Software Defined Networks

Pavol Helebrandt

STU
FIIT

Agenda

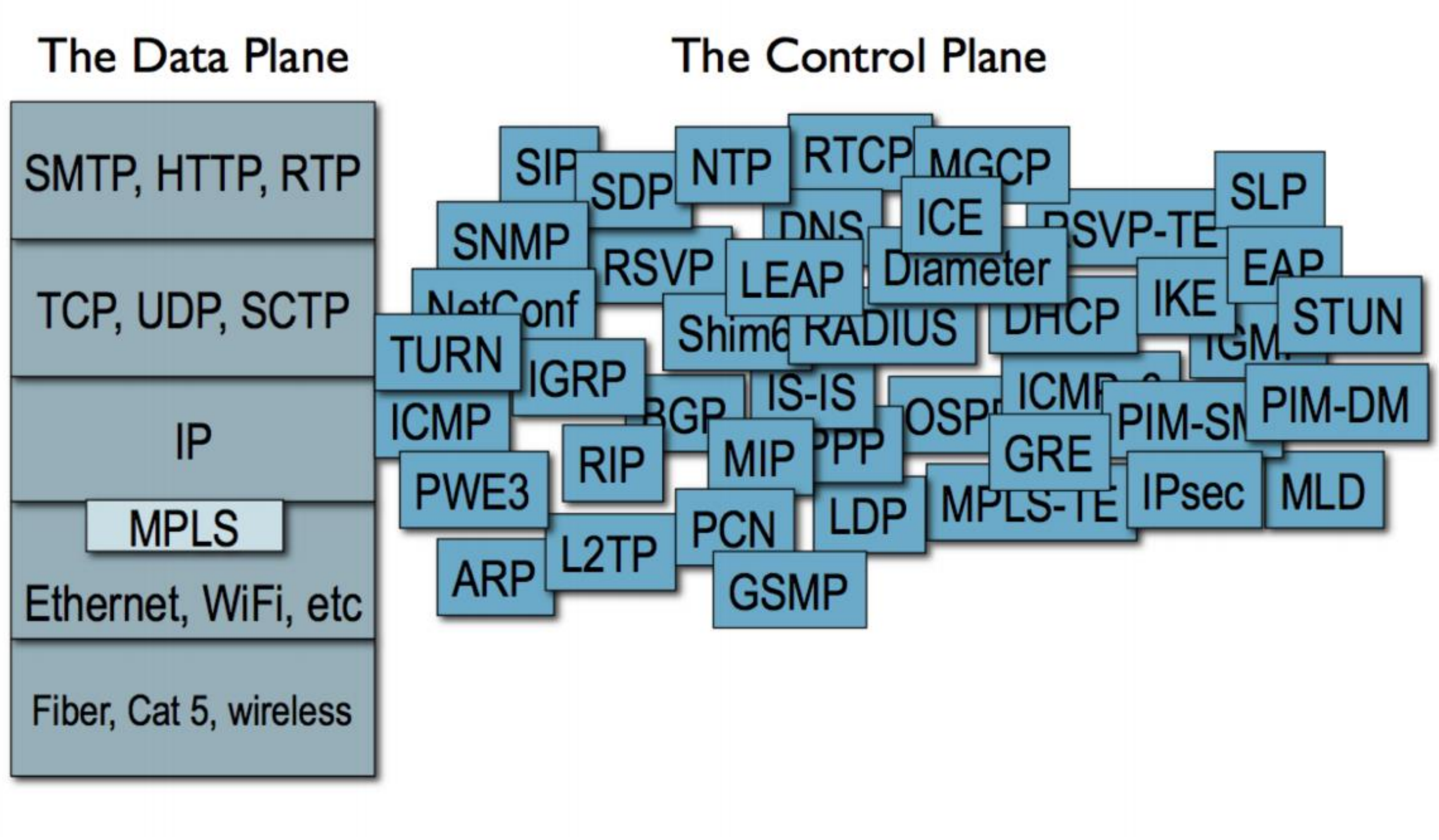
- » Brief history
- » SDN
- » OpenFlow
- » P4

Brief history

- Internet is a great success
  - Provides applications and services
  - Reliable (or unreliable) transport
  - Global best effort network
  - Local best effort networks
  - Physical transfer of bits
- Change and innovation only at the edge

Brief history

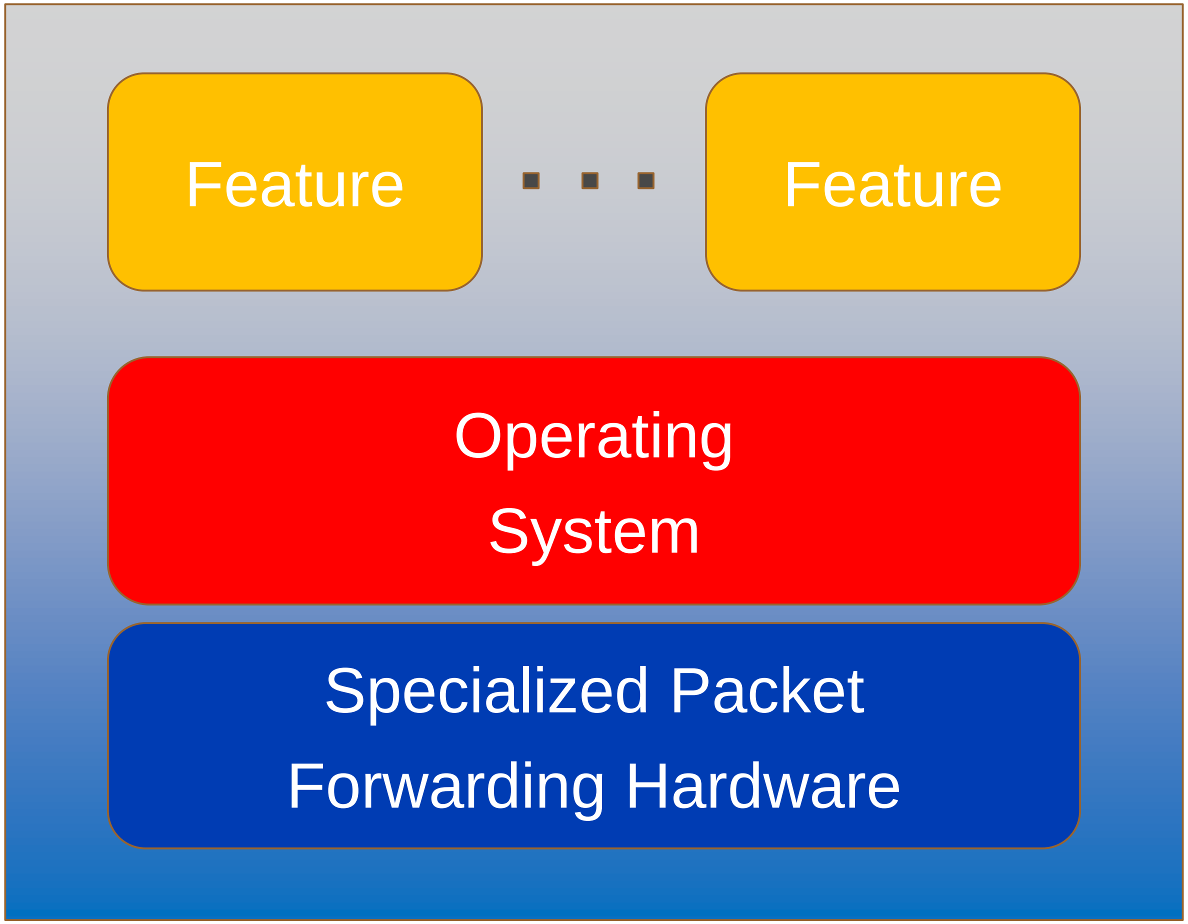
- Other IT fields (OS, DB, etc.):
  - Teach basic principles
  - Are easily managed
  - Continue to evolve
- Traditional networking:
  - Teach a lot of protocols
  - Difficult to manage
  - Evolves relatively slowly



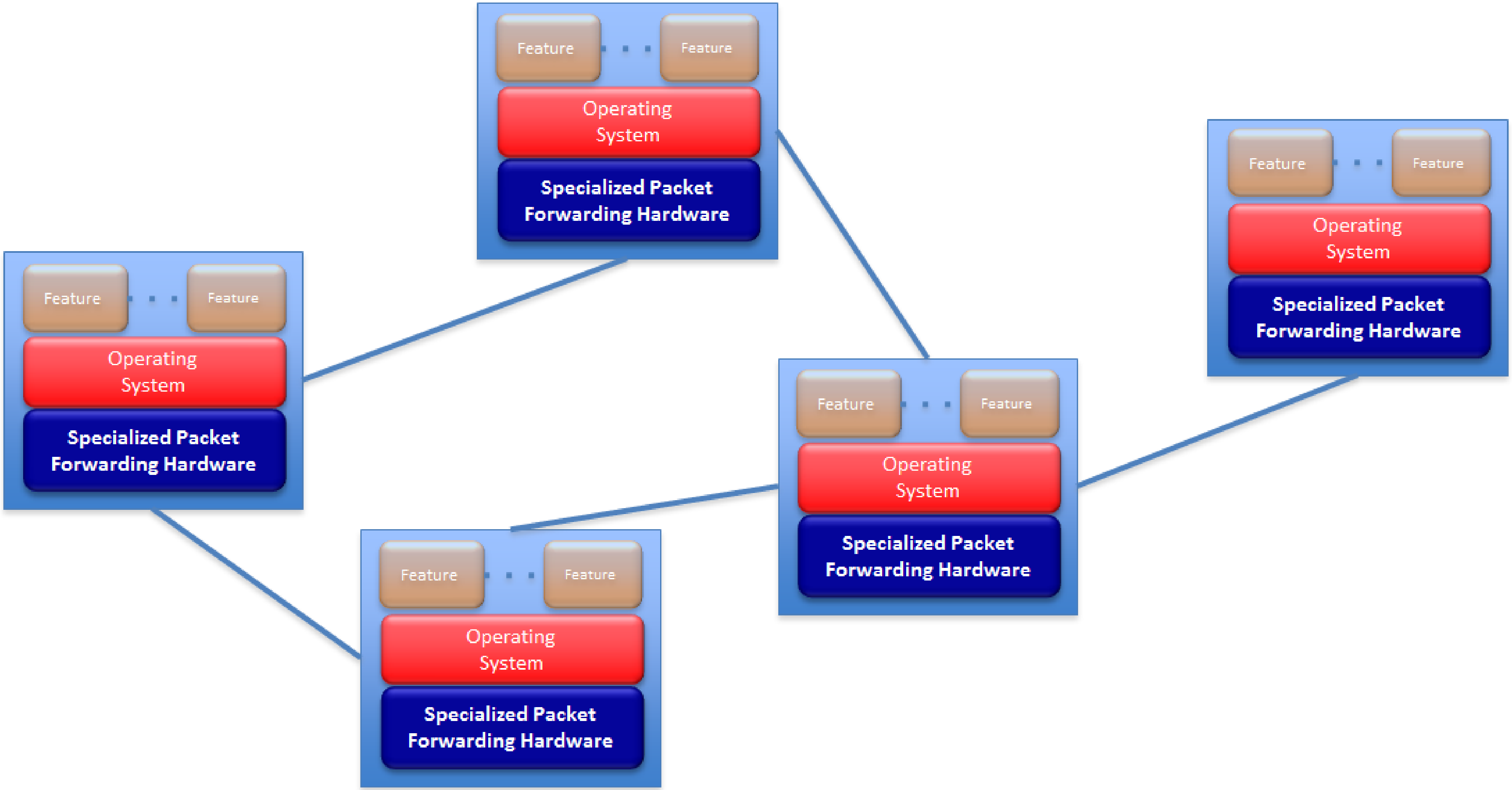
- Mark Handley. Re-thinking the control architecture of the internet.
- Keynote talk. REARCH. December 2009.

```
•interface Ethernet0/0.17
•encapsulation dot1Q 17
•ip address 125.1.17.7 255.255.255.0
•ip pim bsr-border
•ip pim sparse-mode
•!
•router ospf 1
•router-id 120.1.7.7
•redistribute bgp 700 subnets
•!
•router bgp 700
•neighbor 125.1.17.1 remote-as 100
•!
•address-family ipv4
•redistribute ospf 1 match internal external 1 external 2
•neighbor 125.1.17.1 activate
•!
•address-family ipv4 multicast
•network 125.1.79.0 mask 255.255.255.0
•redistribute ospf 1 match internal external 1 external 2
```

Traditional networking



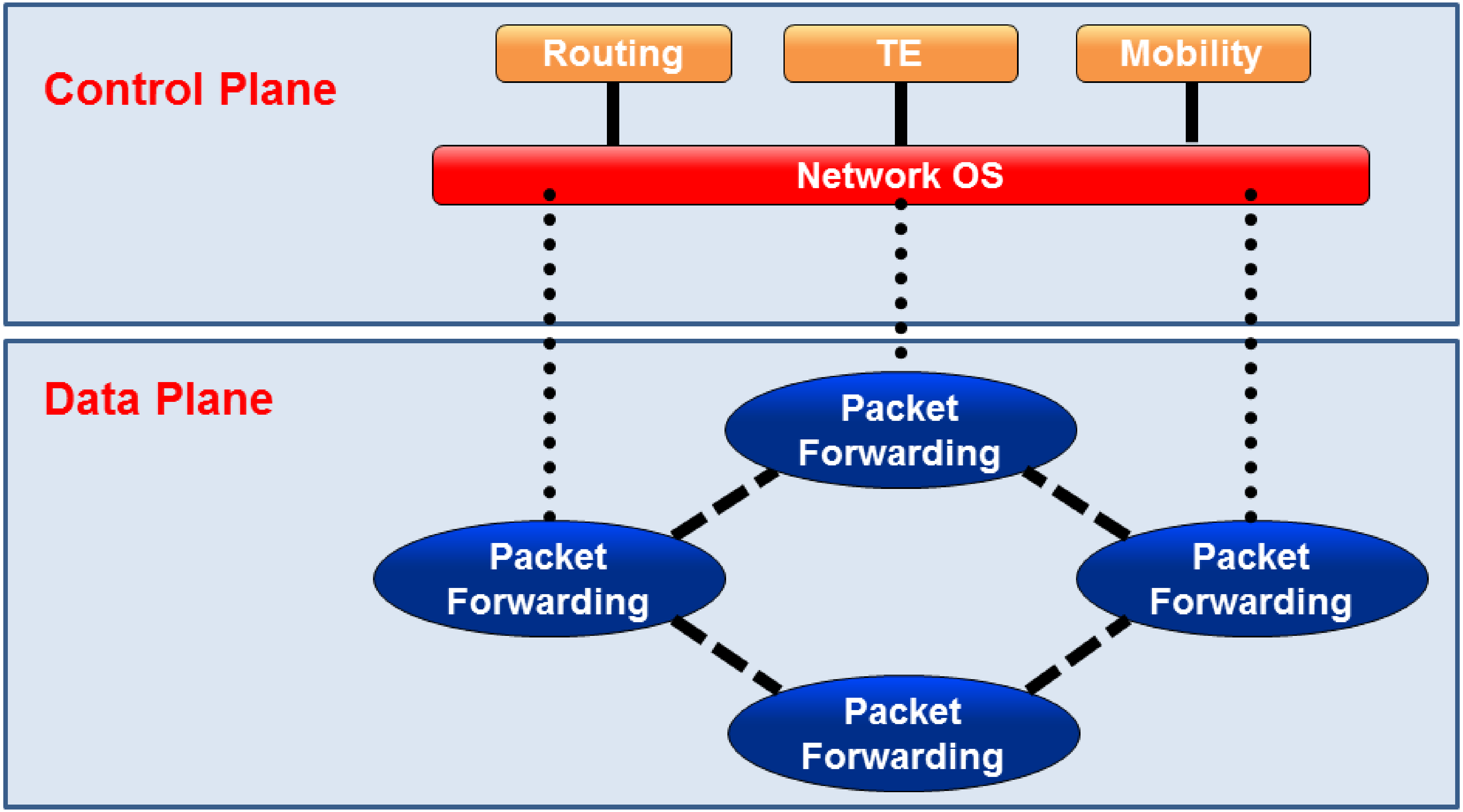
Traditional networking


Feature
Feature
Operating System
Specialized Packet Forwarding Hardware
Feature
Feature
Operating System
Specialized Packet Forwarding Hardware
Feature
Feature
Operating System
Specialized Packet Forwarding Hardware
Feature
Feature
Operating System
Specialized Packet Forwarding Hardware
Feature
Feature
Operating System
Specialized Packet Forwarding Hardware

SDN

- SDN defining features:
  - Separate control plane from data plane
  - (Logically) Centralized control of data plane devices

SDN


Control Plane
Routing
TE
Mobility
Network OS
Data Plane
Packet Forwarding
Packet Forwarding
Packet Forwarding
Packet Forwarding

SDN

- Separate control plane
  - Abstract view of network
  - Simplify interface for controlling program
  - Make networking innovation easier

Data (forwarding) plane

- Packet processing and forwarding
- Fast and “simple” hardware implementation
- Software implementation for VMs directly in server
- Standard interface (API) to data plane
- Simple set of operations over packet

Control plane

- High level processing and routing decisions
- VERY complex
- Slow(er) and “intelligent” software implementation
- Provide unified network abstraction for applications
- Simplified network management
- Greater network flexibility

Management plane

- Configuration and management
- Network tuning

- Traffic engineering, device provisioning, …

OpenFlow

- Standardized and maintained by Open Networking Foundation
- Developed for experiments and prototyping on production network - slicing
- Current OF switch specification v1.5.1 (important v1.0 and v1.3)

OpenFlow

- Enables the network to react to topology, application or user changes in real-time
- Early commercial adoption in data centres
- (Sometimes) Misunderstood to be the only SDN solution

OpenFlow

- » OpenFlow Channel
- » Message types

- » Matches
- » Actions

OpenFlow

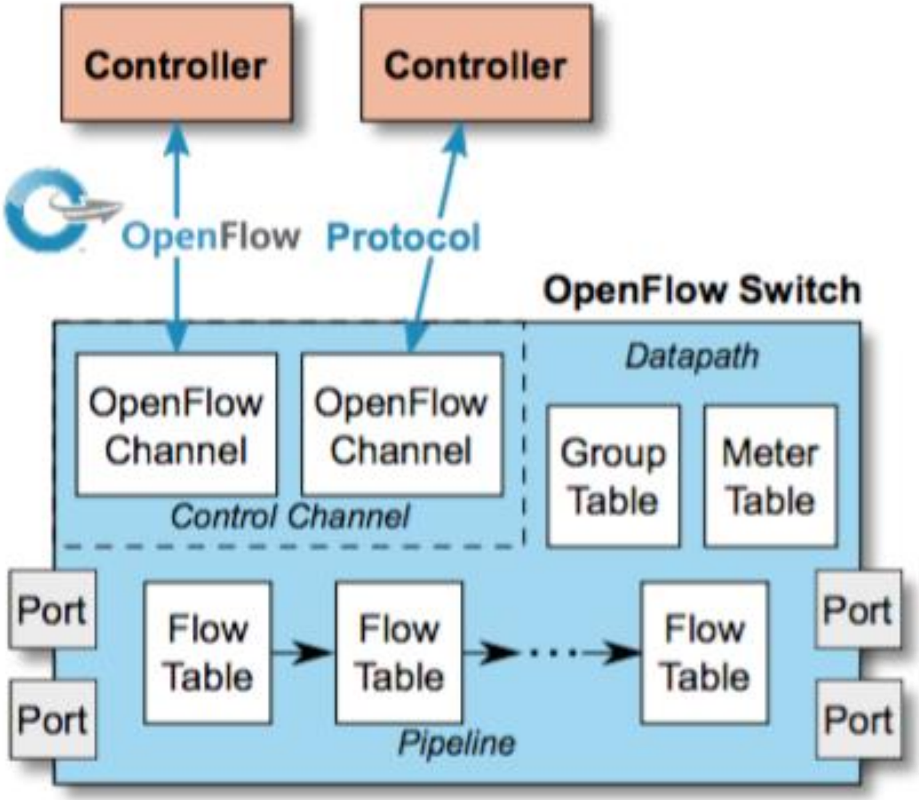


Figure 1: Main components of an OpenFlow switch.

OpenFlow

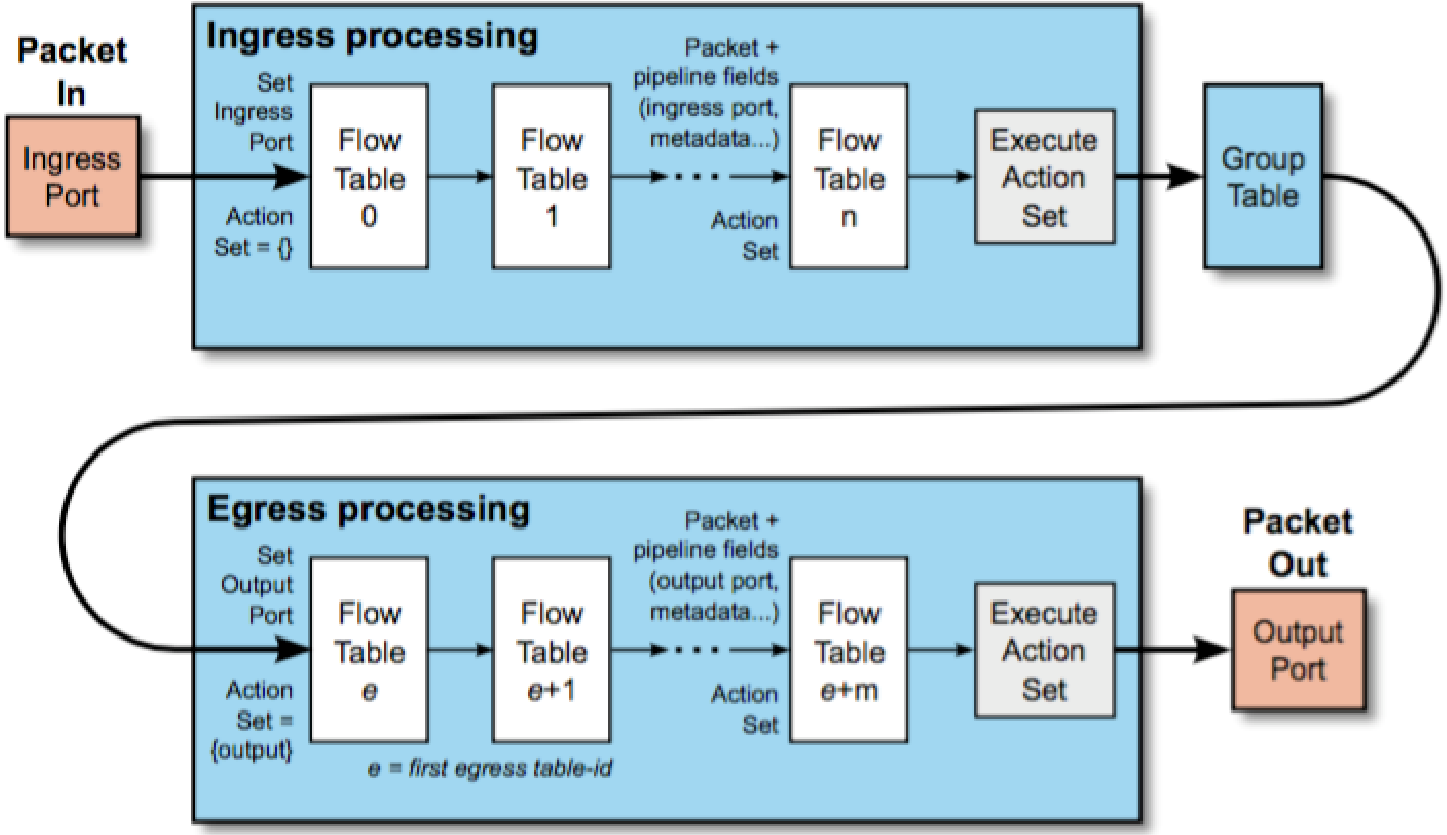


Figure 2: Packet flow through the processing pipeline.

OpenFlow

```
sh ovs-ofctl dump-flows s1
 cookie=0x8e67bd412a95a3fb, duration=5662.610s, table=0,
n_packets=61, n_bytes=20971, idle_age=5083,
priority=4,in_port=2,dl_vlan=1 actions=strip_vlan,NORMAL

 cookie=0x8e67bd412a95a3fb, duration=5679.407s, table=0, n_packets=3,
n_bytes=258, idle_age=5662, priority=2,in_port=2 actions=drop

 cookie=0x8e67bd412a95a3fb, duration=5679.641s, table=0,
n_packets=7779256, n_bytes=30272629666, idle_age=1, priority=0
actions=flood
```

OpenFlow match

```
sh ovs-ofctl dump-flows s1
```

```
 cookie=0x8e67bd412a95a3fb, duration=5662.610s, table=0,
n_packets=61, n_bytes=20971, idle_age=5083,
priority=4,in_port=2,dl_vlan=1 actions=strip_vlan,NORMAL
```

```
 cookie=0x8e67bd412a95a3fb, duration=5679.407s, table=0, n_packets=3,
n_bytes=258, idle_age=5662, priority=2,in_port=2 actions=drop
```

```
 cookie=0x8e67bd412a95a3fb, duration=5679.641s, table=0,
n_packets=7779256, n_bytes=30272629666, idle_age=1, priority=0
actions=flood
```

OpenFlow action

```
sh ovs-ofctl dump-flows s1

 cookie=0x8e67bd412a95a3fb, duration=5662.610s, table=0,
n_packets=61, n_bytes=20971, idle_age=5083,
priority=4,in_port=2,dl_vlan=1 actions=strip_vlan,NORMAL


 cookie=0x8e67bd412a95a3fb, duration=5679.407s, table=0, n_packets=3,
n_bytes=258, idle_age=5662, priority=2,in_port=2 actions=drop


 cookie=0x8e67bd412a95a3fb, duration=5679.641s, table=0,
n_packets=7779256, n_bytes=30272629666, idle_age=1, priority=0
actions=flood
```

OpenFlow

- Disadvantages
  - Protocol is too complex
  - Getting more complex
  - Incomplete implementations

P4

- P4: programming protocol-independent packet processors
- https://dl.acm.org/citation.cfm?id=2656890

P4

- Goals:
  - Reconfigurability
  - Protocol independence
  - Target independence

P4

- P4 program configures forwarding behavior
- Express serial dependencies
- P4 compiler translates into a target-specific representation
- OF can still be used to install and query rules once forwarding model is defined

P4

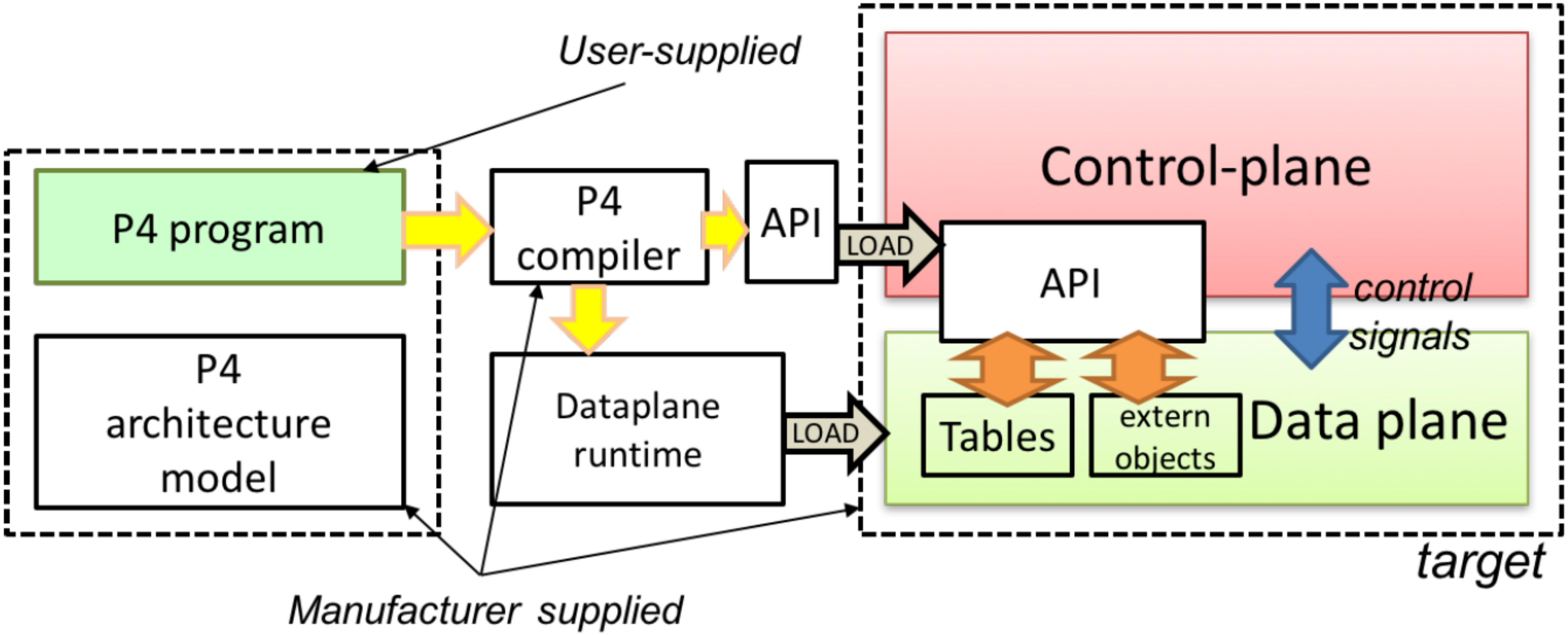


Figure 2. Programming a target with P4.

P4

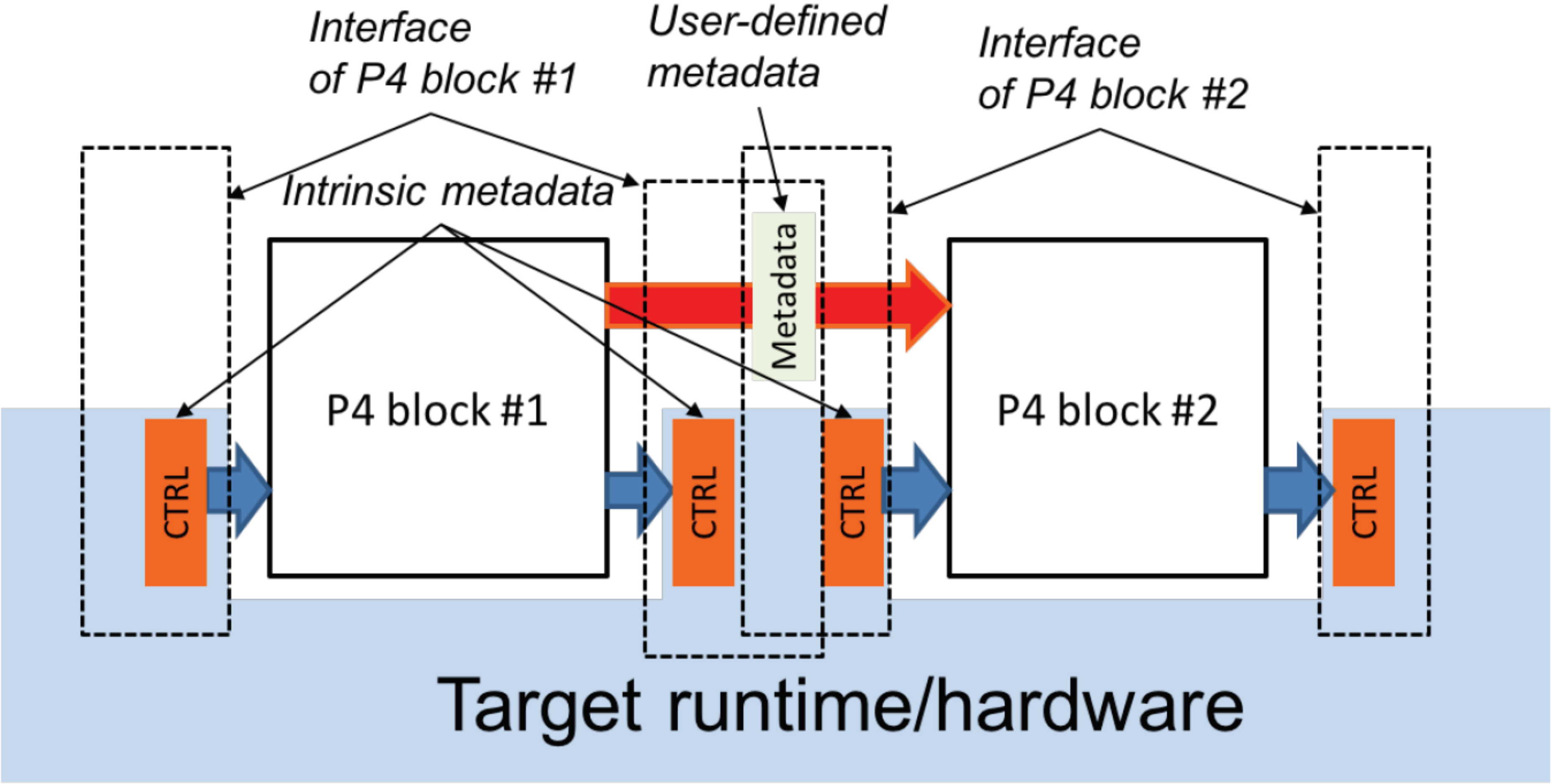


Figure 4. P4 program interfaces.

https://www.opennetworking.org/

https://p4.org/

Nick McKeown, Jennifer Rexford Changhoon Kim and Laurent Vanbever lectures