

# PISCES: A Programmable, Protocol-Independent Software Switch

**P4 + OVS == Fast Forwarding!**

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



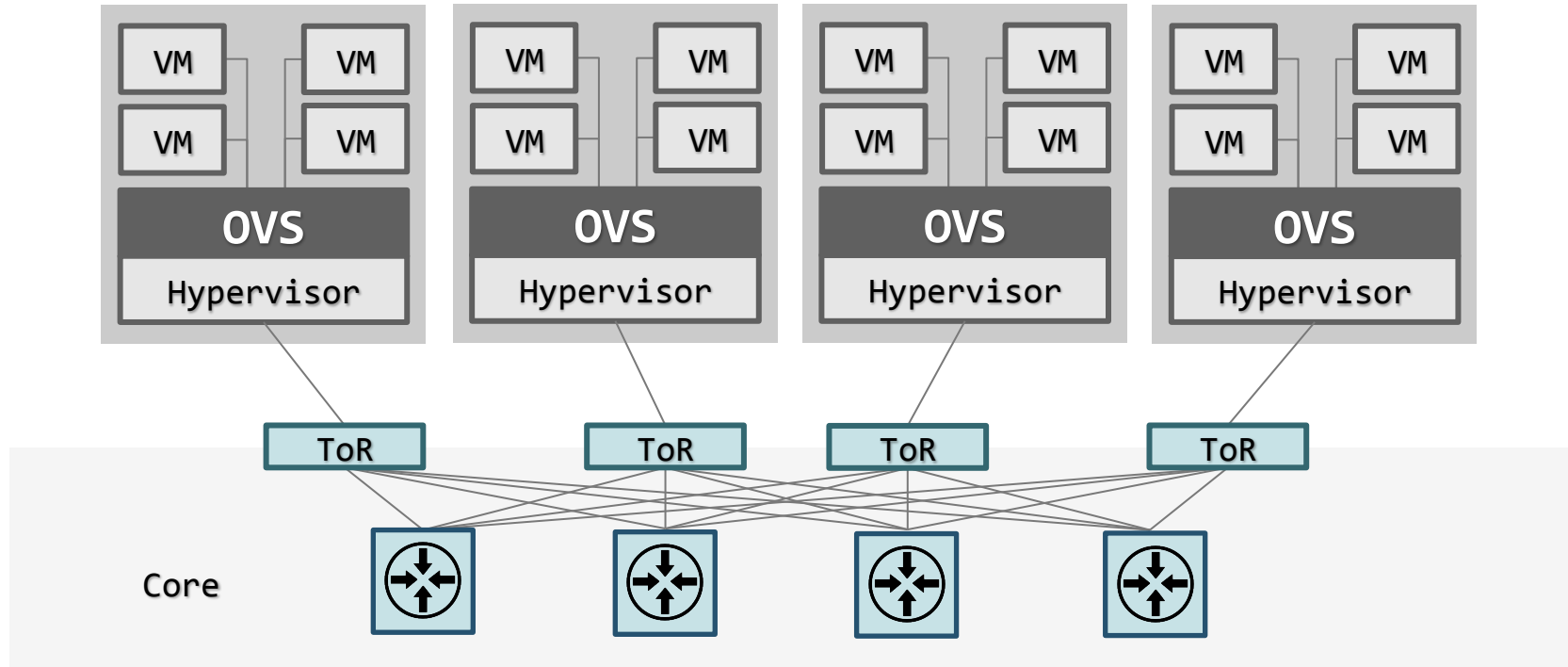
vmware

**BAREFOOT**  
NETWORKS

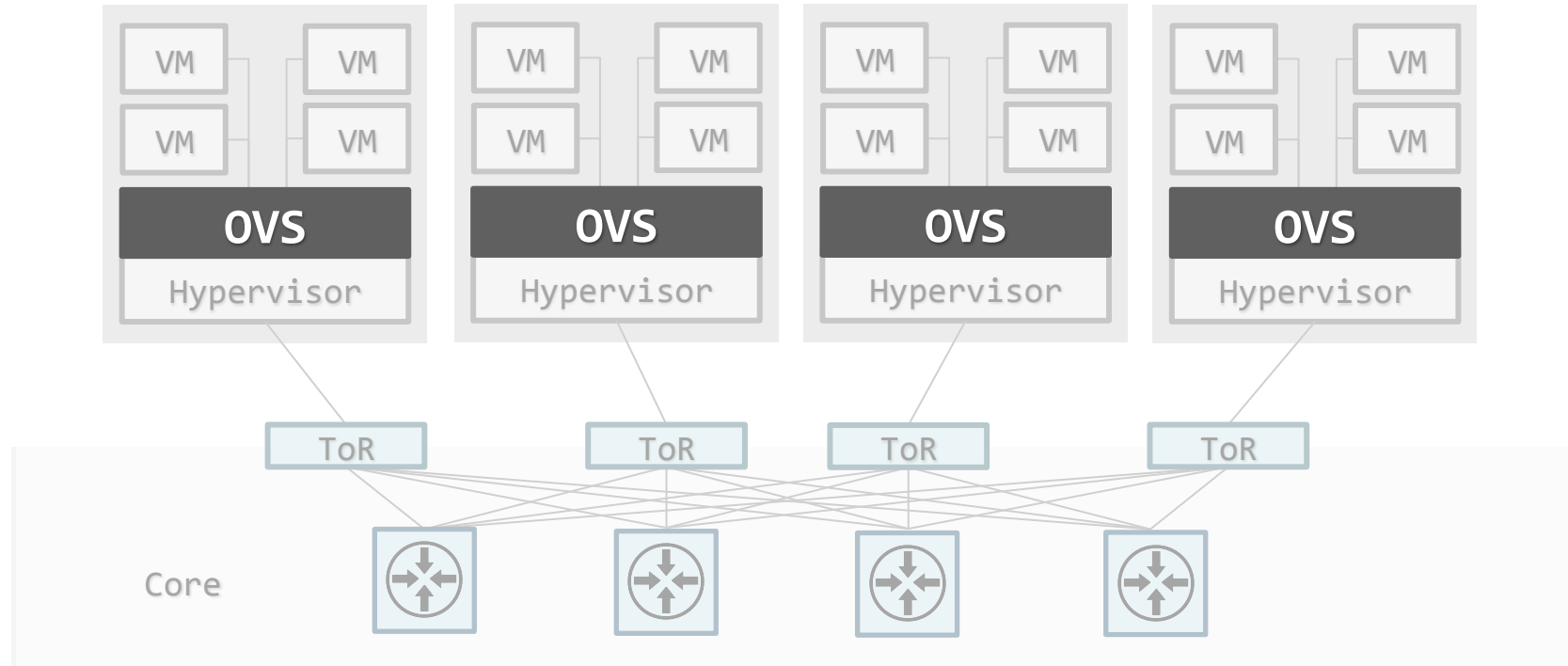
**Also appears at SIGCOMM 2016!**

<http://goo.gl/wmBmTu>

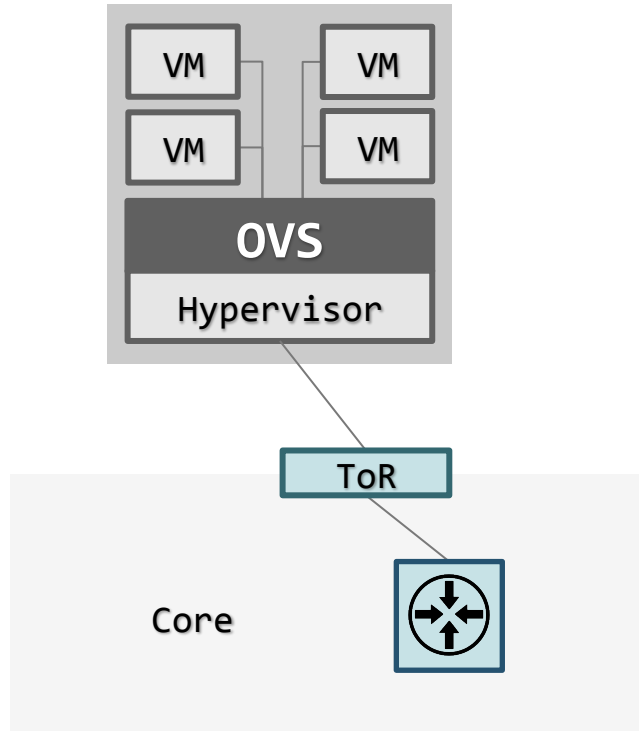
# Importance of Software Switches



# Importance of Software Switches



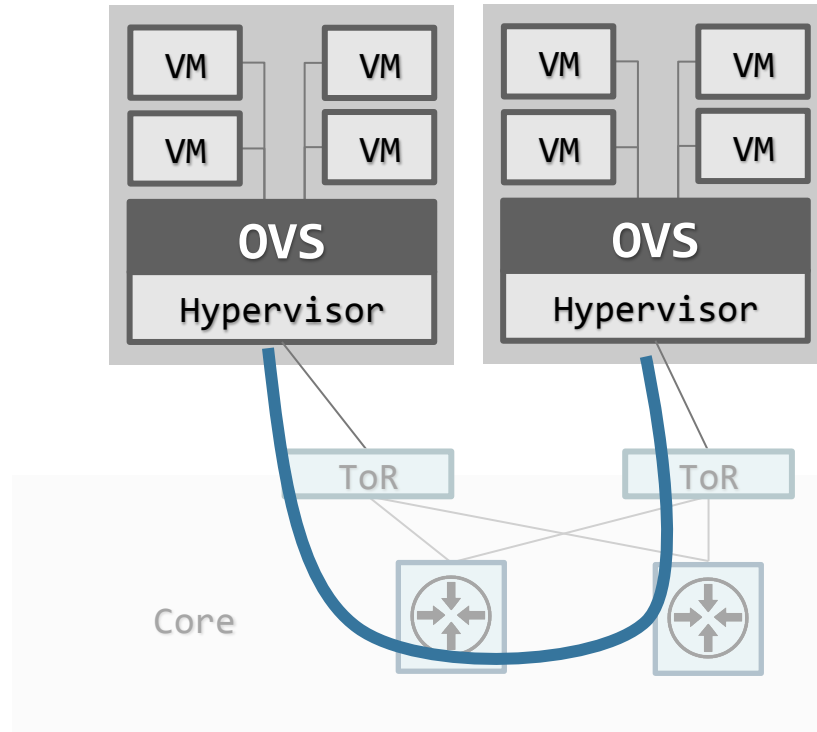
# Ease of Customization?



Enable **Rapid Development** and  
**Deployment of Network Features!**

**Is it REALLY the case?**

# Ease of Customization?

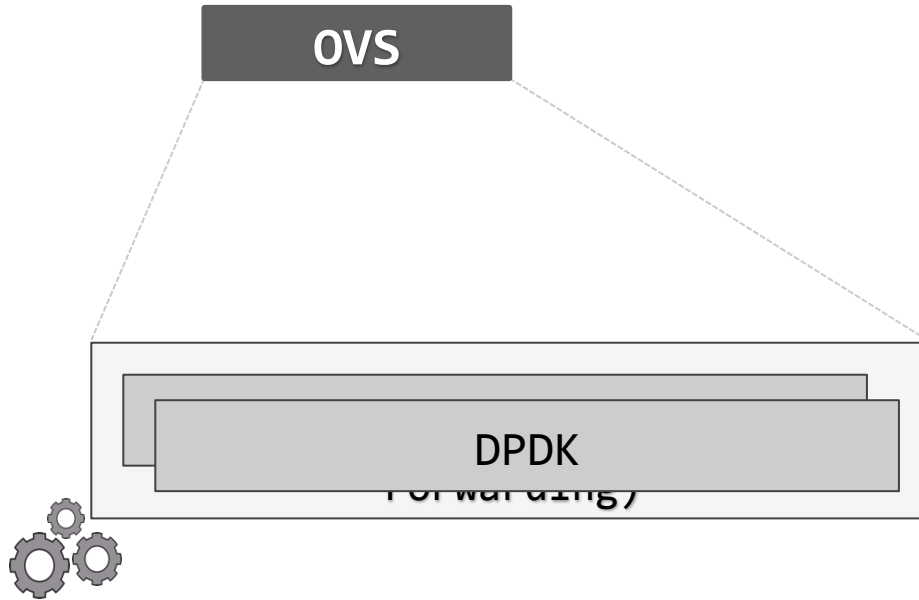


For example, OVS supports following tunneling protocols:

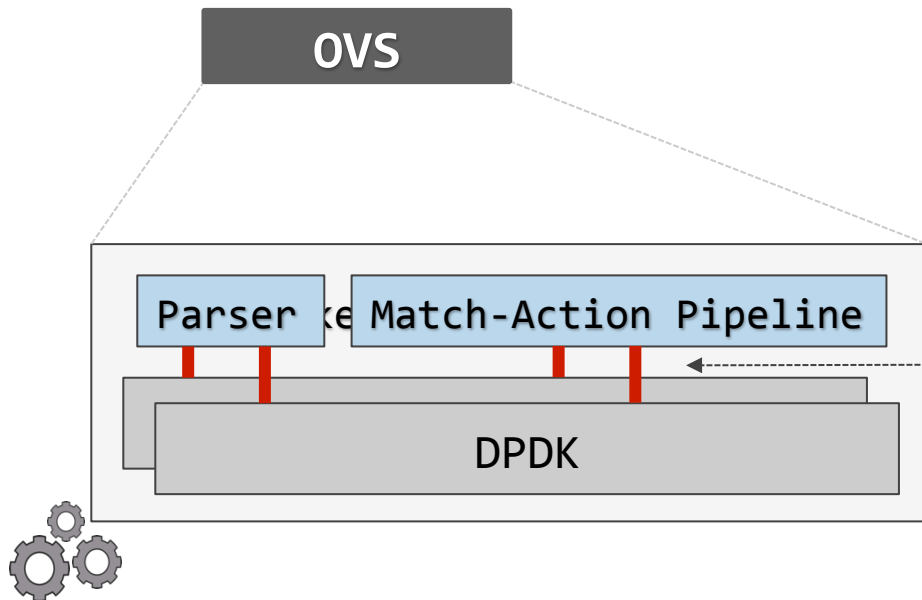
- VXLAN: Virtual Extensible LAN
- STT: Stateless Transport Tunneling
- NVGRE: Network Virtualization Generic Routing

**What about adding new protocols?**

# Rapid Development & Deployment?



# Rapid Development & Deployment?

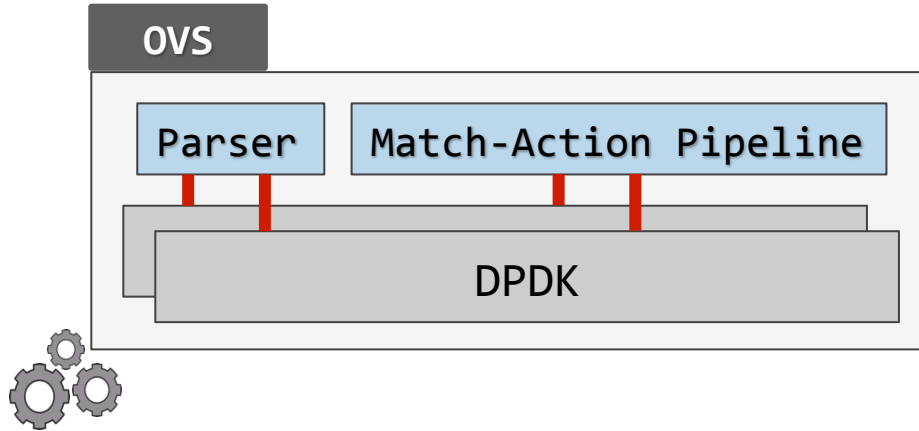


Requires domain expertise in:

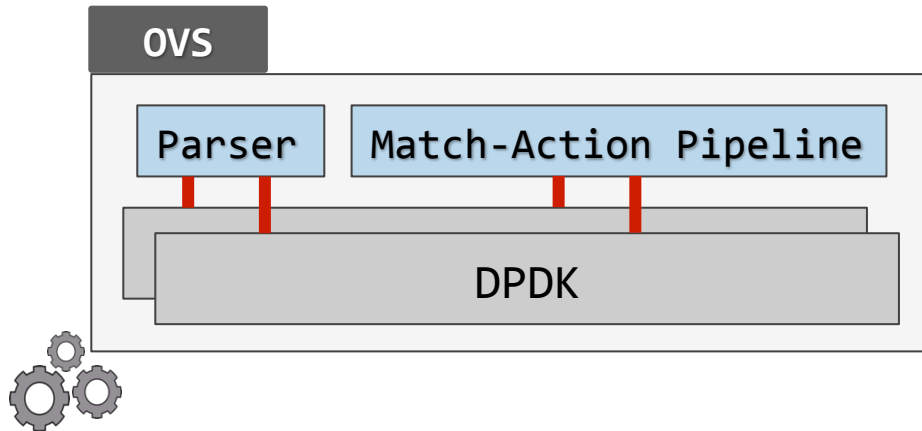
- Network **protocol design**
  - Software development
    - **Develop**
    - **Test**
    - **Deploy**
- ... large, complex codebases.
- Arcane APIs**
- Can take **3-6 months** to get a new feature in.
  - **Maintaining changes** across releases



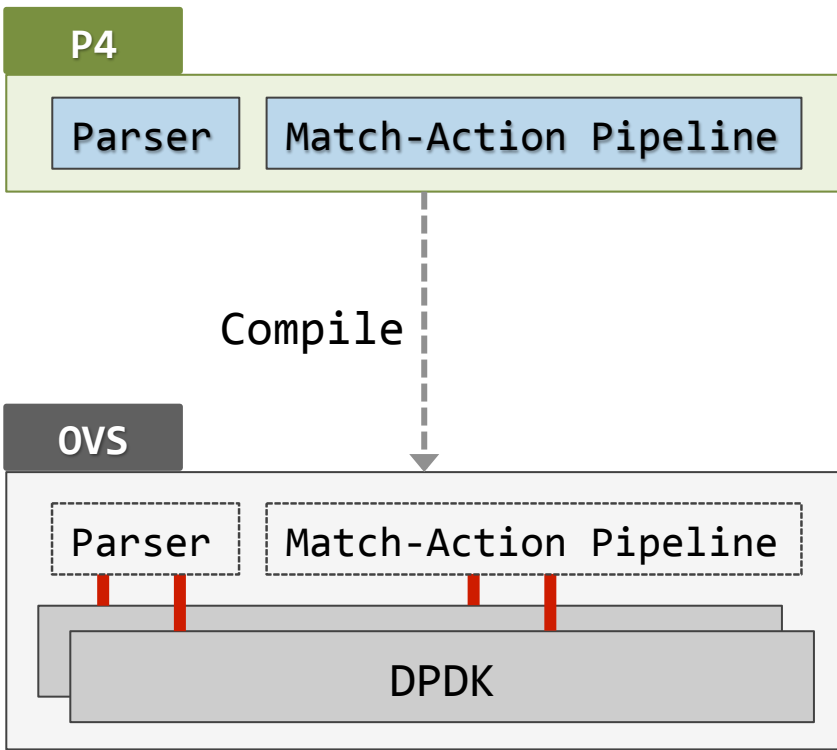
# Rapid Development & Deployment?



# Rapid Development & Deployment?



# Rapid Development & Deployment?

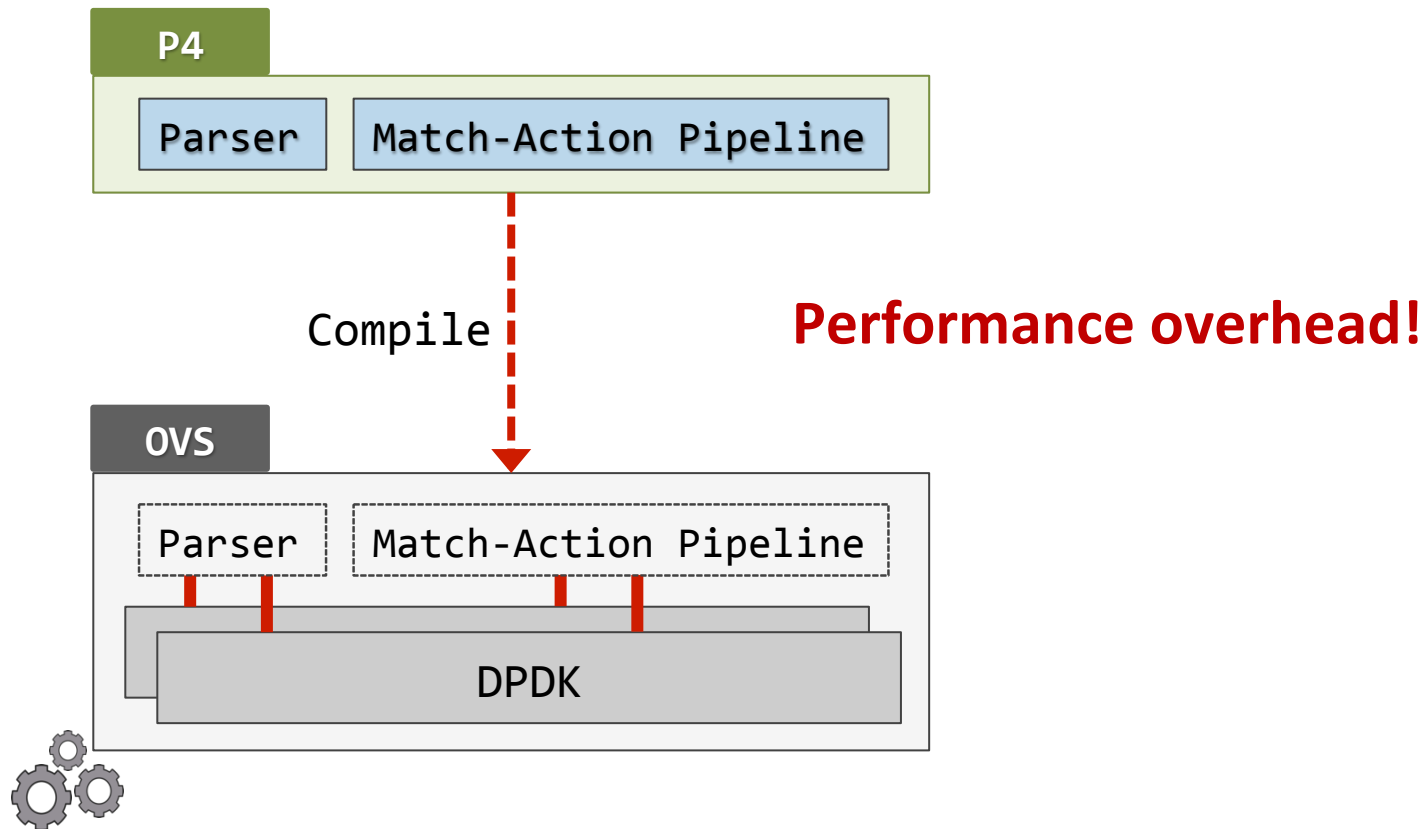


**341** lines of code

**Native OVS**

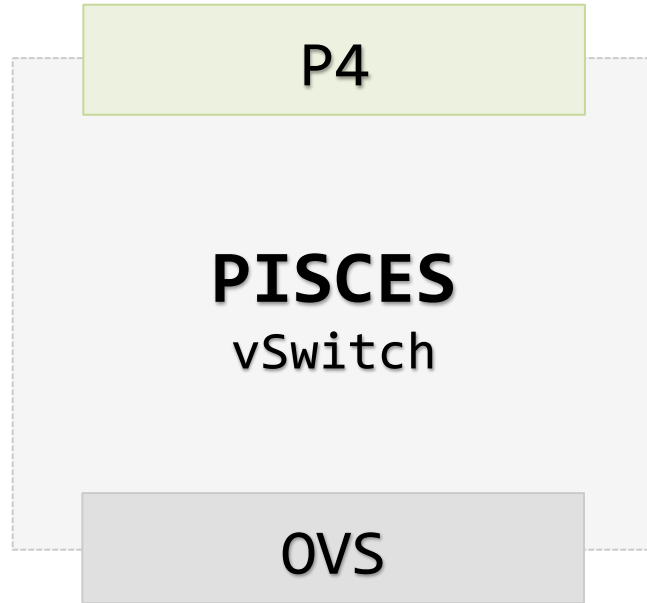
**14,535** lines of code

# Rapid Development & Deployment?

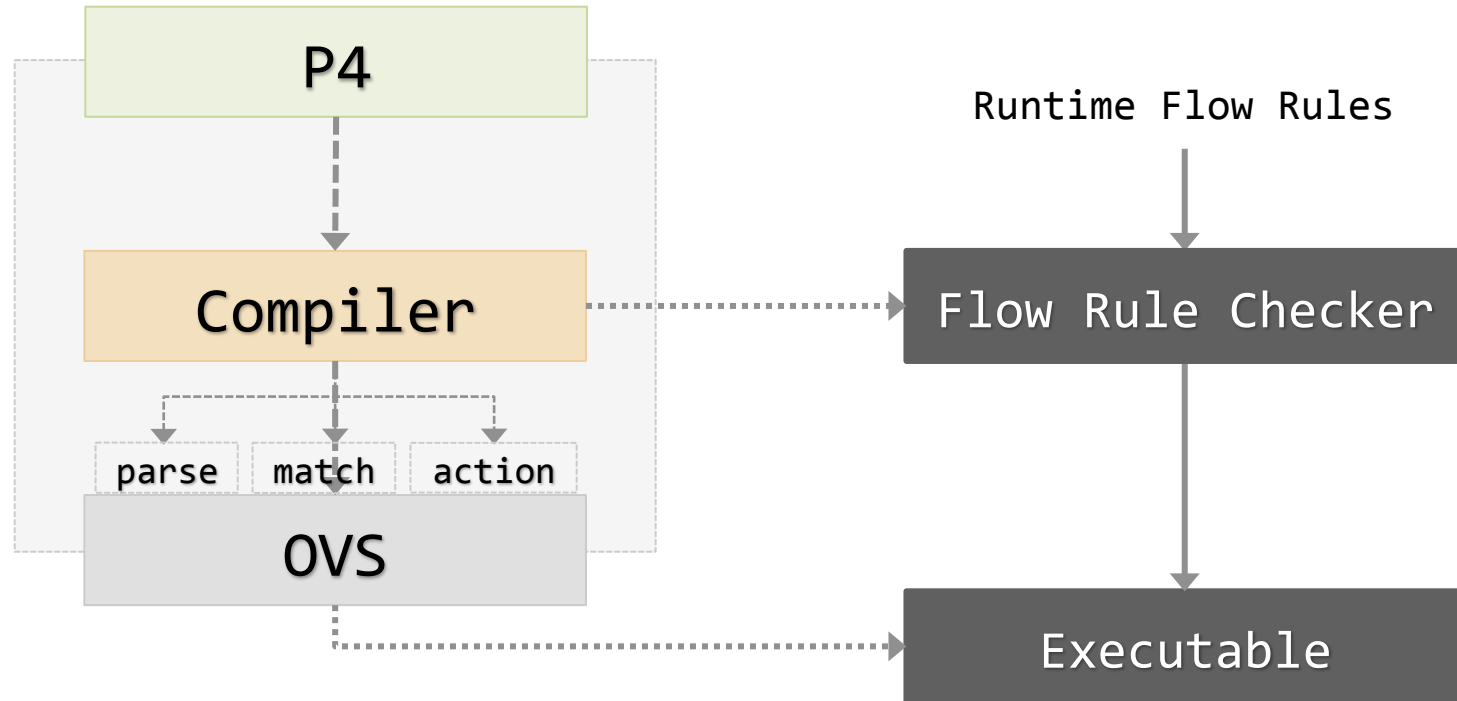


What's the **cost of programmability** on  
**Performance?**

# PISCES: A Protocol-Independent Software Switch



# PISCES: A Protocol-Independent Software Switch

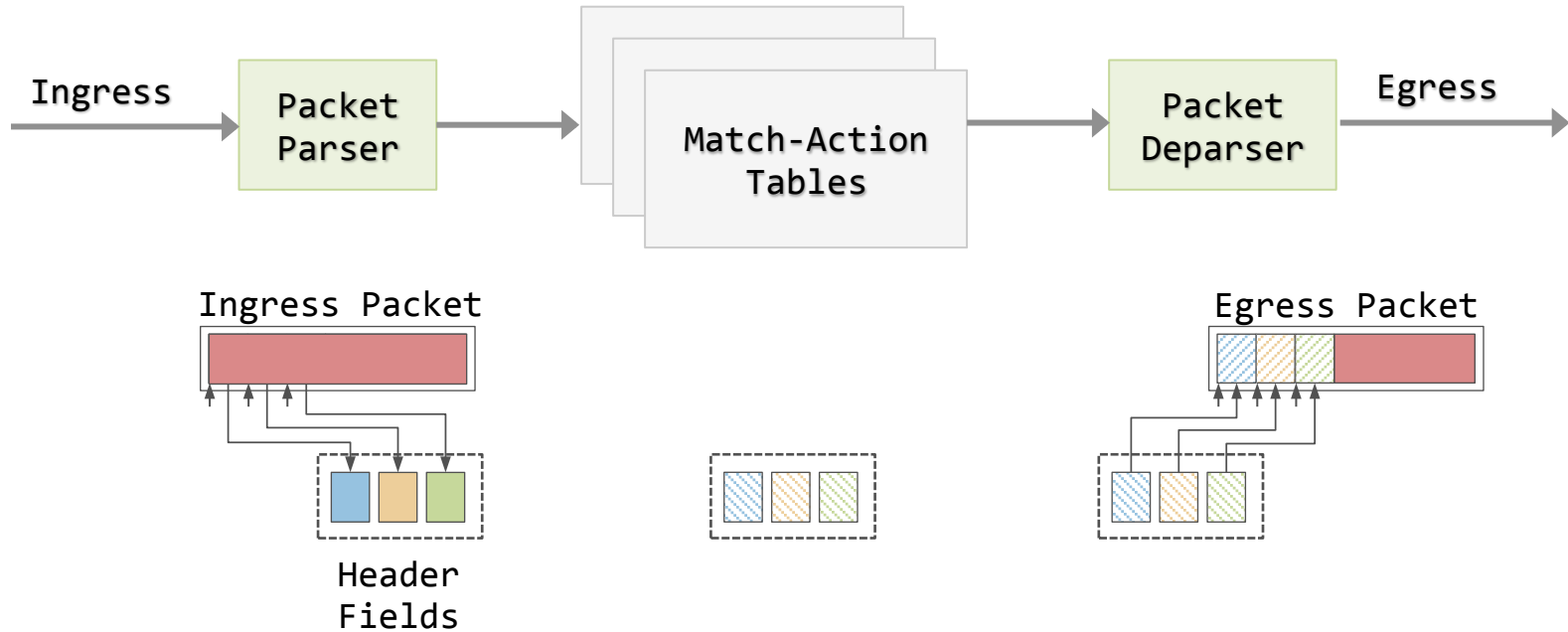


# PISCES: A Protocol-Independent Software Switch

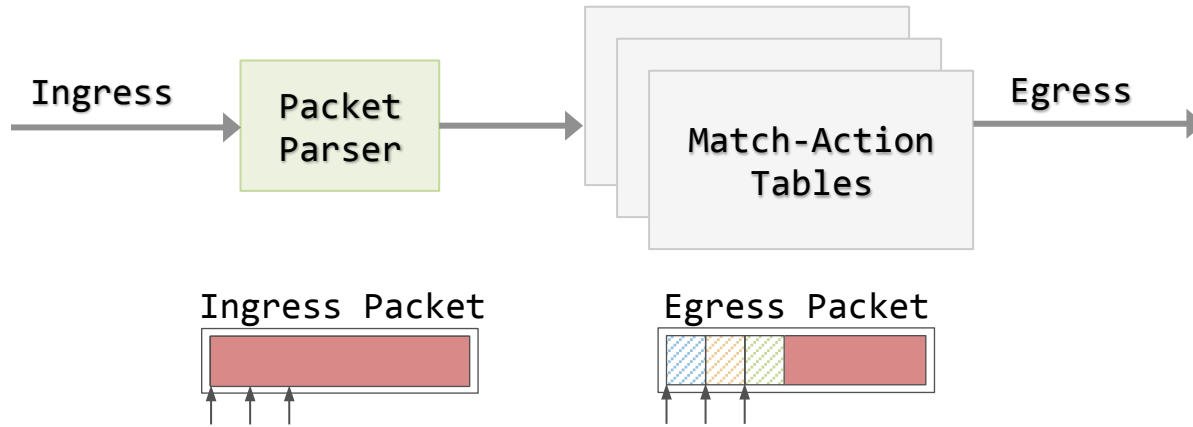
- P4 and OVS **packet forwarding models**.
- **Performance overhead** of a **naïve mapping** from P4 to OVS.
- PISCES **compiler optimizations** to reduce the performance overhead.



# P4 Forwarding Model (or Post-Pipeline Editing)



# OVS Forwarding Model (or Inline Editing)



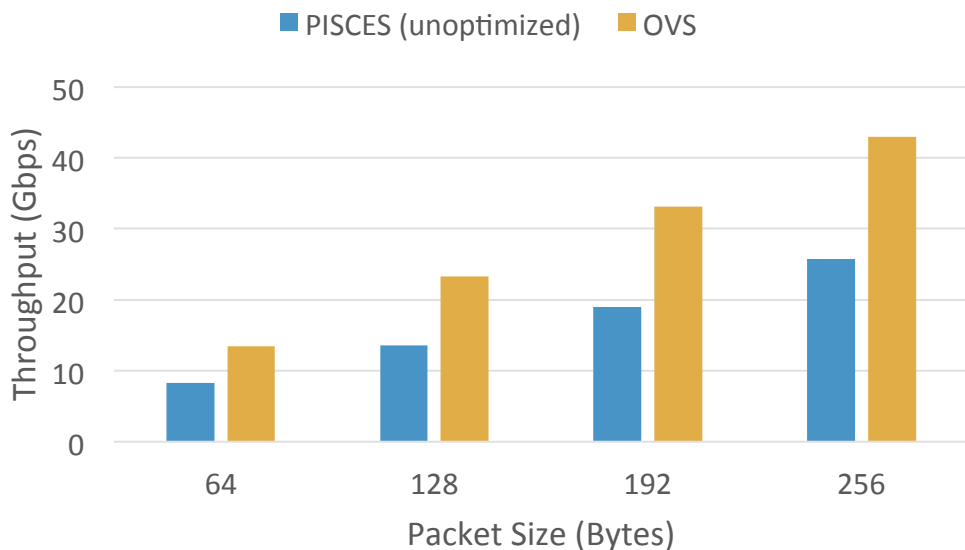
# (Modified) OVS Forwarding Model



- Supports both editing modes:
  - **Inline Editing**
  - **Post-pipeline Editing**

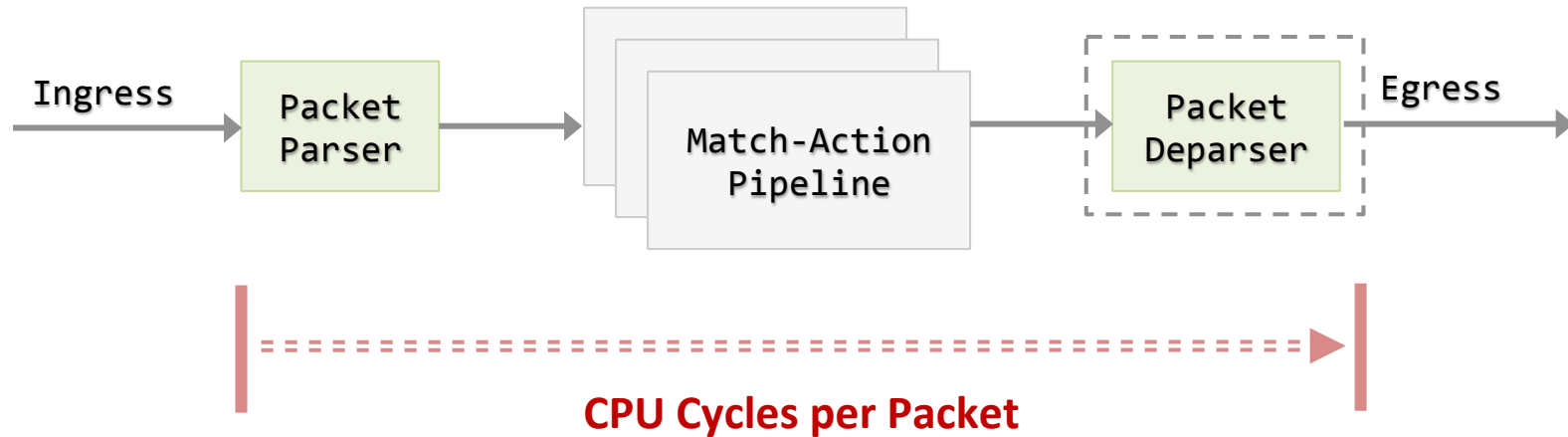
# Naïve Mapping from P4 to OVS

A naïve compilation of **L2L3-ACL** benchmark application



Performance overhead of  
~ **40%**

## Causes of Performance Degradation



# Causes of Performance Degradation

- Factors affecting CPU cycles:
  - **Extra copy of headers** in the post-pipeline editing mode
  - **Fully-specified checksum** calculation
  - **Redundant parsing** of header fields and more ...

# Causes of Performance Degradation

## Factor #1: Extra copy of headers

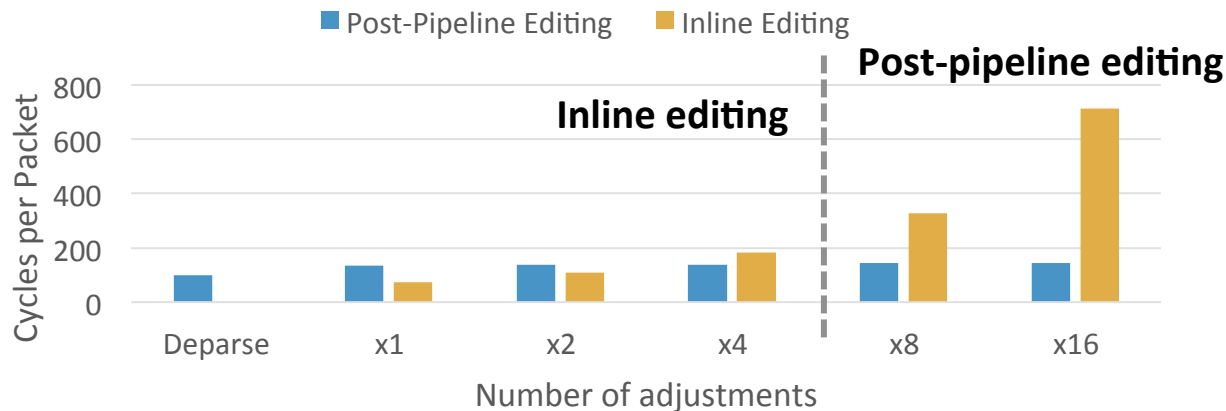
Editing Mode	Pros	Cons
Post-Pipeline		Extra copy of headers
Inline	No extra copy of headers	

- **Post-pipeline** editing consumes **2x** more cycles than **inline** editing when **parsing VXLAN protocol**.

# Causes of Performance Degradation

## Factor #1: Extra copy of headers

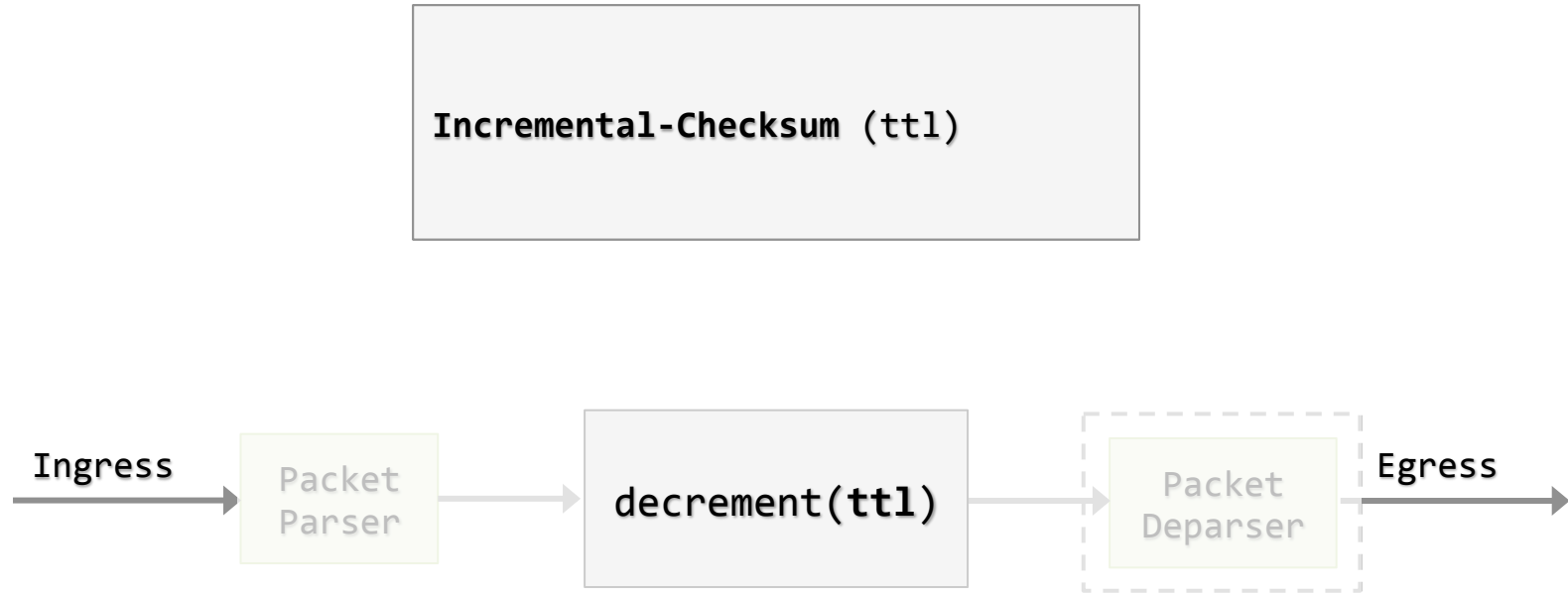
Editing Mode	Pros	Cons
Post-Pipeline	Packets are adjusted once	Extra copy of headers
Inline	No extra copy of headers	Multiple adjustments to packet





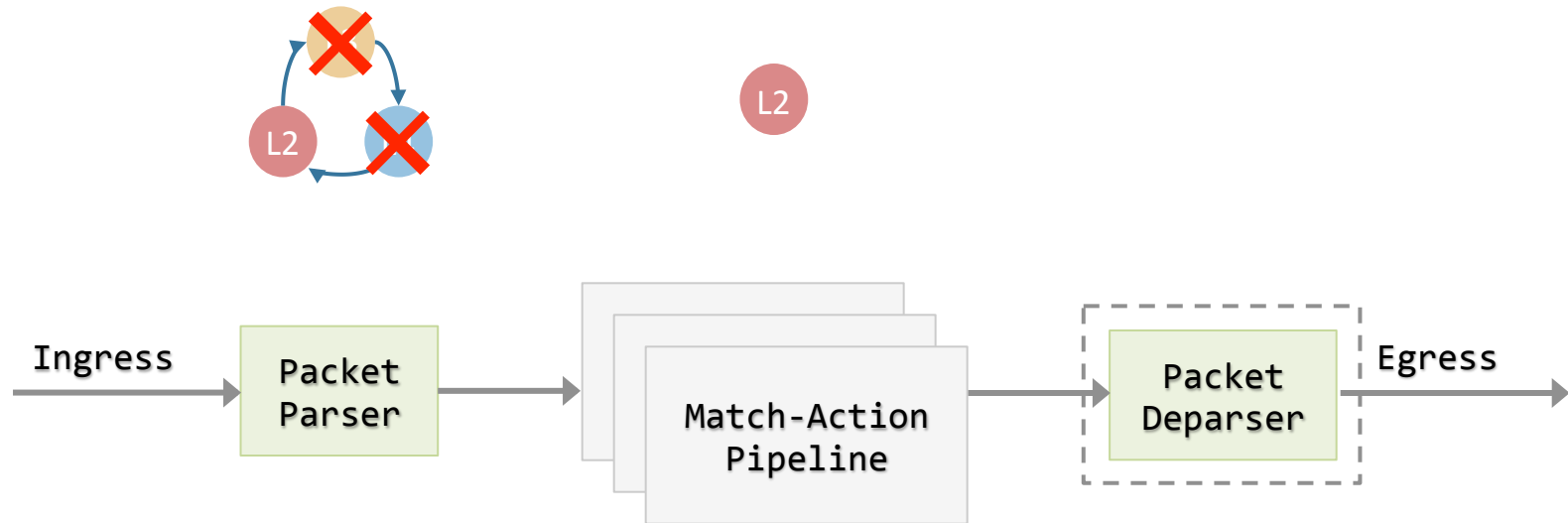
# Causes of Performance Degradation

## Factor #2: Fully-Specified Checksums



# Causes of Performance Degradation

## Factor #3: Redundant parsing of headers



# Optimizing for CPU Cycles

## Optimizations

Inline vs. post-pipeline editing

Incremental checksum

Parser specialization

Action specialization

Action coalescing

# Optimizing for CPU Cycles

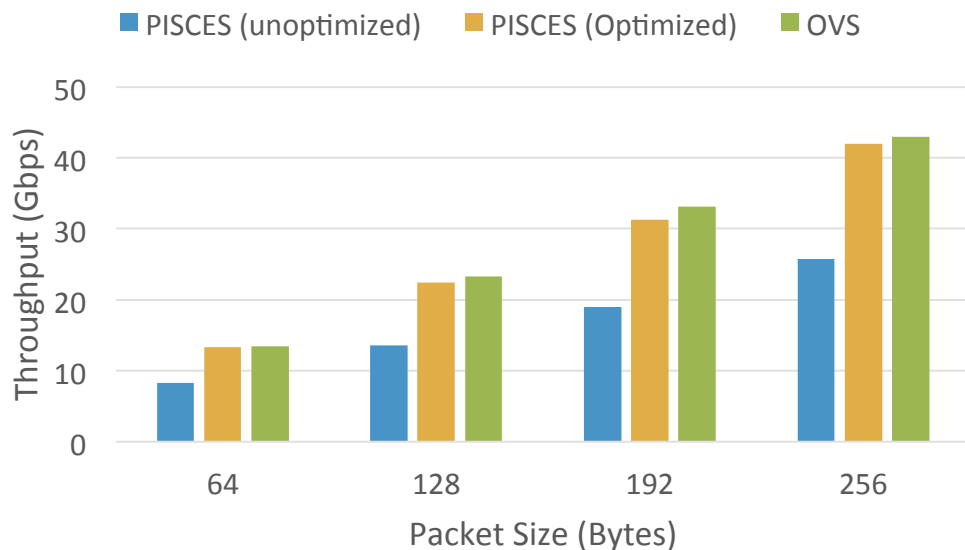
Optimizations	
Inline vs. post-pipeline editing	Extra Copy of Headers
Incremental checksum	Fully-Specified Checksum
Parser specialization	Redundant Parsing
Action specialization	
Action coalescing	

# Optimizing for CPU Cycles

Optimizations
Inline vs. post-pipeline editing
Incremental checksum
Parser specialization
Action specialization
Action coalescing

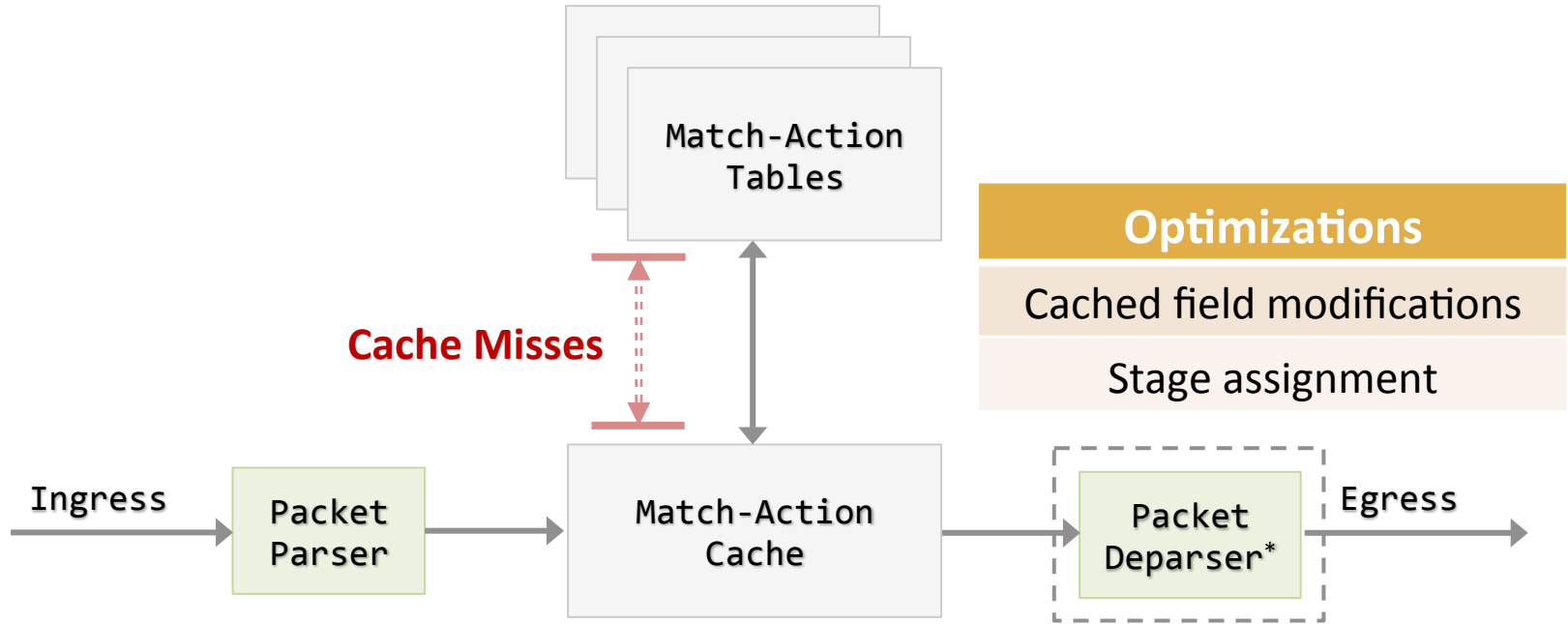
# Optimized Mapping from P4 to OVS

All optimizations together



Performance overhead of  
**< 2%**

# Another Cause for Performance Degradation



# Next Steps

- Support for **stateful memories** and **INT**
- **Integration** with the **mainline OVS**
  - **Interning at VMware to make this happen!**



# Summary

With appropriate compiler optimizations ...

**P4 + OVS == Fast Forwarding!**

# Questions?

Learn more and try PISCES here:

<https://github.com/P4-vSwitch>

Muhammad Shahbaz

[mshahbaz@cs.princeton.edu](mailto:mshahbaz@cs.princeton.edu)



vmware®

**BAREFOOT**  
NETWORKS