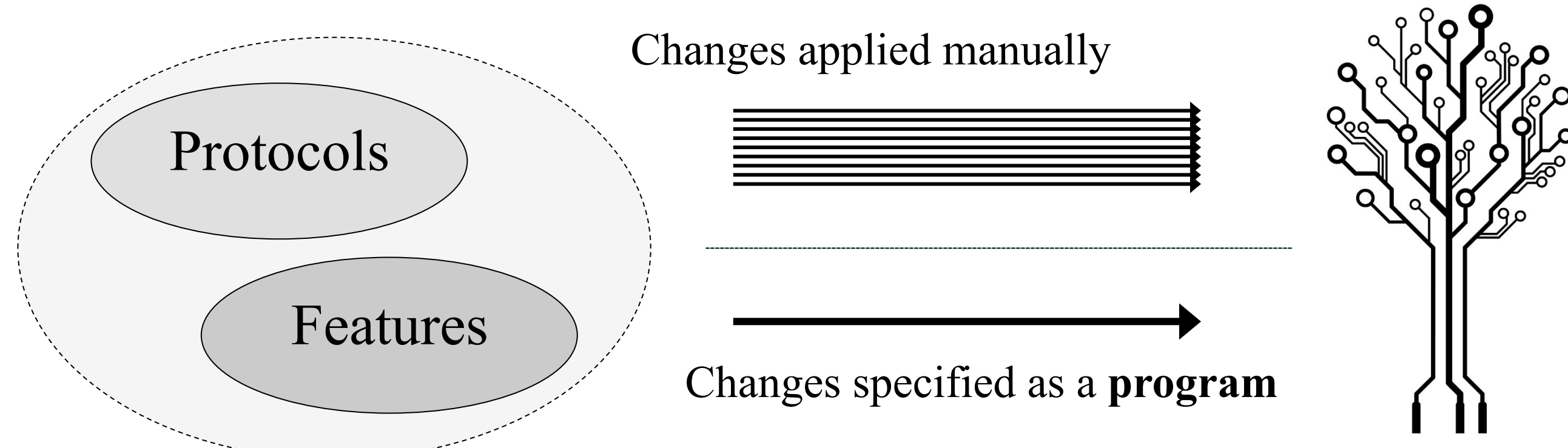


# PISCES: A Programmable, Protocol-Independent Software Switch

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

## 1. Problem Statement

- Every new protocol or feature requires changes across multiple locations in the source tree.
- At present these changes are applied manually.

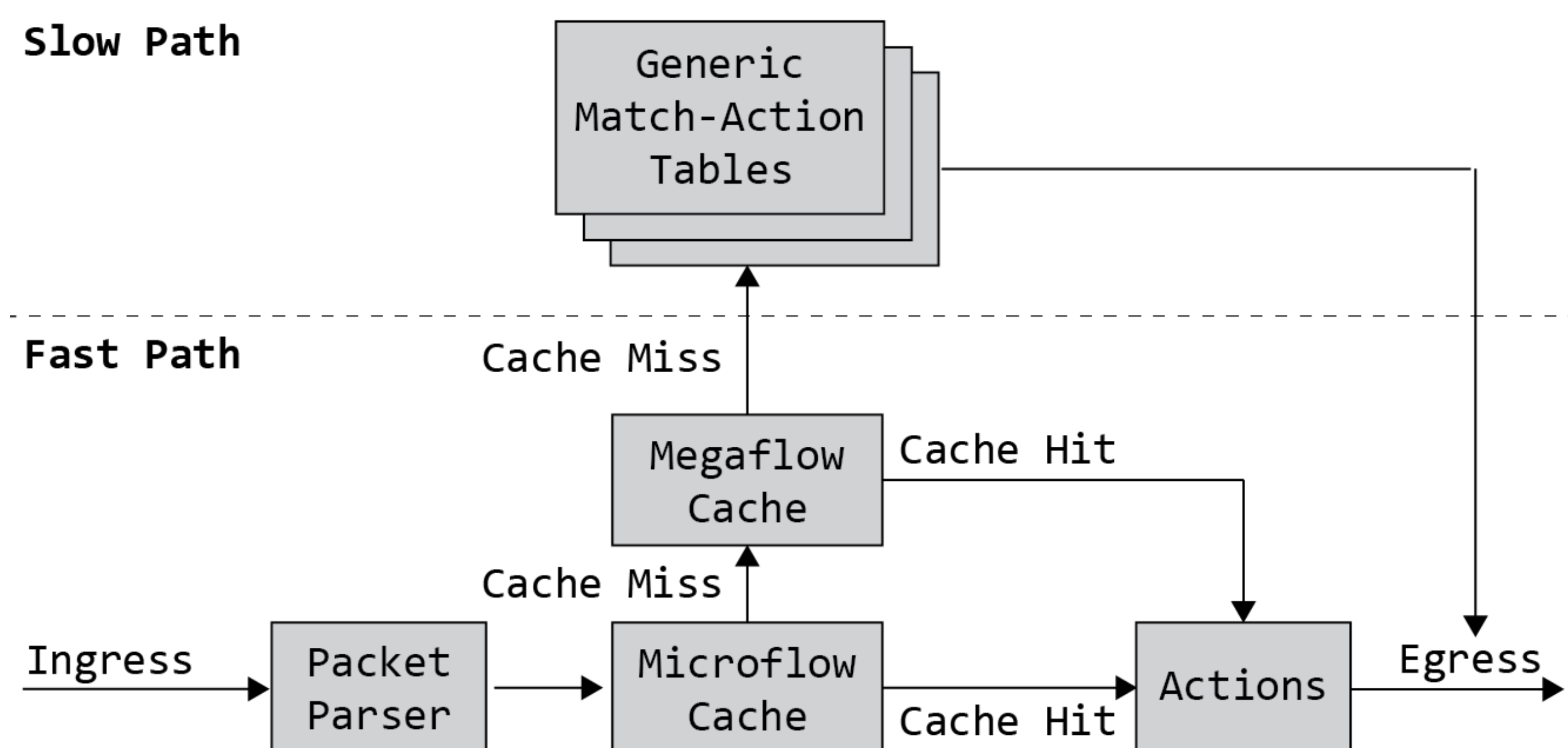


- What if we let system administrators describe these changes as **P4** programs? This has many benefits:
  - a. Rapid addition of new protocols and features
  - b. Automation via compiler can lead to correct behavior
  - c. A single system for coordinating all packet processing

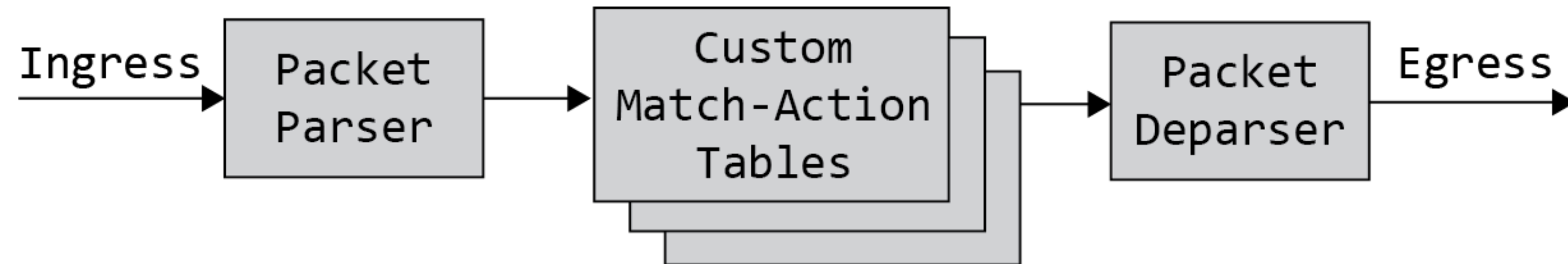
## 2. Approach

- We present a P4 to OvS compiler.
  - OvS is one of the most widely used software switch
  - P4 and OvS share a similar abstract forwarding model

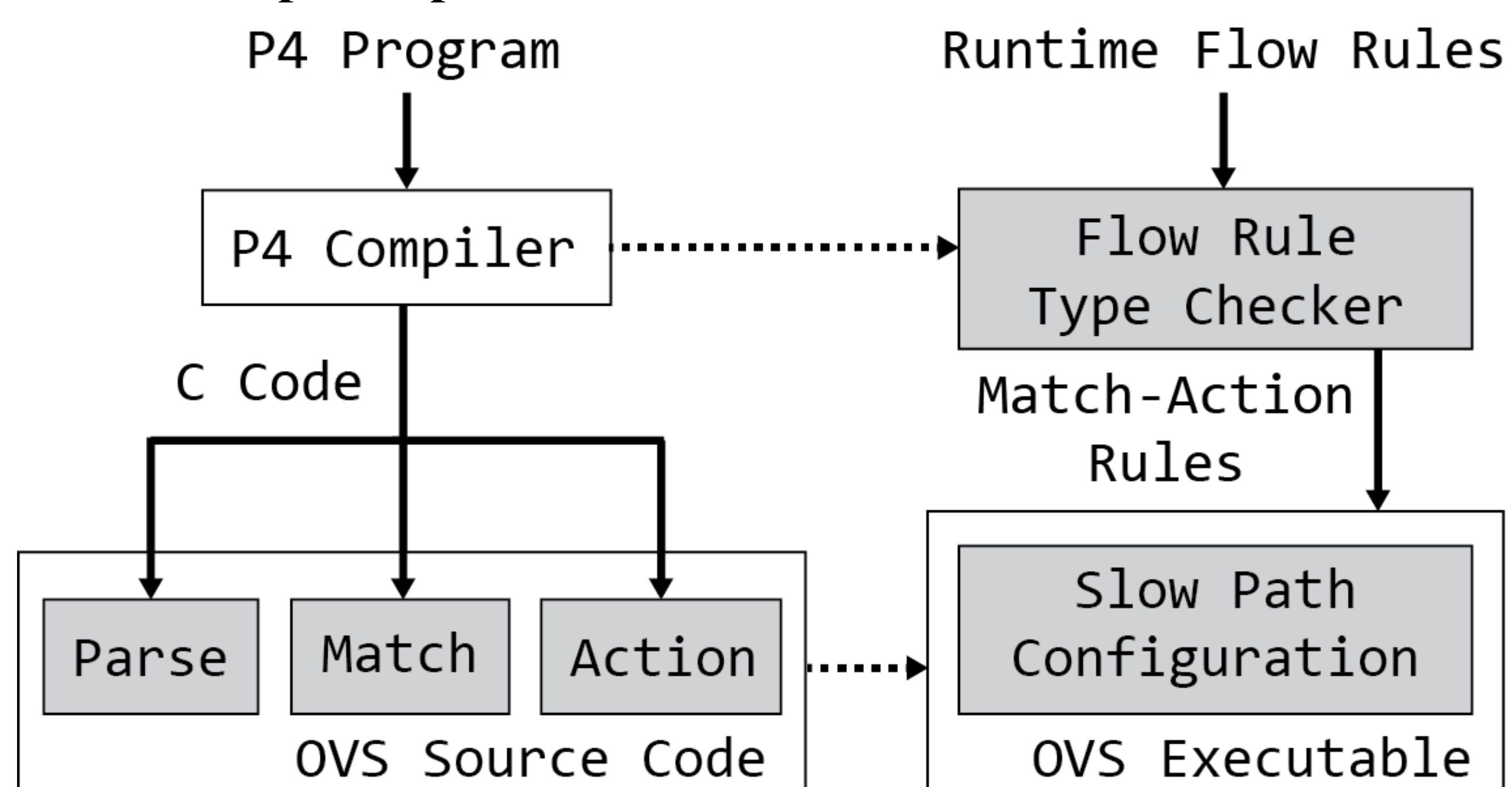
### OvS Abstract Forwarding Model



### P4 Abstract Forwarding Model

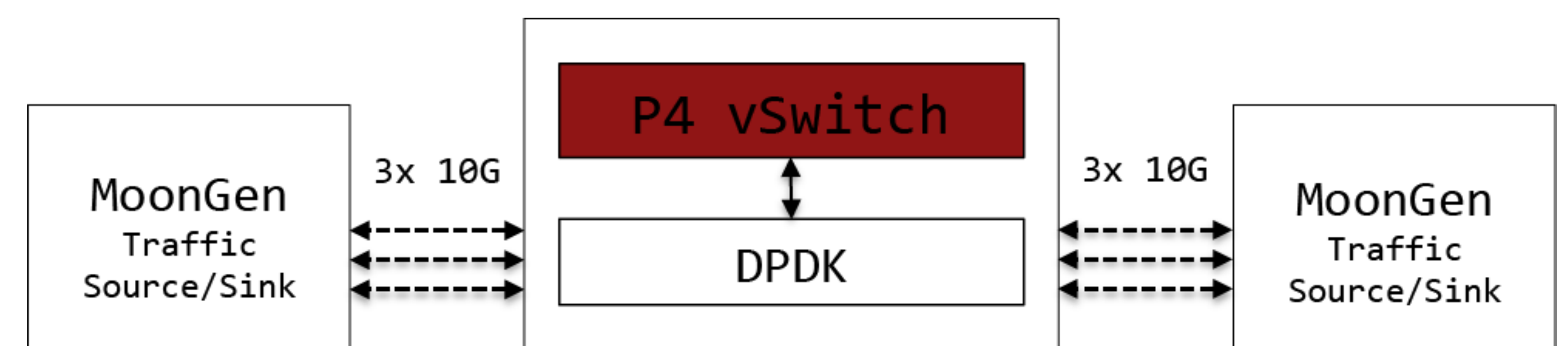


### P4-OvS Compiler Specifications



## 3. Experimental Setup

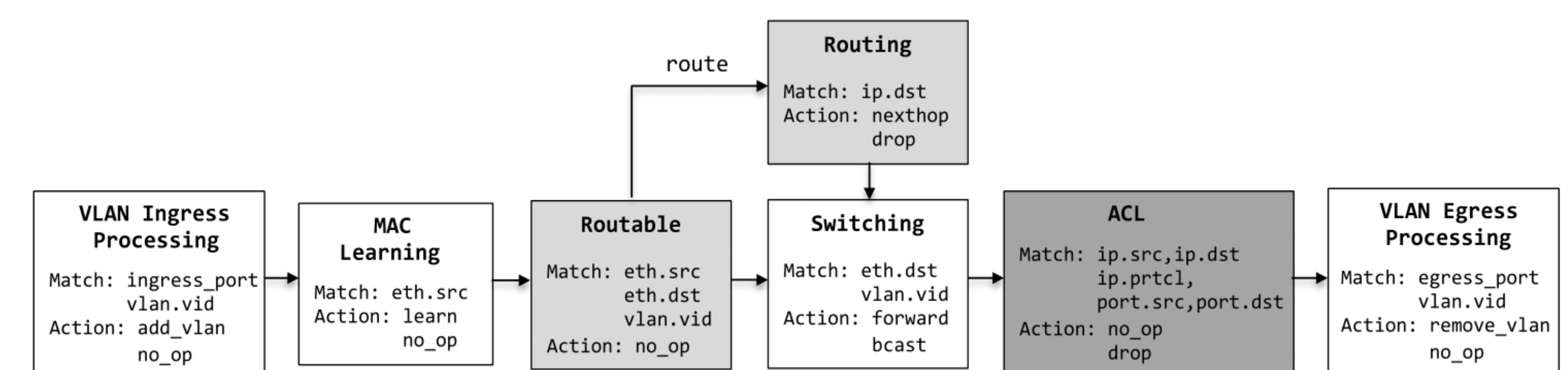
### Evaluation Platform Topology



### Server Specifications.

CPU: Intel Xeon E5-2640 v3 2.6GHz  
 Memory: 32GB RDIMM, 2133 MT/s, Dual Rank  
 Hard Disk: 1TB 7.2K RPM NLSAS 6Gbps  
 NICs: Intel X710 DP/QP DA SFP+ Cards

### L2L3 ACL Benchmark Application



## 4. Results

### Complexity Analysis

	LOC	Methods	Method Size
Native OVS	14,535	106	137.13
Ovs.p4	341	40	8.53

	Files Changed	Lines Changed
Connection Label OVS	28	411
Connection Label ovs.p4	1	5
TCP Flags OVS	20	370
TCP Flag ovs.p4	1	4
Tunnel OAM Flag OVS	18	170
Tunnel OAM Flag ovs.p4	1	6

### Throughput Comparison

