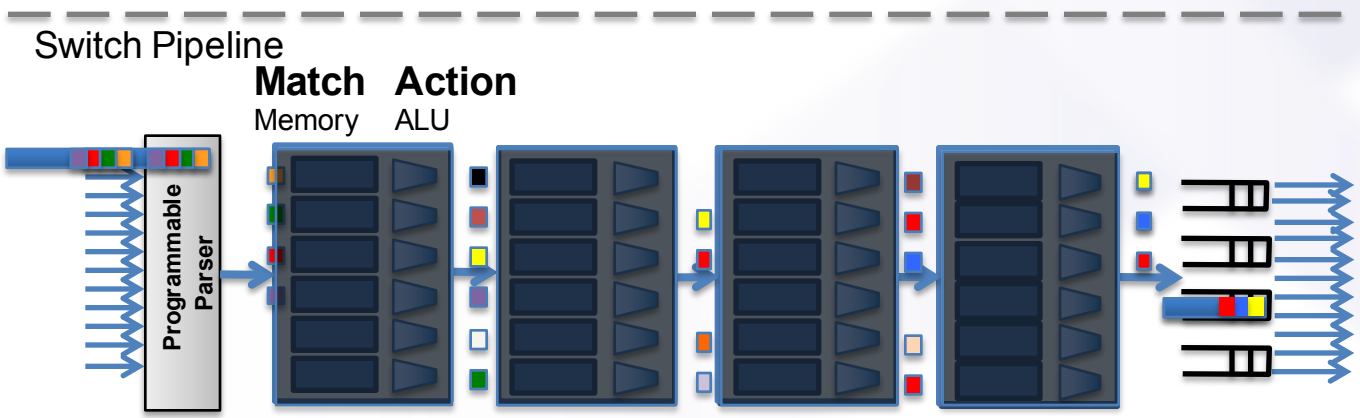
Programming The Network Data Plane in P4

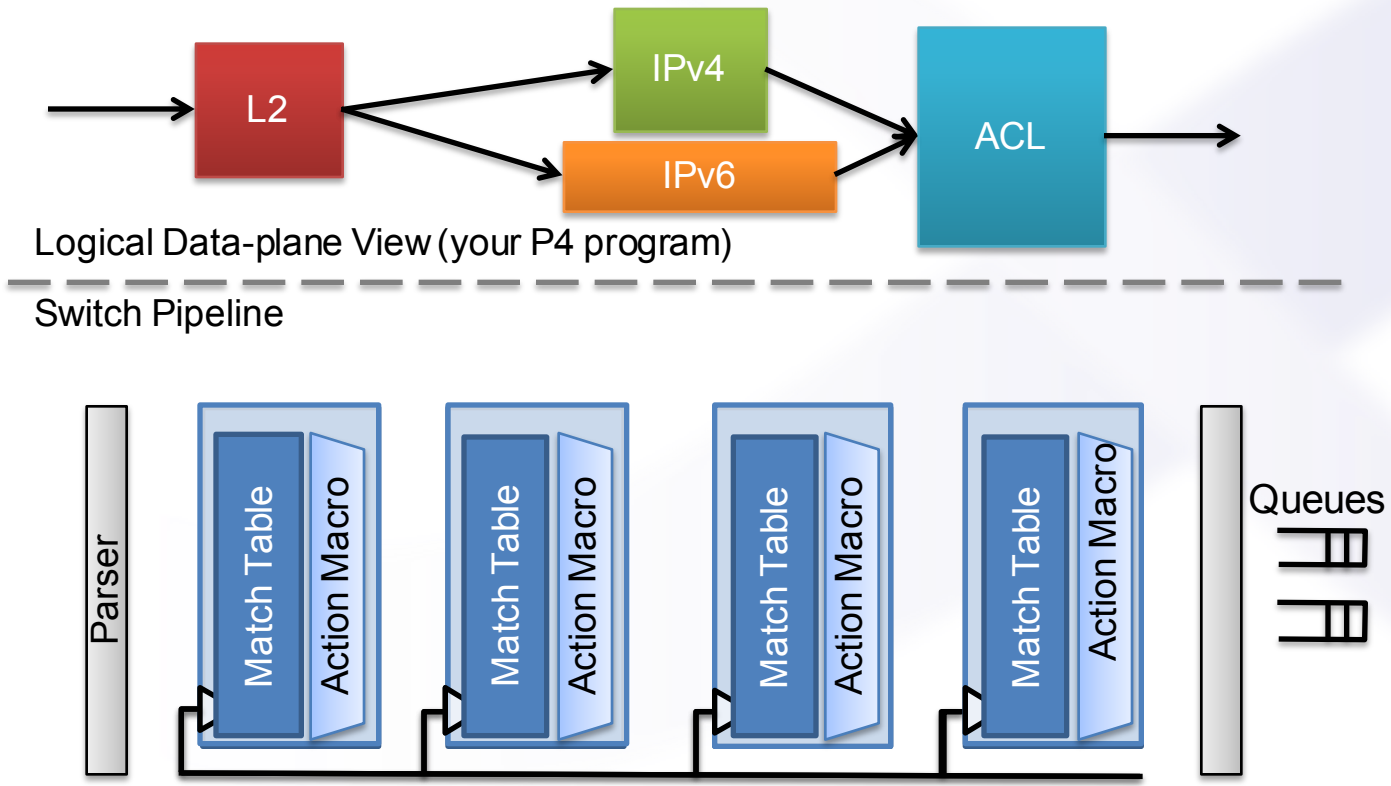
SIGCOMM'16
Tutorial

Barefoot Networks
Aug 2016

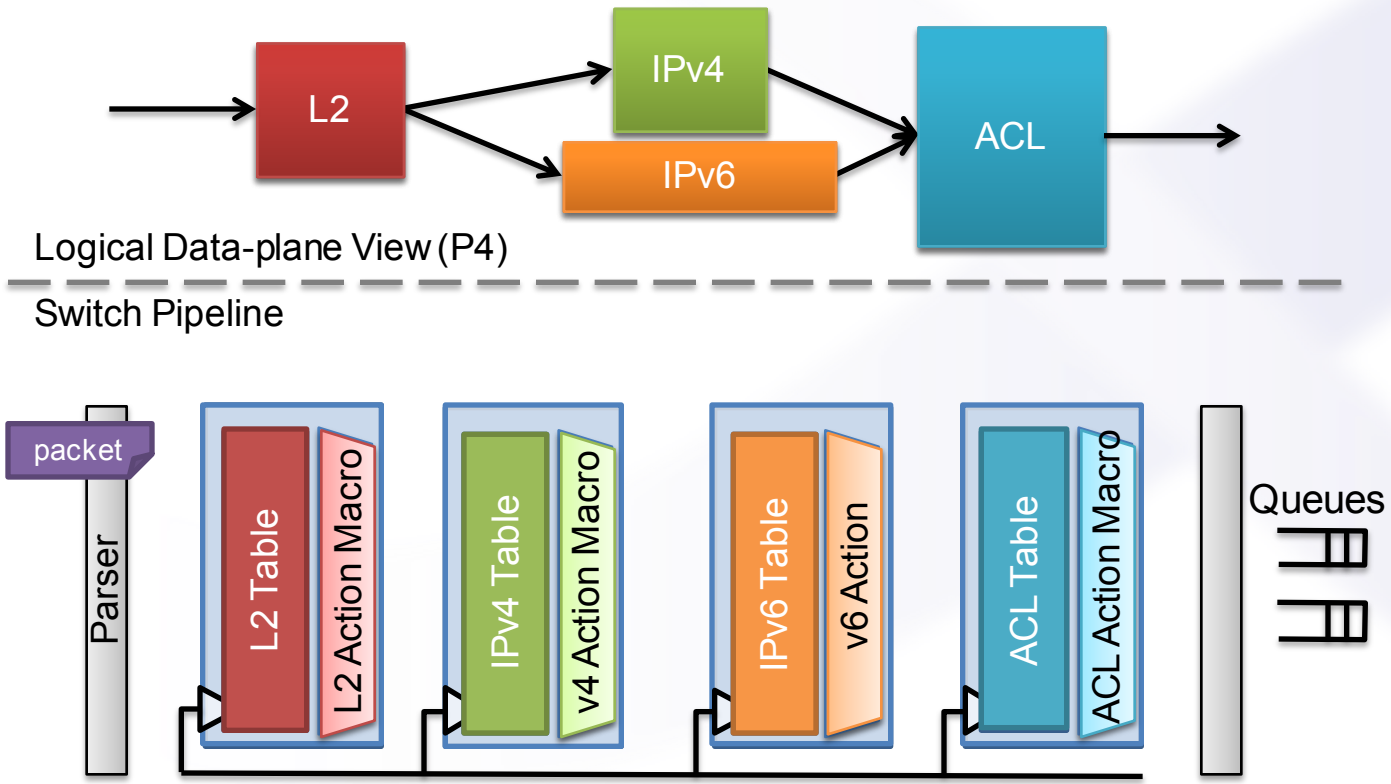
Protocol-Independent Switch Architecture (PISA)



Protocol-Independent Switch Architecture (PISA)



Mapping to Physical Resources



P4: Three Goals

Protocol independence

- Define a packet parser
- Define a set of typed match+action tables

Target independence

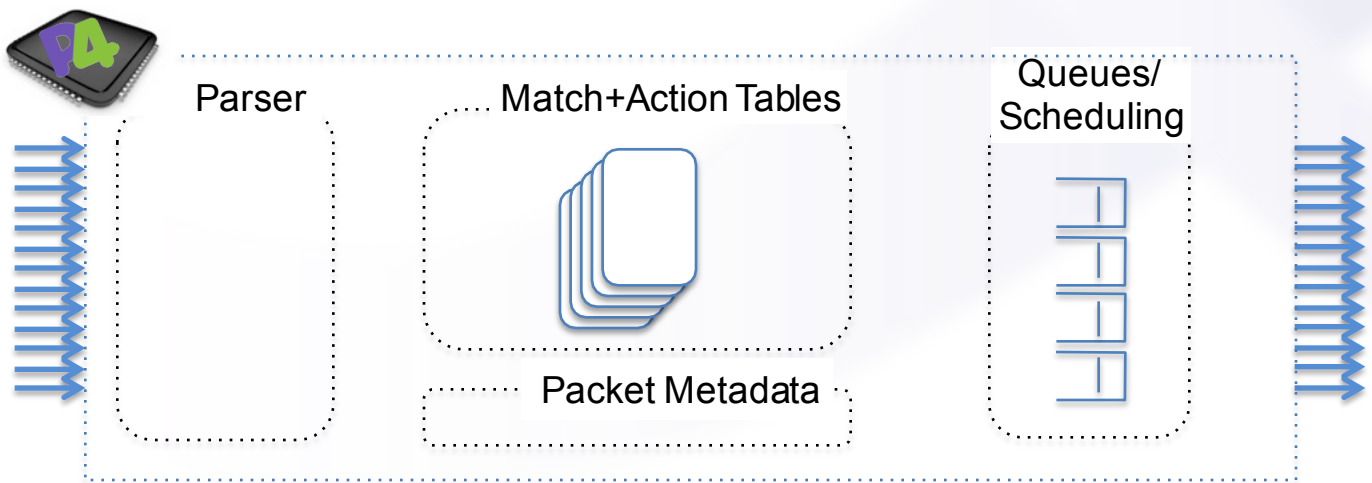
- Program without knowledge of packet-processing device details
- Let compilers configure the target device

In-field Re-configurability

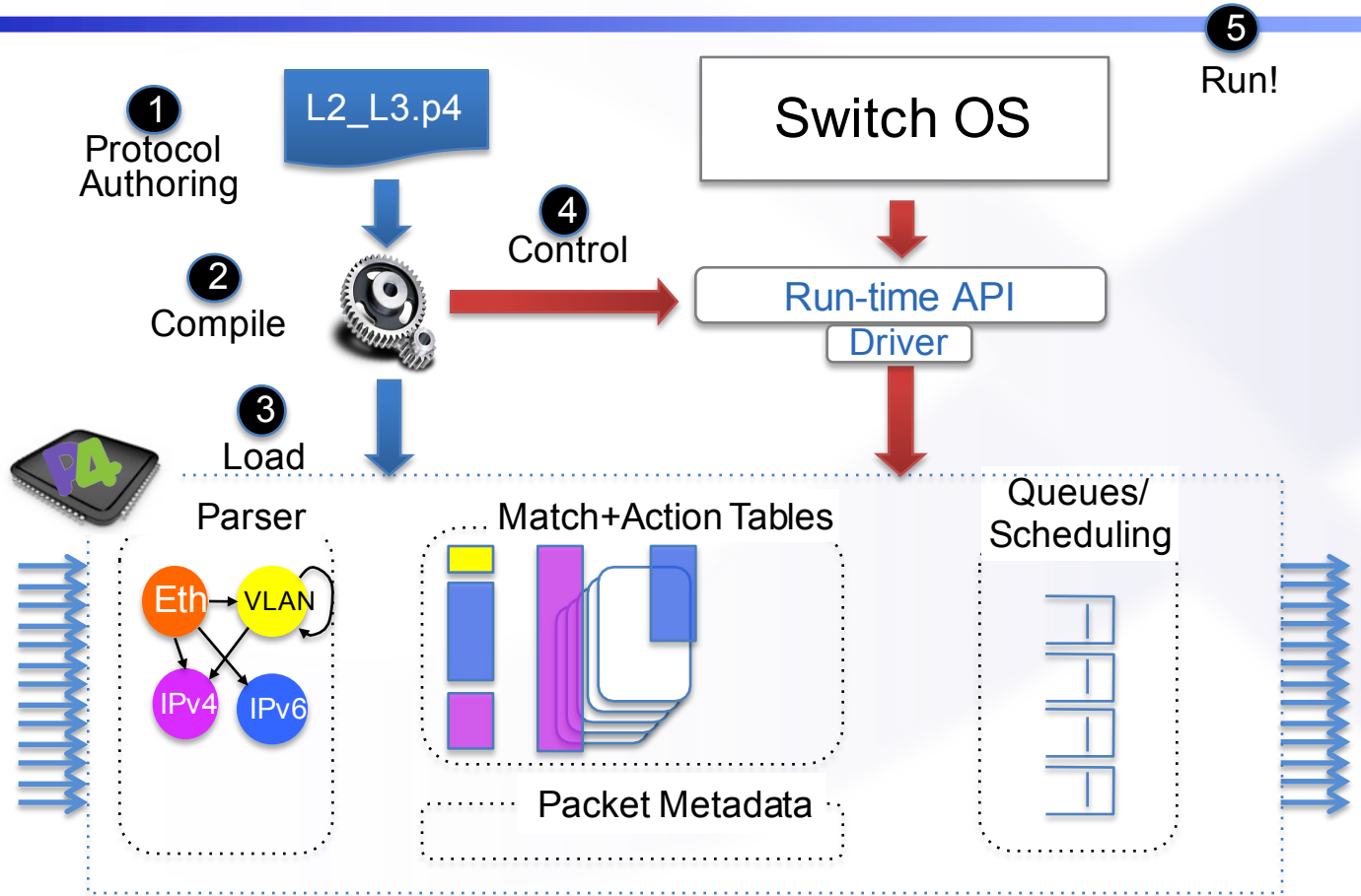
- Allow the users to change parsing and processing program in the field

P4-Based Workflow

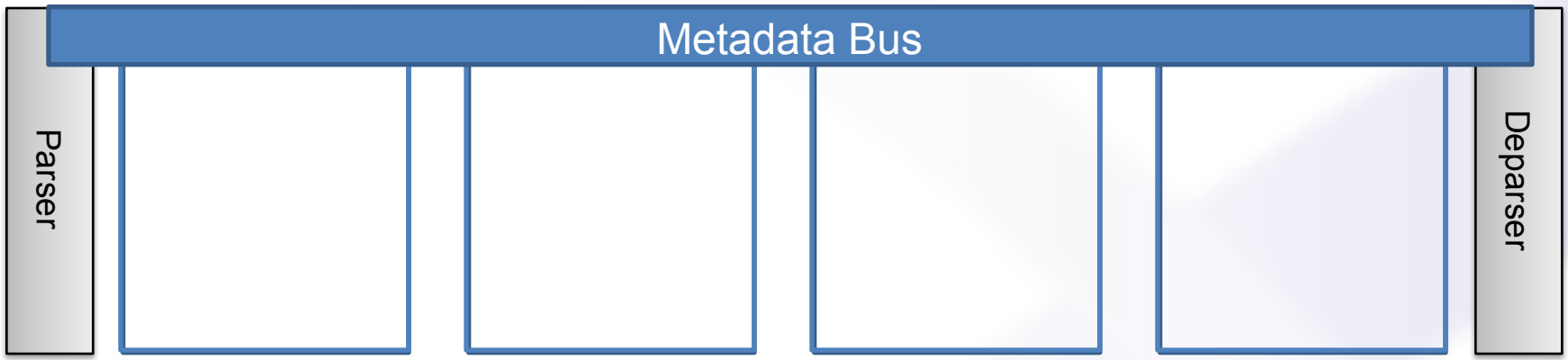
- **Device is not yet programmed**
  - Does not know about any packet formats or protocols



P4-Based Workflow



The anatomy of a basic pipeline



- **Parser**
  - Converts packet data into a metadata (Parsed Representation)
- **Match+Action Tables**
  - Operate on metadata
- **Deparser**
  - Converts metadata back into a serialized packet
- **Metadata Bus**
  - Carries the information within the pipeline

All are optional

P4 Program Sections

program.p4

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;
```

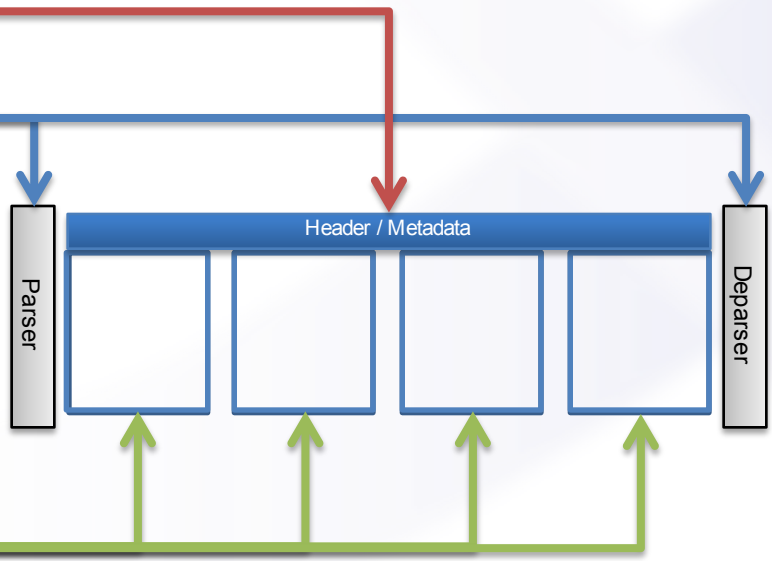
Parser Program

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

Table + Control Flow Program

```
table port_table { … }

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



P4 program defines what each table CAN do