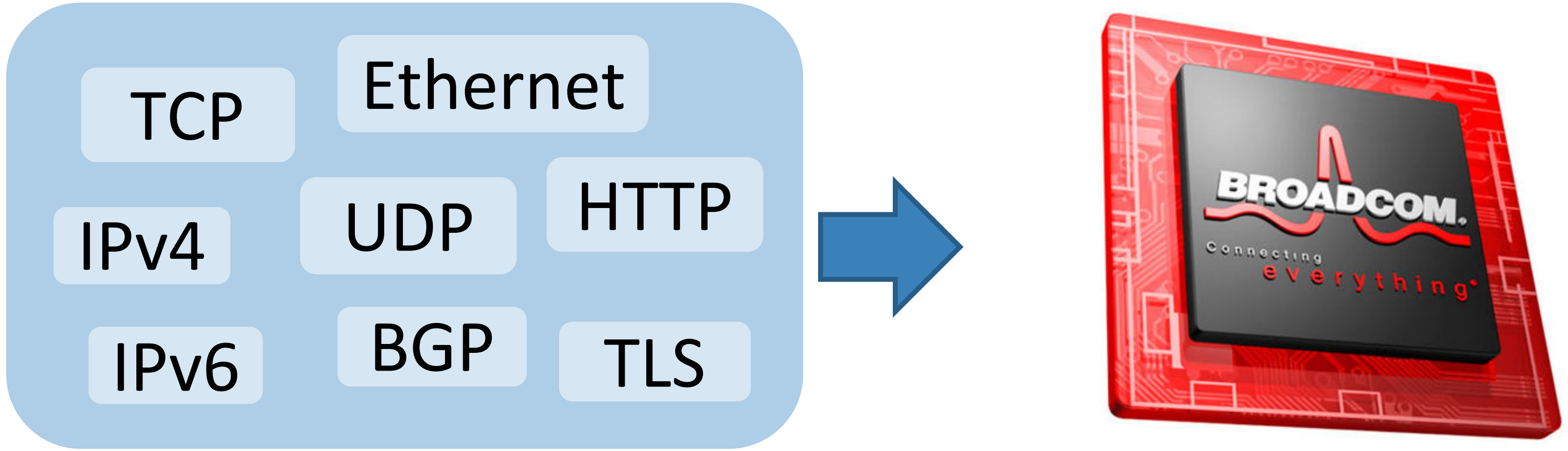
# PISCES: A Programmable, Protocol-Independent Software Switch

Muhammad Shahbaz, **Sean Choi**, Ben Pfaff, Changhoon Kim, Nick Feamster, Nick McKeown, and Jennifer Rexford

## Fixed Set of Protocols

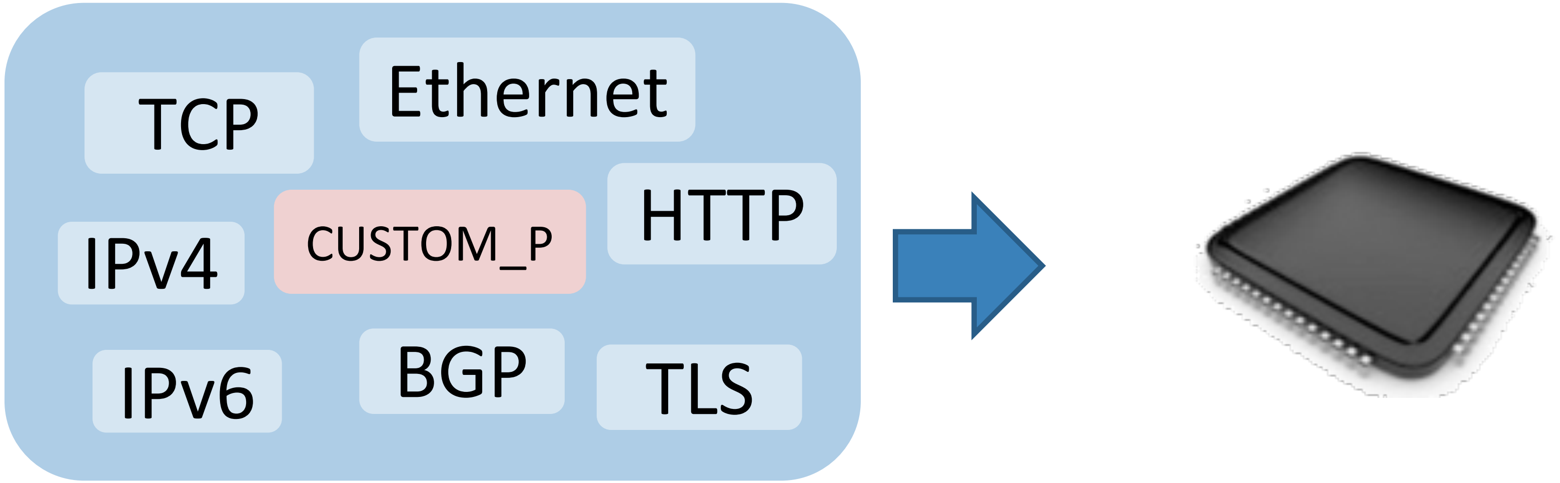
## Fixed-Function Switch Chip



- Ease of **Adding** new protocols
- Ease of **Removing** unused protocols
- Gain greater **Visibility** into the network
- **Perform** network functions at the switch

Custom Protocols

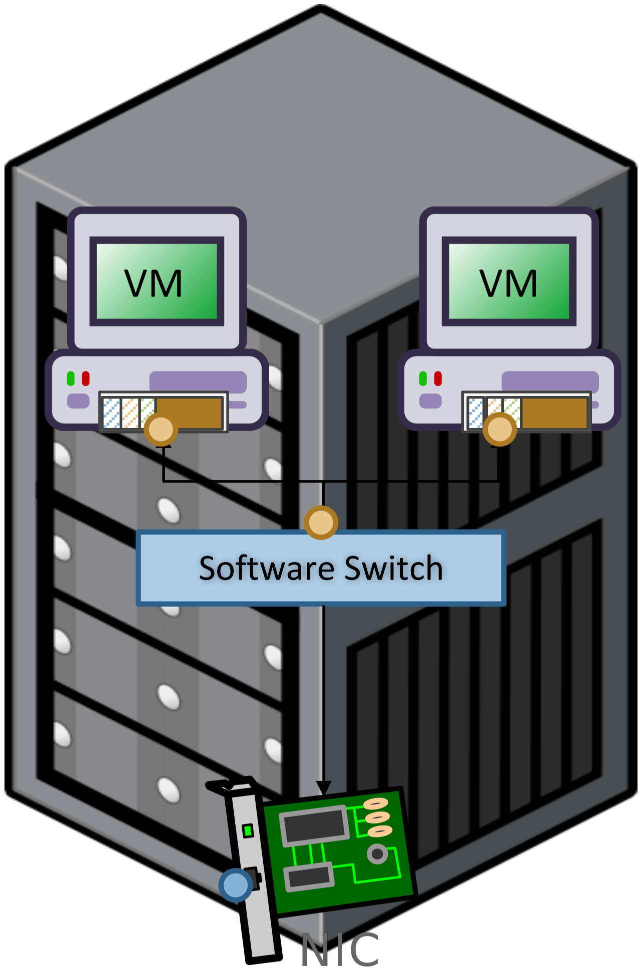
Programmable Switching Chip



intel®

Xpliant

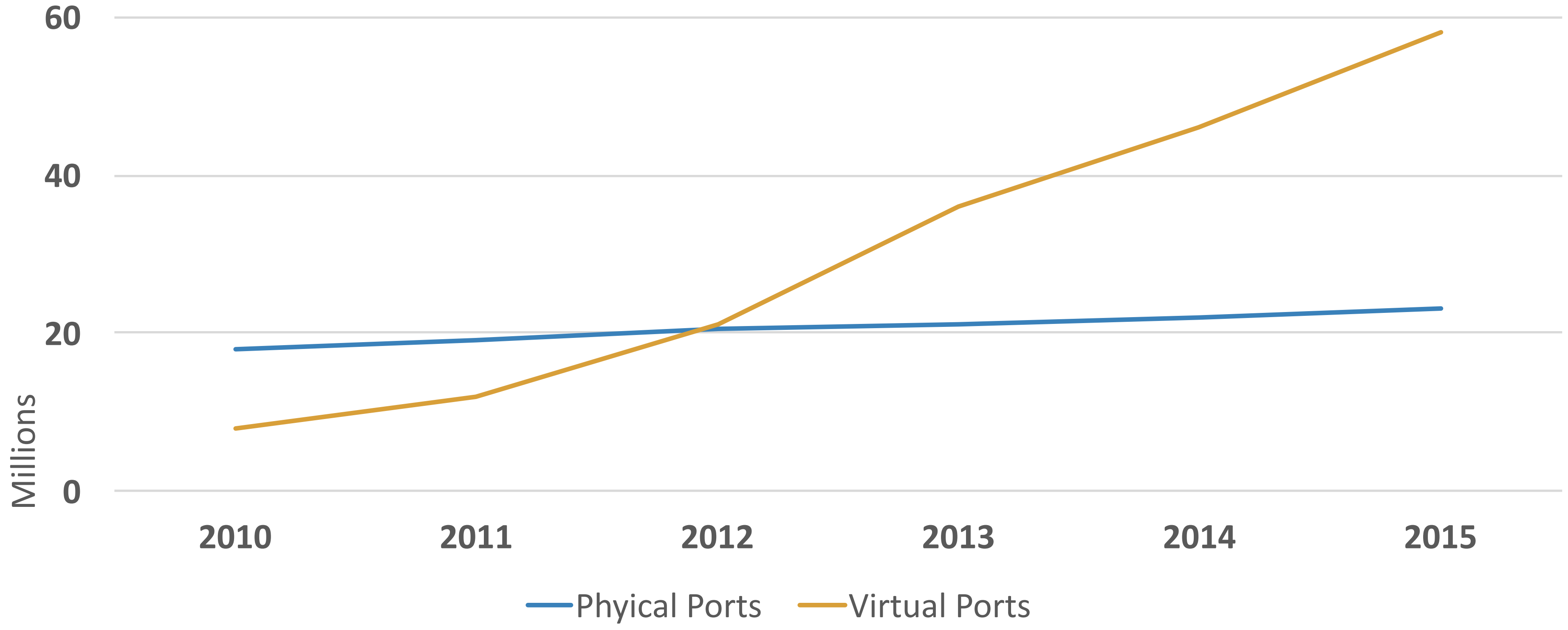
BAREFOOT NETWORKS



● Virtual Port

● Physical Port

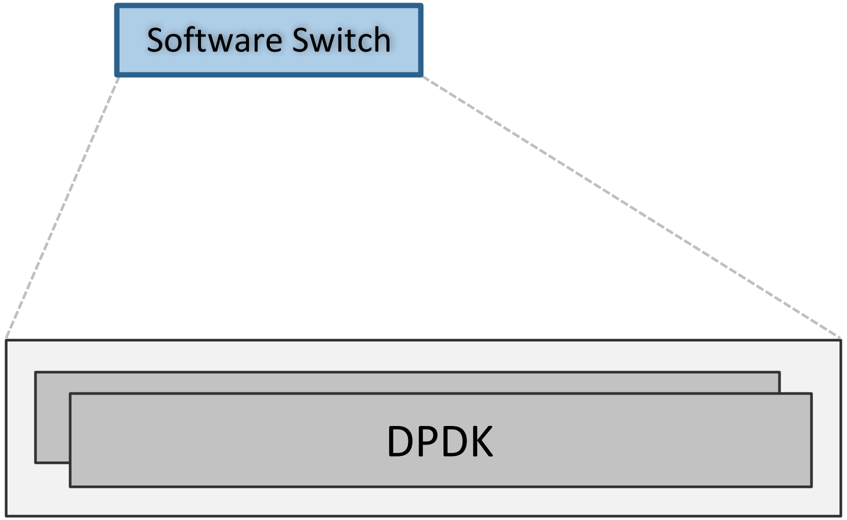
# Approx. Number of Physical Ports vs. Virtual Ports in Global Data Centers [1]

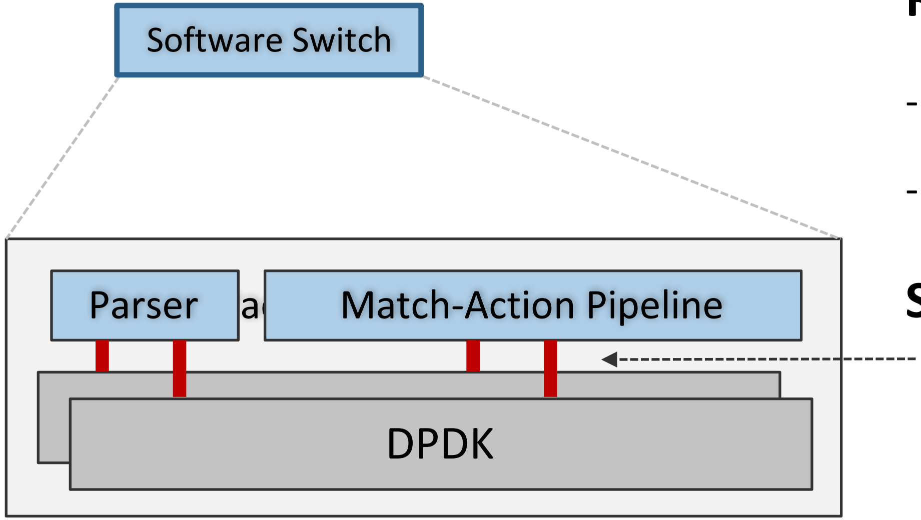


[1] Martin Casado, VMWorld 2013

It should be **EASY** to **program** software switches!

**Not really…**


Software Switch
DPDK

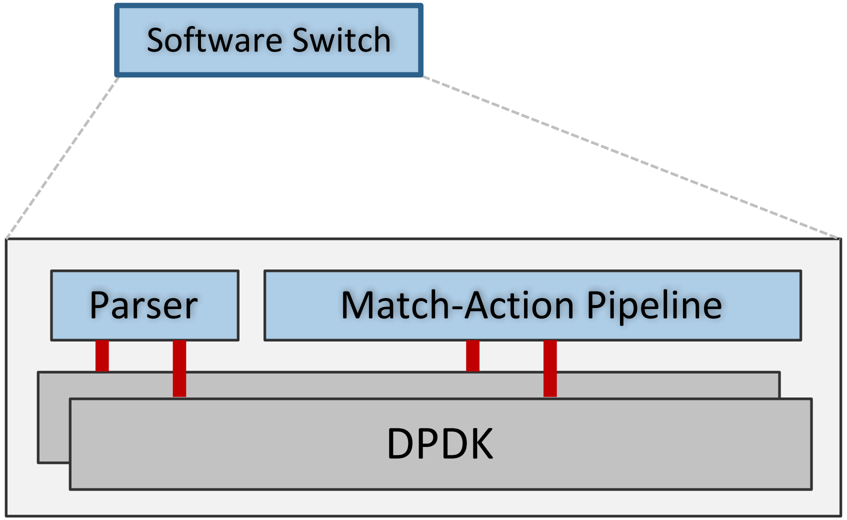


**Requires domain expertise in:**

- Network protocol design
- Kernel development

**Slow to release changes**

**Specialized APIs**



To add **TCP Flag** in Open vSwitch…

changed **20 files** and **370 lines of code**[1]

**Weeks** of development and Test

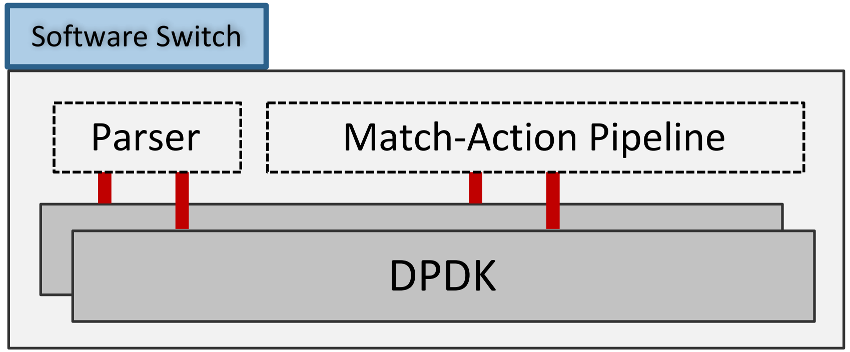
[1] https://github.com/openvswitch/ovs/commit/dc235f7fbcff

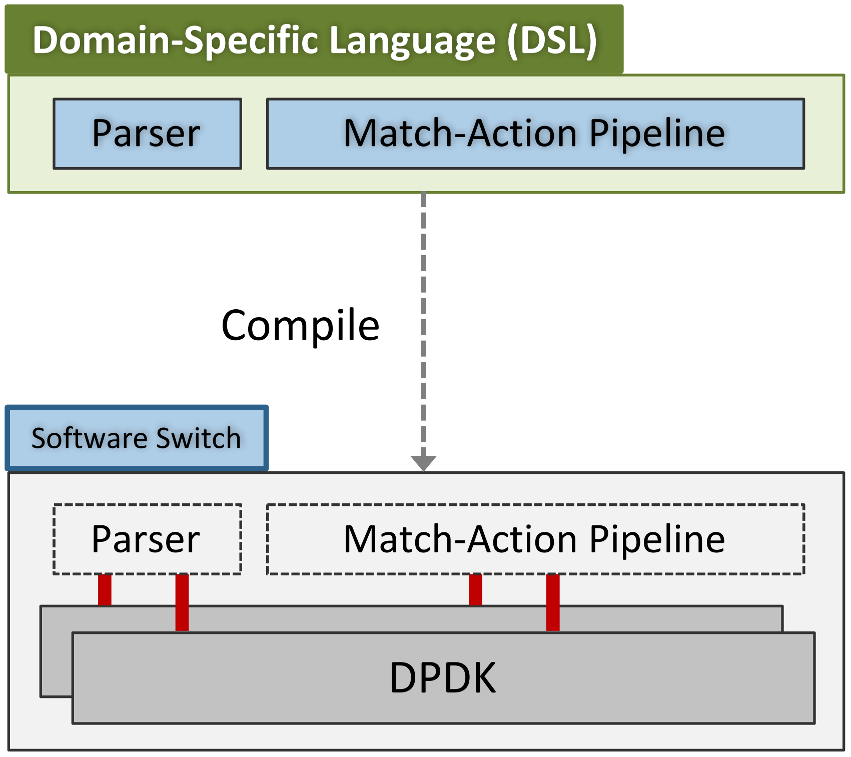
## We can do this in **4 lines** and **within minutes** with **PISCES!**

```
header_type tcp_t {
  fields {
    srcPort : 16;
    dstPort : 16;
    seqNo : 32;
    ackNo : 32;
    dataOffset : 4;
    res : 4;
    tcp_flags : 12;
    window : 16;
    checksum : 16;
    urgentPtr : 16;
  }
}
```

```
header_type flow_t {
  fields {
    ...
    tcp_flags_pad : 4;
    tcp_flags : 12;
    ...
  }
}
```

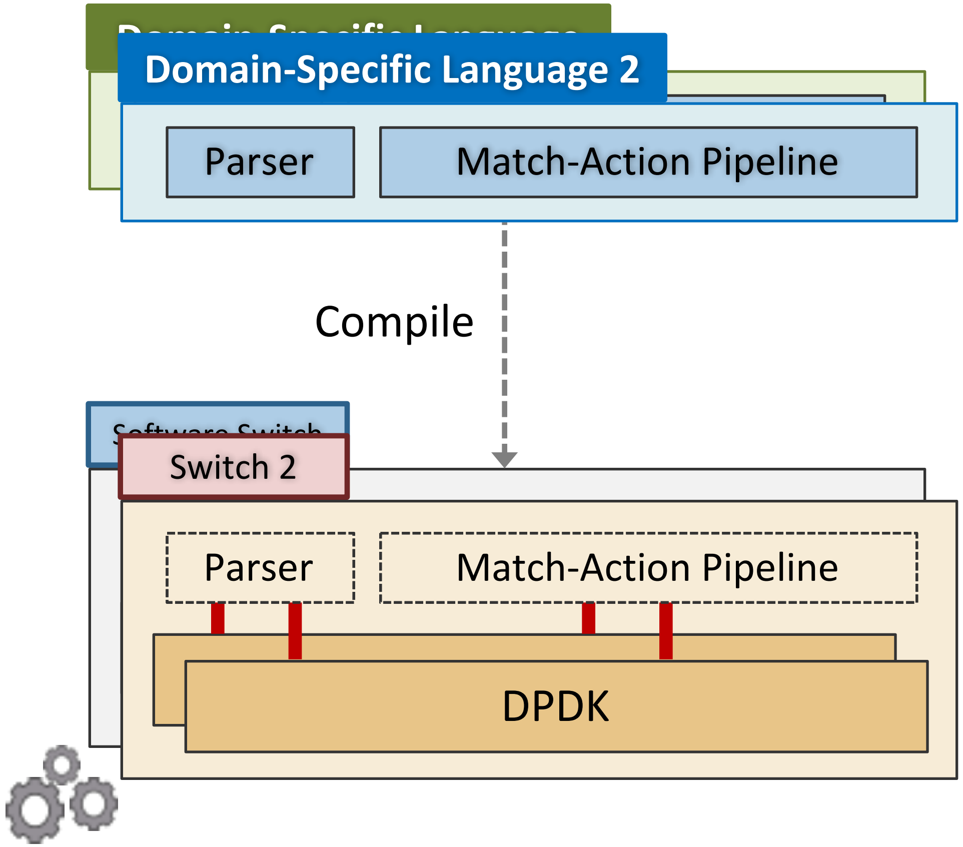
```
parser tcp {
  extract(tcp);
  set_metadata(flow.tcp_flags,
               tcp.tcp_flags);
  return ingress;
}
```


Software Switch
Parser
Match-Action Pipeline
DPDK



## TCP Header

```
header_type tcp_t {
  fields {
    srcPort : 16;
    dstPort : 16;
    seqNo : 32;
    ackNo : 32;
    dataOffset : 4;
    res : 4;
    window : 16;
    checksum : 16;
    urgentPtr : 16;
  }
}
parser tcp {
  extract(tcp);
  return ingress;
}
...
```

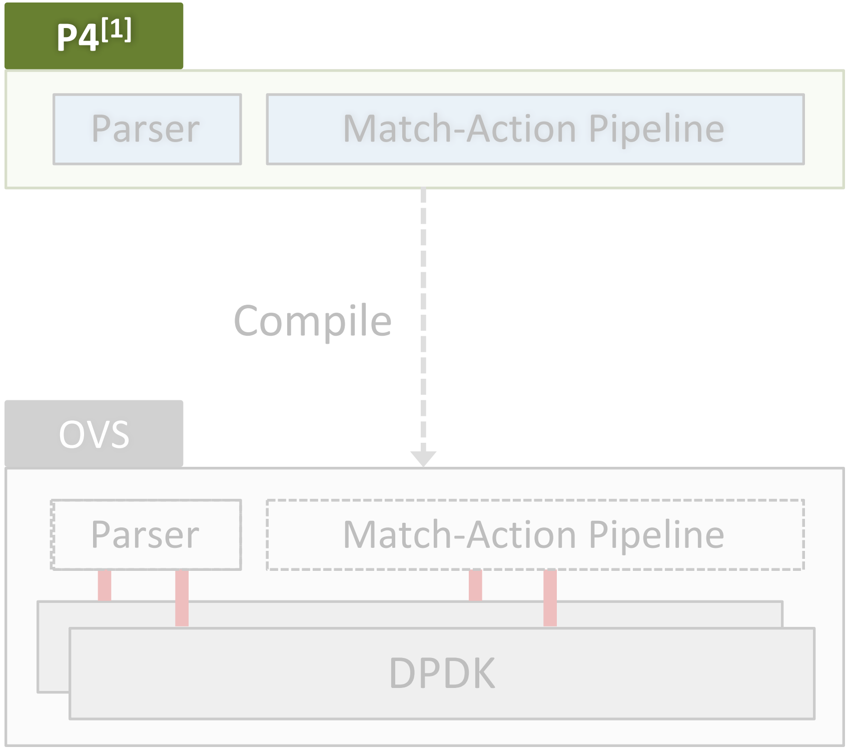


**PISCES** is a **software switch** that takes

- a **Domain-Specific Language** input

  and outputs a customized

- a **Software Switch Target**

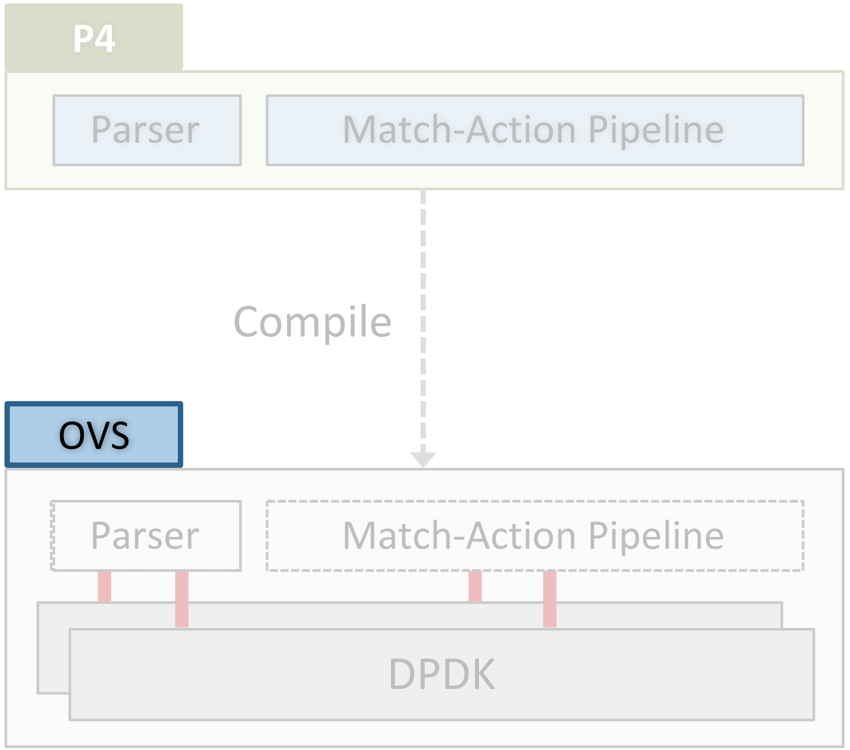


P4 is an **open-source language.**[1]

Easily defines

- **Packet headers and fields**
- **Parser**
- **Actions**
- **Match-Action Tables**

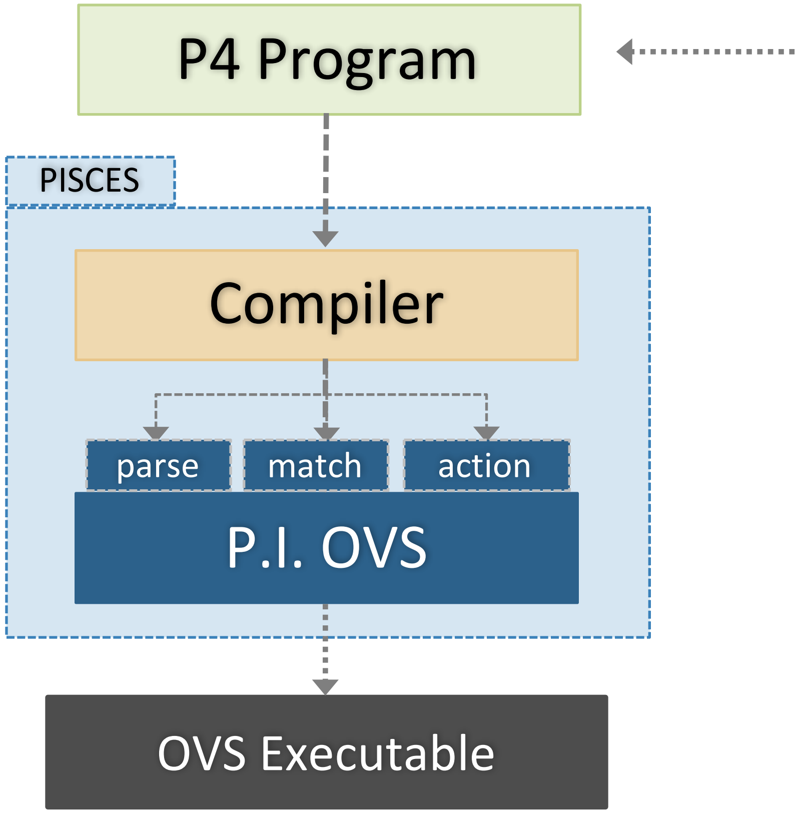
[1] http://p4.org

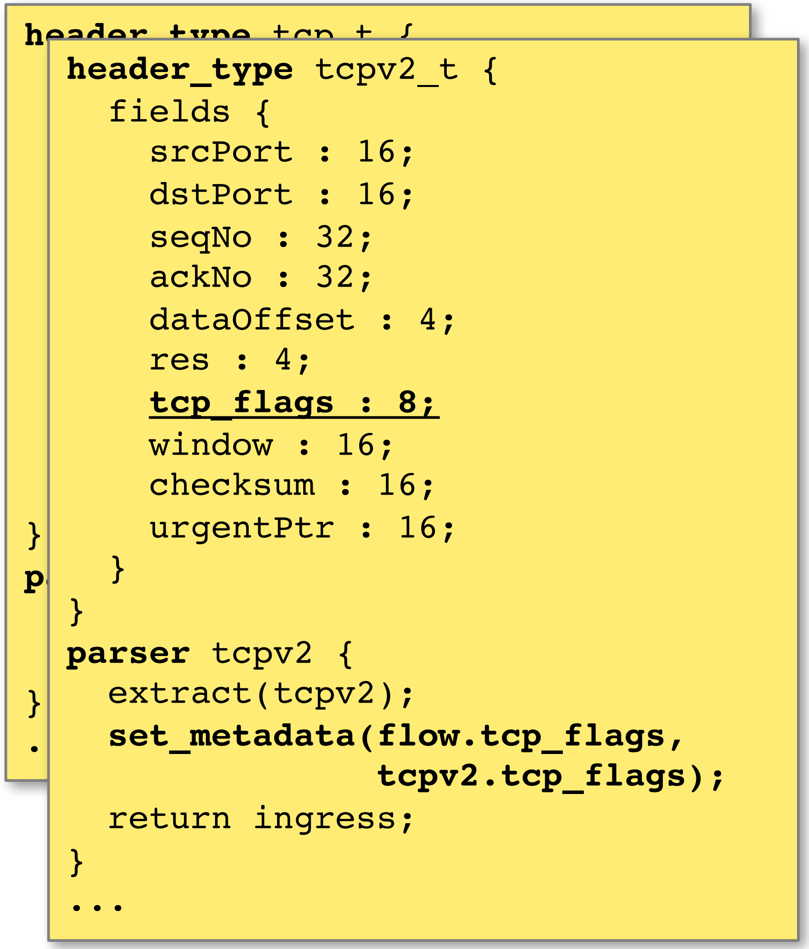


**341** lines of P4 code

Native OVS
Packet Processing Logic

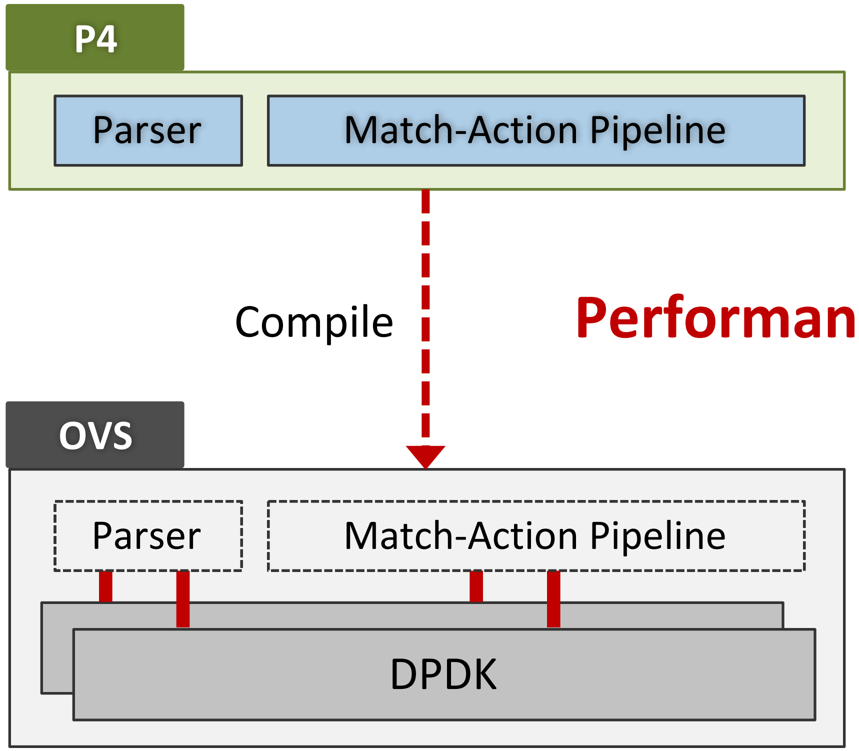
**14,535** lines of C code





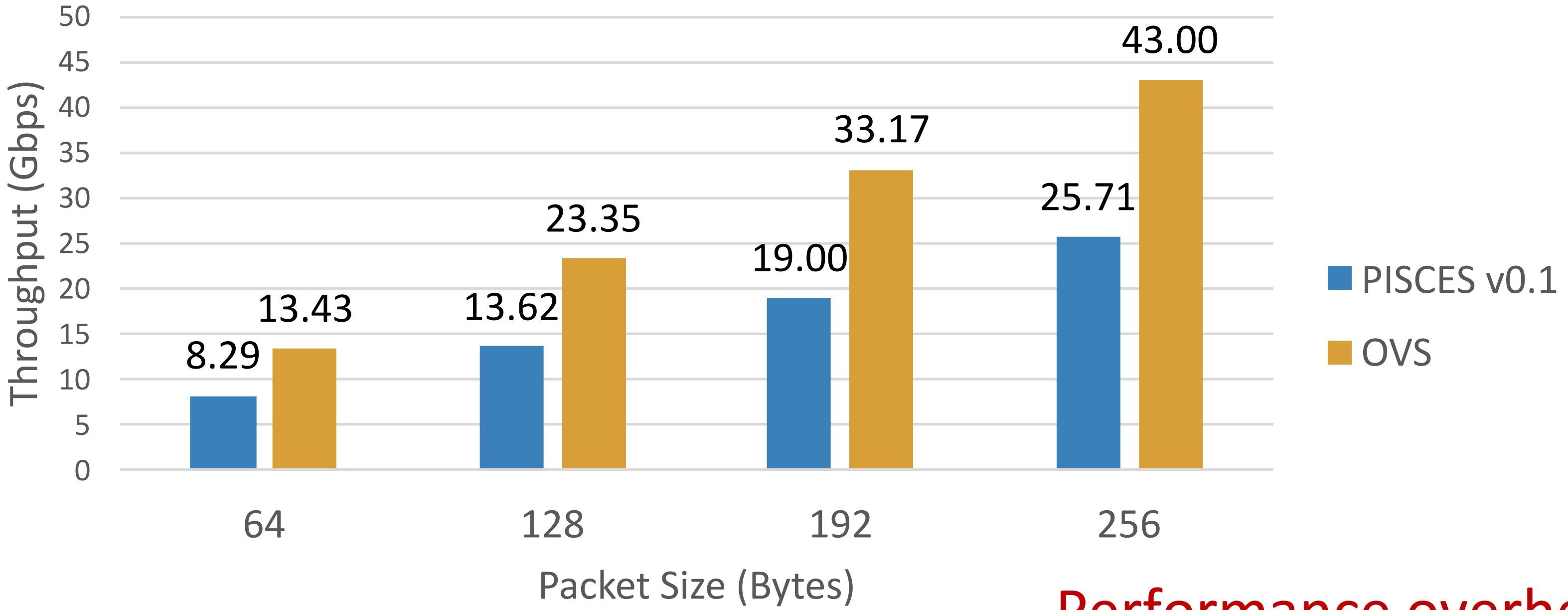
```
header_type tcp_t {
```

```
header_type tcpv2_t {
  fields {
    srcPort : 16;
    dstPort : 16;
    seqNo : 32;
    ackNo : 32;
    dataOffset : 4;
    res : 4;
    tcp_flags : 8;
    window : 16;
    checksum : 16;
    urgentPtr : 16;
  }
}
parser tcpv2 {
  extract(tcpv2);
  set_metadata(flow.tcp_flags,
               tcpv2.tcp_flags);
  return ingress;
}
...
```



**Performance Overhead?**

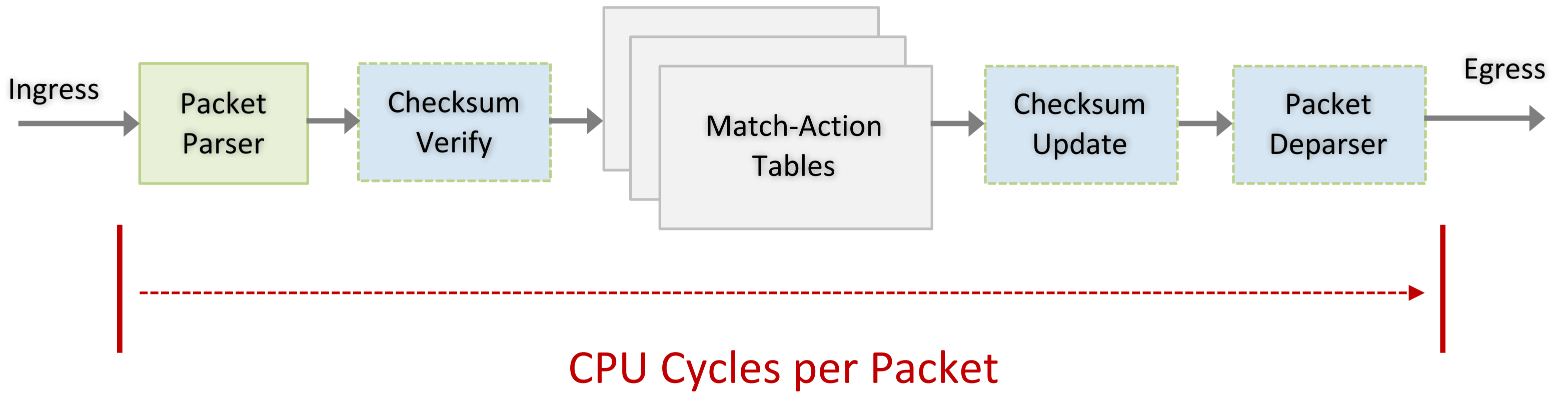
## Throughput on **Eth + IPv4 + ACL** benchmark application



Performance overhead of

**~40%**

## Cause for the Overhead

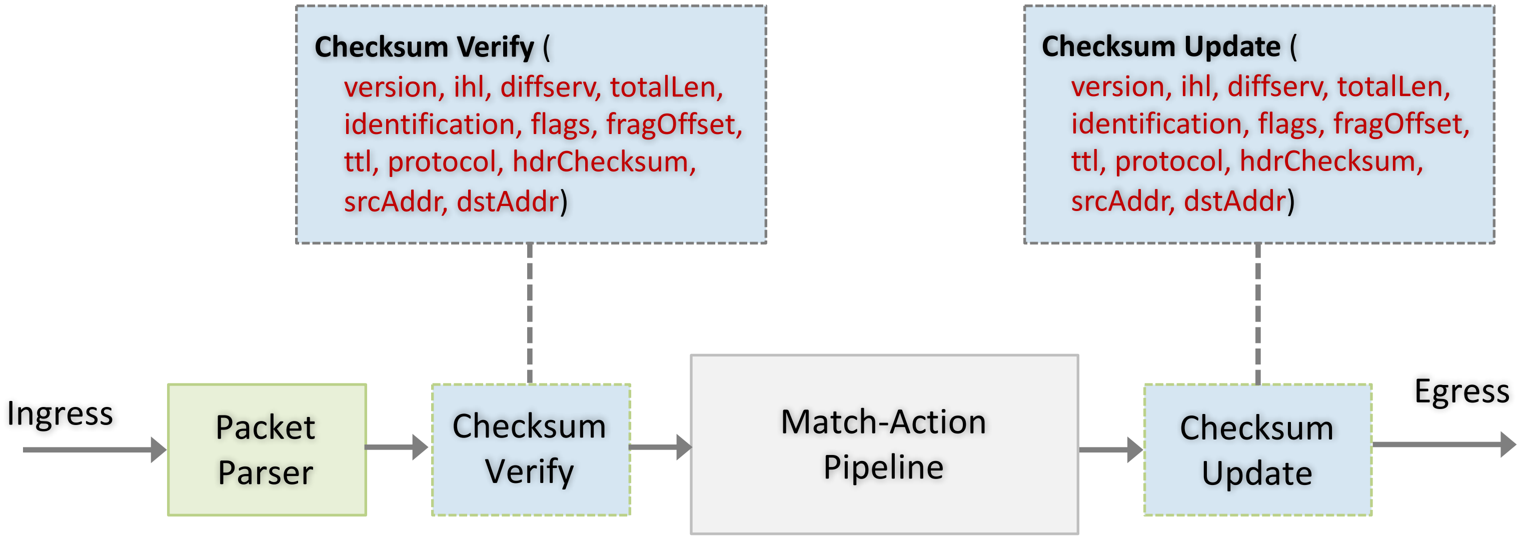


Extra CPU cycles are consumed by

- Checksum computation
- Packet header editing mode

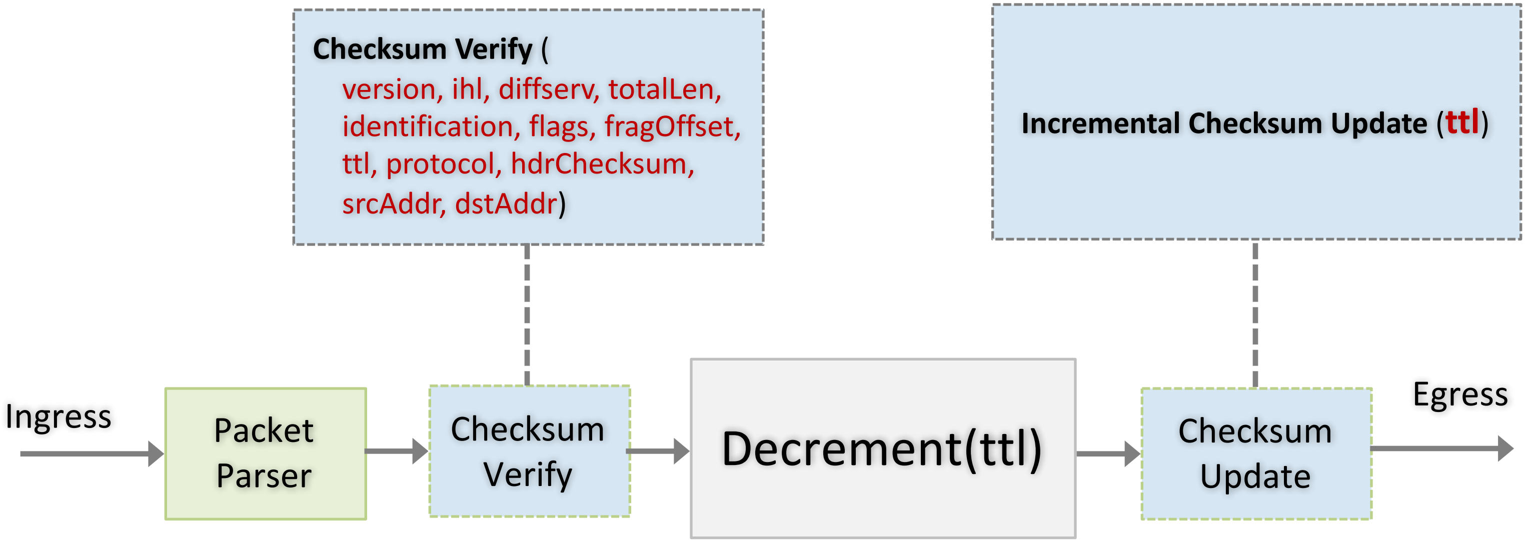
and more…

# Checksum Inefficiencies

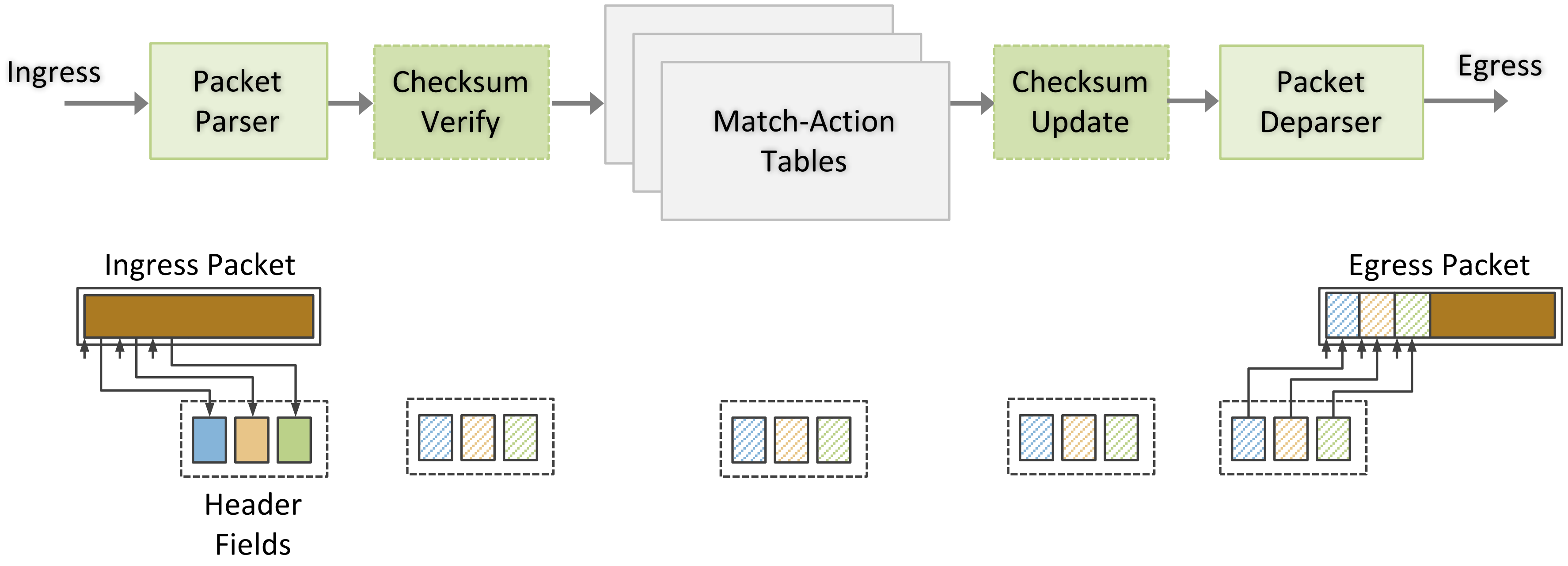


**Checksum Verify** (
version, ihl, diffserv, totalLen,
identification, flags, fragOffset,
ttl, protocol, hdrChecksum,
srcAddr, dstAddr)
**Checksum Update** (
version, ihl, diffserv, totalLen,
identification, flags, fragOffset,
ttl, protocol, hdrChecksum,
srcAddr, dstAddr)
Ingress
Packet Parser
Checksum Verify
Match-Action Pipeline
Checksum Update
Egress

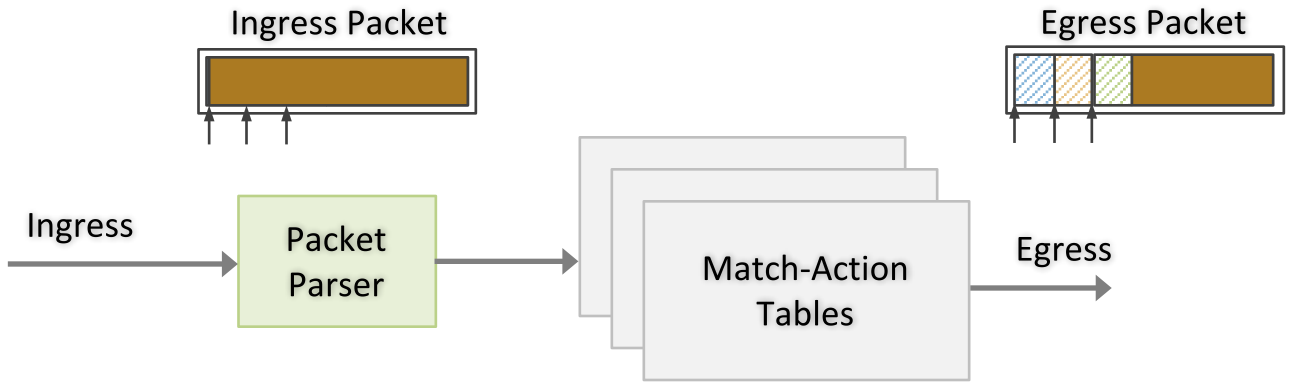
# Checksum Inefficiencies


**Checksum Verify (**
version, ihl, diffserv, totalLen, identification, flags, fragOffset, ttl, protocol, hdrChecksum, srcAddr, dstAddr)
**Incremental Checksum Update (ttl)**
Ingress
Packet Parser
Checksum Verify
Decrement(ttl)
Checksum Update
Egress

# Post-Pipeline Editing



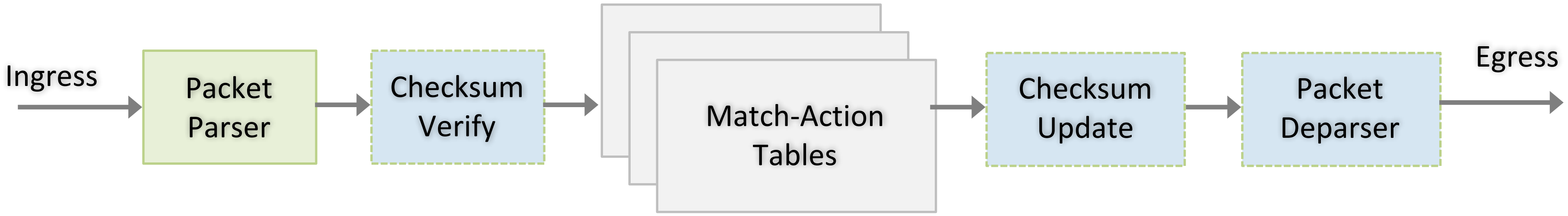
# Inline Editing



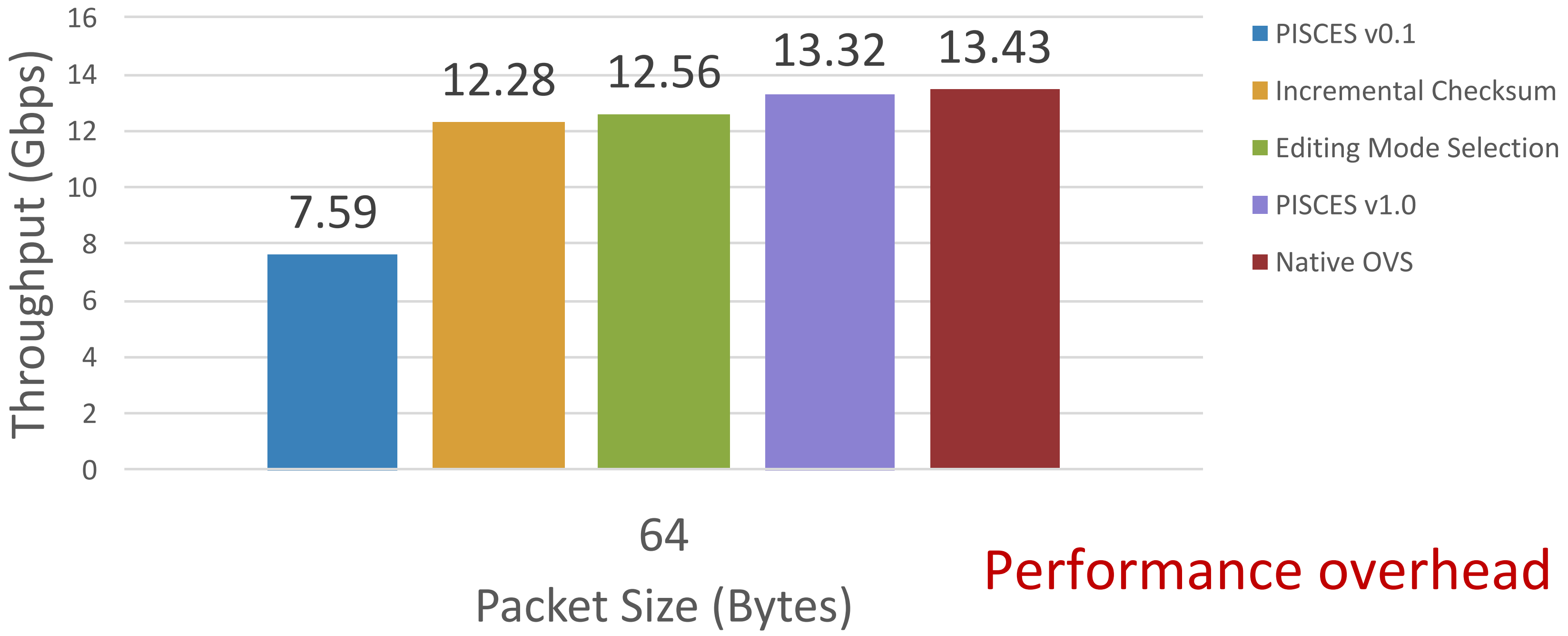
| Editing Mode | Advantage | Disadvantage |
|---|---|---|
| Post-Pipeline | | Extra copy of headers |
| Inline | No extra copy of headers | |

PISCES automatically chooses between

- **Inline Editing**
- **Post-pipeline Editing**

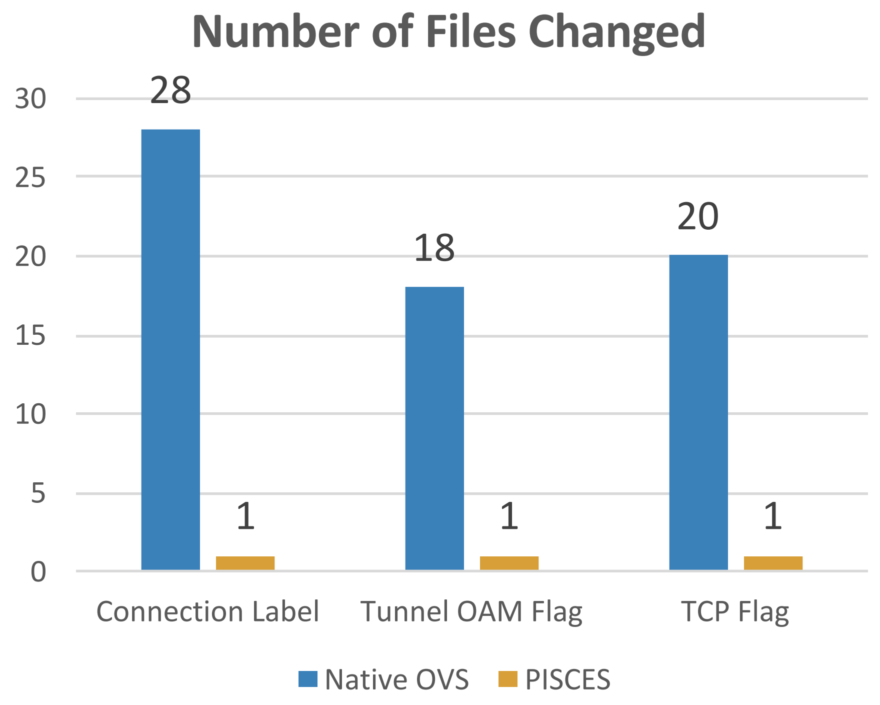


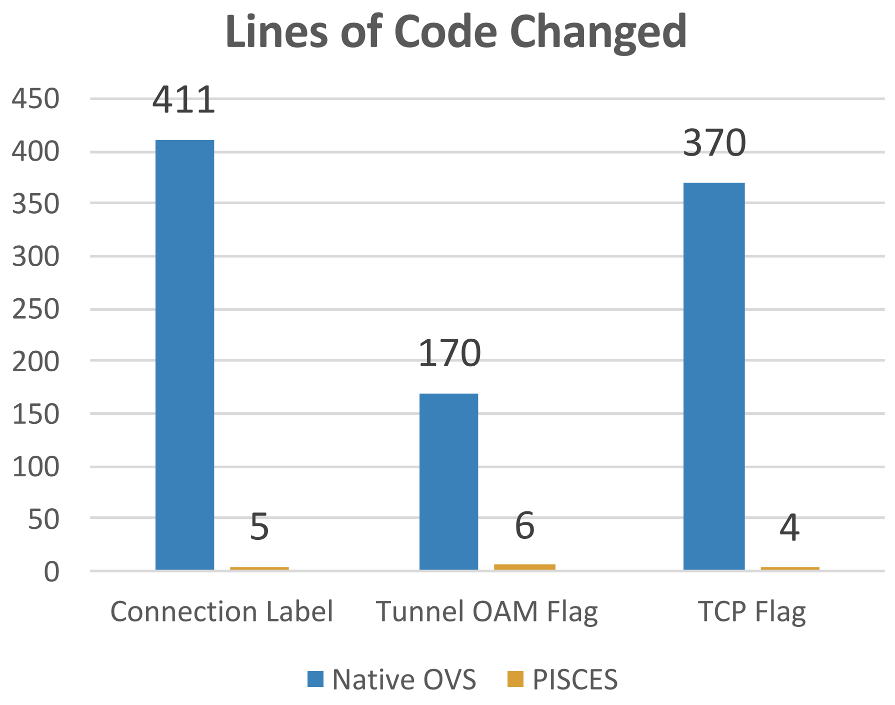
## Throughput on **Eth + IPv4 + ACL** benchmark application



Performance overhead of

**< 2%**


Number of Files Changed
30
25
20
15
10
5
0
28
1
18
1
20
1
Connection Label
Tunnel OAM Flag
TCP Flag
Native OVS
PISCES


Lines of Code Changed
450
400
350
300
250
200
150
100
50
0
411
5
170
6
370
4
Connection Label
Tunnel OAM Flag
TCP Flag
Native OVS
PISCES

- A method to **quickly develop and deploy** packet processing logic on a software switch
- With hardly **any performance cost!**

**Learn more** and **Try** PISCES here:

[http://pisces.cs.princeton.edu](http://pisces.cs.princeton.edu)