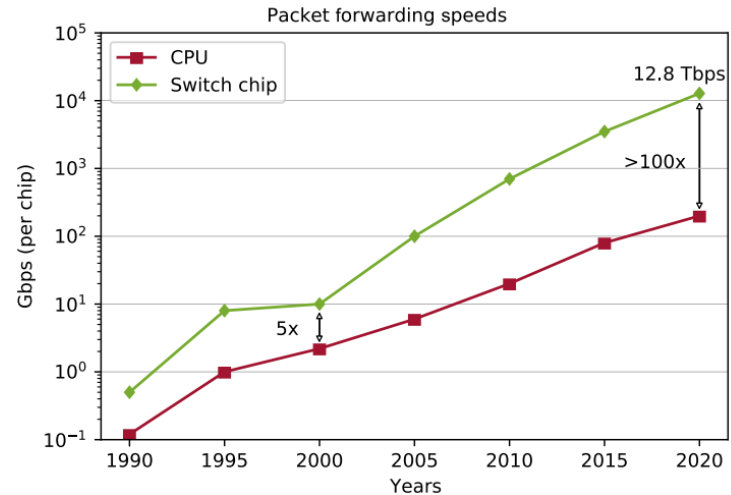
- P4<sup>1</sup> programmable switches permit a programmer to program the data plane
  - Define and parse new protocols
  - Customize packet processing functions
  - Measure events occurring in the data plane with high precision
  - Offload applications to the data plane



1. P4 stands for stands for Programming Protocol-independent Packet Processors

# P4 Programmable Switches

- P4<sup>1</sup> programmable switches permit a programmer to program the data plane
  - Define and parse new protocols
  - Customize packet processing functions
  - Measure events occurring in the data plane with high precision
  - Offload applications to the data plane



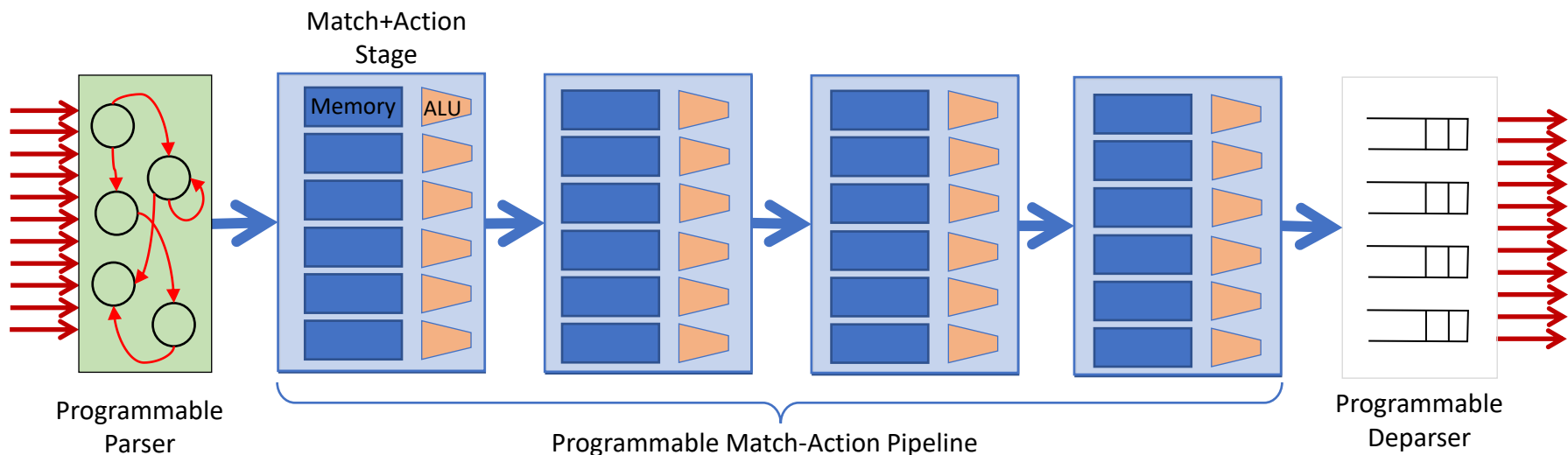
Reproduced from N. McKeown. Creating an End-to-End Programming Model for Packet Forwarding.  
Available: <https://www.youtube.com/watch?v=fiBuao6YZI0&t=4216s>

# Generalized Forwarding: Match + Action

---

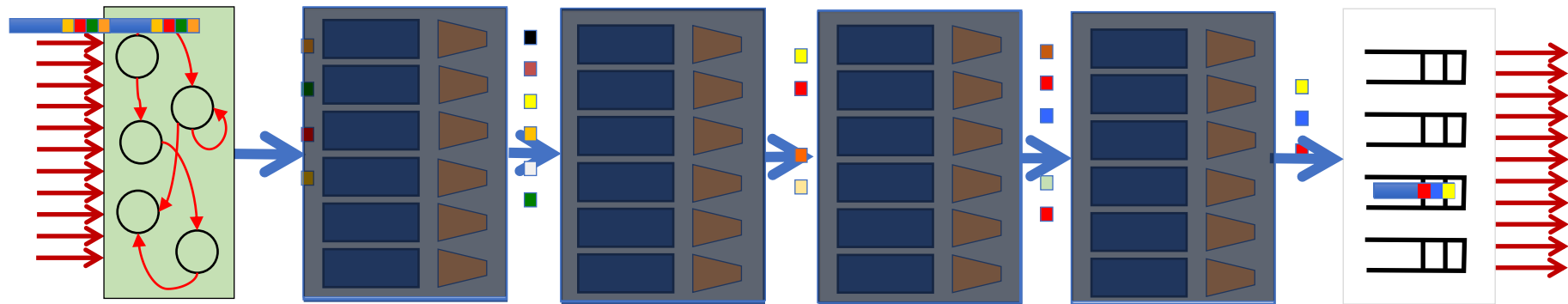
- The data plane contains table/s
  - Match bits in arriving packet (match phase)
  - Take action - Many header fields can determine action (action phase)
    - Drop
    - Copy
    - Modify
    - Log packet
    - Forward out a link (destination-based forwarding is just a particular case)

# PISA: Protocol Independent Switch Architecture



Reproduced from N. McKeown. Creating an End-to-End Programming Model for Packet Forwarding.  
Available: <https://www.youtube.com/watch?v=fiBuao6YZI0&t=4216s>

# PISA: Protocol Independent Switch Architecture



Reproduced from N. McKeown. Creating an End-to-End Programming Model for Packet Forwarding.  
Available: <https://www.youtube.com/watch?v=fiBuao6YZI0&t=4216s>

# Example P4 Program

## Parser Program

```
parser parse_ethernet {  
  extract(ethernet);  
  return switch(ethernet.ethertype) {  
    0x8100 : parse_vlan_tag;  
    0x0800 : parse_ipv4;  
    0x8847 : parse_mpls;  
    default: ingress;  
  }  
}
```

## Header and Data Declarations

```
header_type ethernet_t { ... }  
header_type l2_metadata_t { ... }  
  
header ethernet_t ethernet;  
header vlan_tag_t vlan_tag[2];  
metadata l2_metadata_t l2_meta;
```

## Tables and Control Flow

```
table port_table { ... }  
  
control ingress {  
  apply(port_table);  
  if (l2_meta.vlan_tags == 0) {  
    process_assign_vlan();  
  }  
}
```

