

Overview of P4 and Programmable Data Plane Switches

Jorge Crichigno
University of South Carolina
<http://ce.sc.edu/cyberinfra>
jcrichigno@cec.sc.edu

CENIC 2022 Annual Conference

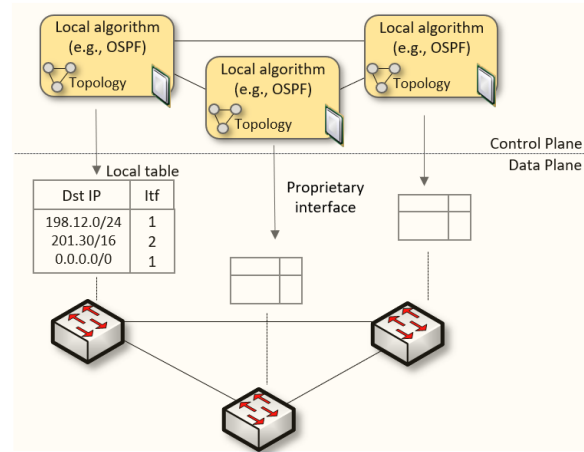
Wednesday September 28th, 2022
Monterey, California

Agenda

- Motivation
 - Limitations of traditional devices
 - Limitations of Software Defined Networking (SDN)
- Data plane programmability
 - Evolution of computing industry
 - Fixed-function vs programmable
- Essentials of P4 programmable switches
 - Advantages
 - Generalized forwarding: Match + Action
 - Protocol Independent Switch Architecture (PISA)

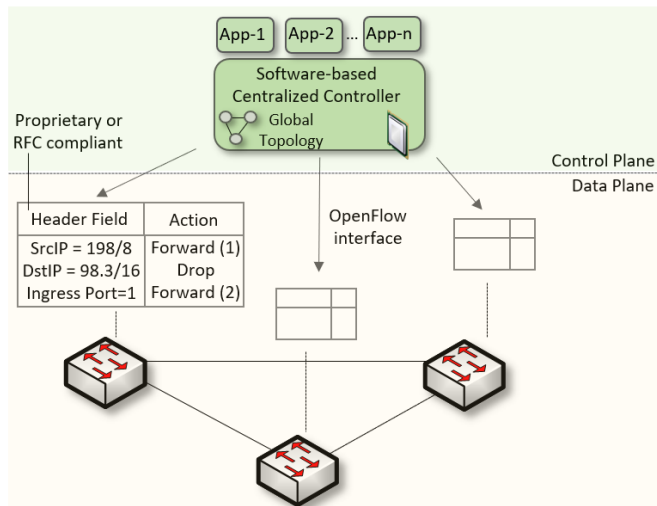
Traditional (Legacy) Networking

- 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



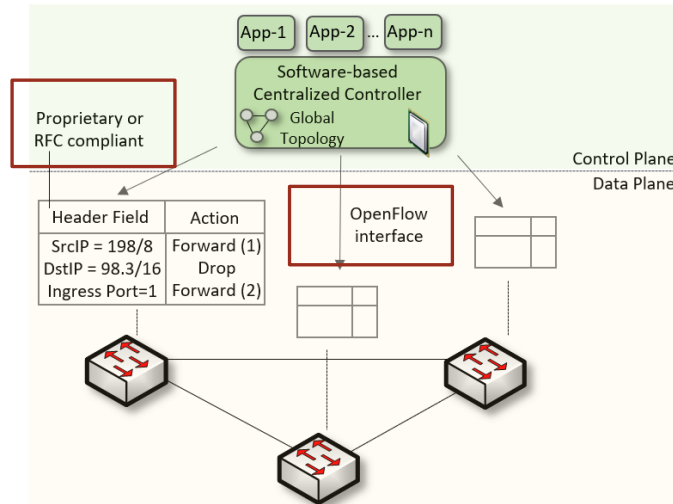
SDN

- 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



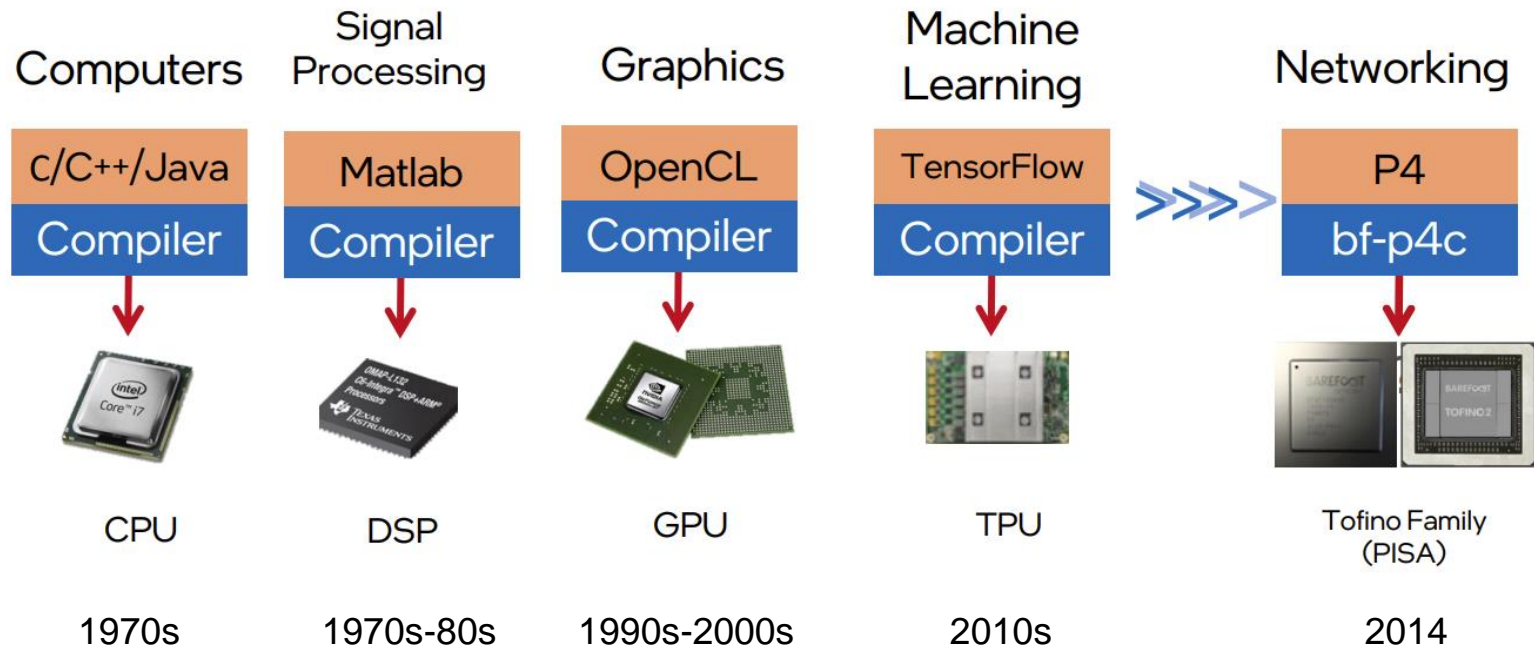
Can the Data Plane be Programmable?

- “Programmable switches are 10-100 times slower than non-programmable ones. They are more expensive and consume more power”¹

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

Can the Data Plane be Programmable?

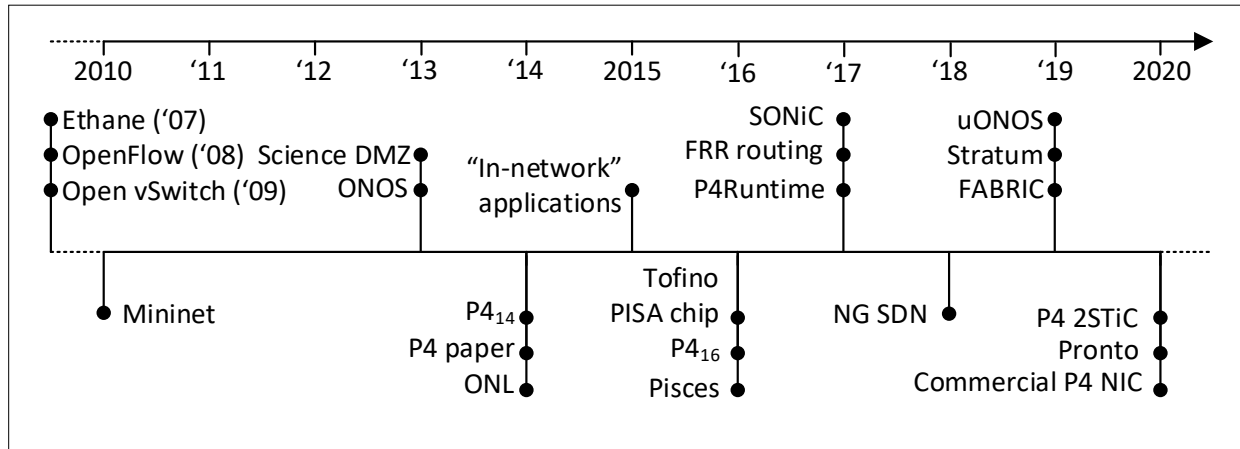
- Evolution of the computing industry



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

Can the Data Plane be Programmable?

- “Programmable switches are 10-100 times slower than non-programmable ones. They are more expensive and consume more power”
- The above assumption was challenged by a group of researchers at Stanford and Texas Instruments that led to “Barefoot Networks” in 2013



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

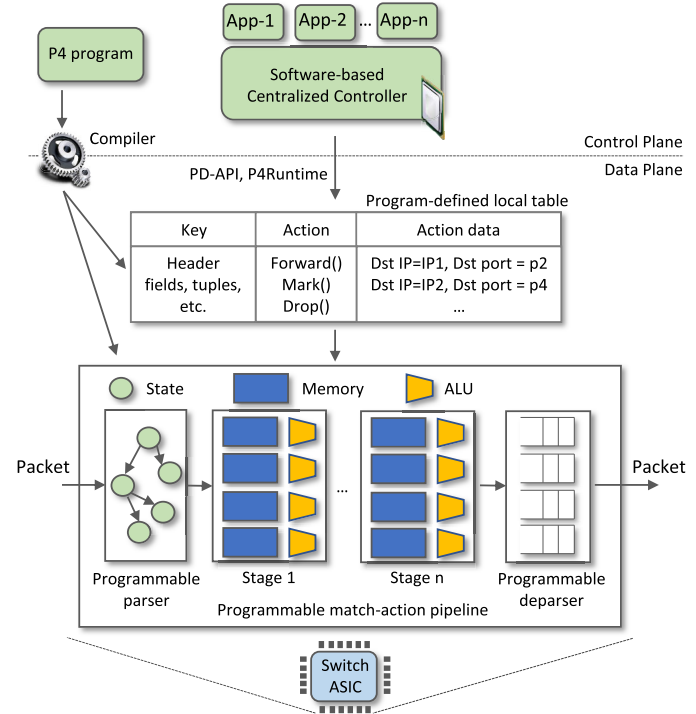


64x100GE
Barefoot Tofino

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

P4 Programmable Switches

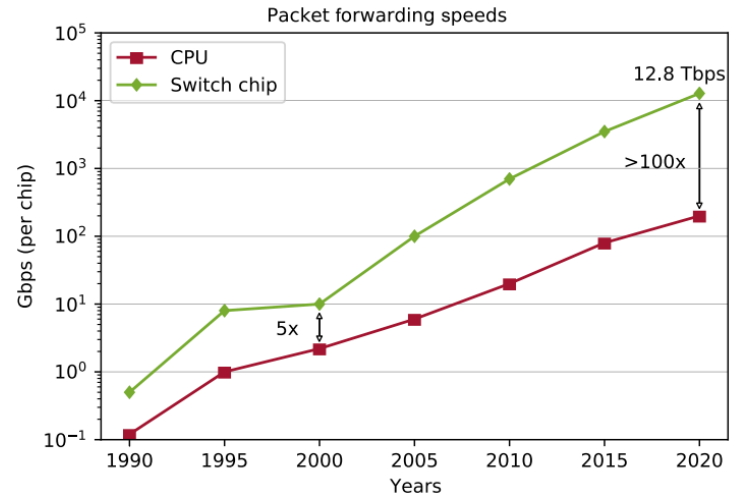
- P4¹ programmable switches permit a programmer to program the data plane
 - Define and parse new protocols
 - Customize packet processing functions
 - Measure events occurring in the data plane with high precision
 - Offload applications to the data plane



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

P4 Programmable Switches

- P4¹ 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

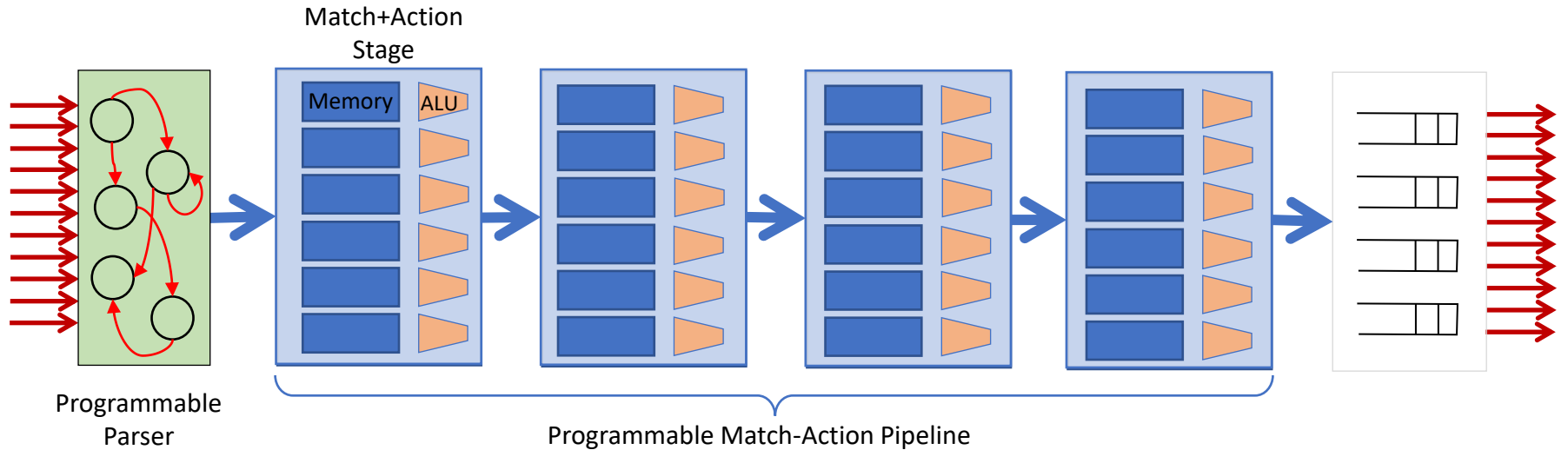


Reproduced from N. McKeown. Creating an End-to-End Programming Model for Packet Forwarding. Available: <https://www.youtube.com/watch?v=fiBuao6YZI0&t=4216s>

Generalized forwarding: Match + Action

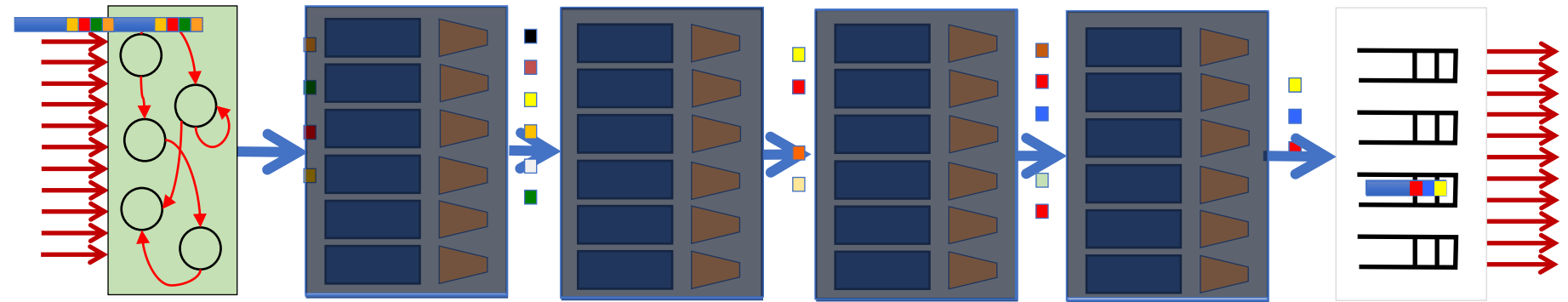
- Each switch contains table/s
 - Match bits in arriving packet (match phase)
 - Take action - Many header fields can determine action (action phase)
 - Drop
 - Copy
 - Modify
 - Log packet
 - Forward out a link (destination-based forwarding is just a particular case)

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();  
  }  
}
```

