

# OFBench: Performance test suite on OpenFlow Switches

Author: Chen-You Wang  
Advisor: Dr. Ying-Dar Lin  
Affiliation: NCTU  
Date: 2016/06/03

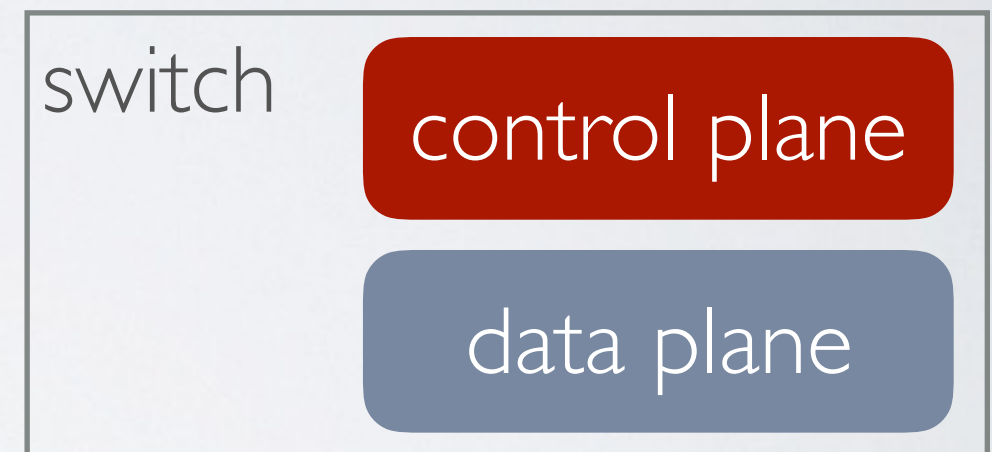
# Outline

- Motivation
- Background
  - OpenFlow
  - OpenFlow Instructions
  - OpenFlow Performance Parameters
- Issues
- Survey
  - Traditional Data Plane
  - OpenFlow Data Plane
  - Bidirectional
- Problem statement
- Approach
  - Mirror-in-progressing
  - Calculated traffic
  - Masked entry
- Implementation
- Experiment and results
- Conclusion
- Future work
- Reference

# Motivation

Open Network Foundation (ONF) [1]

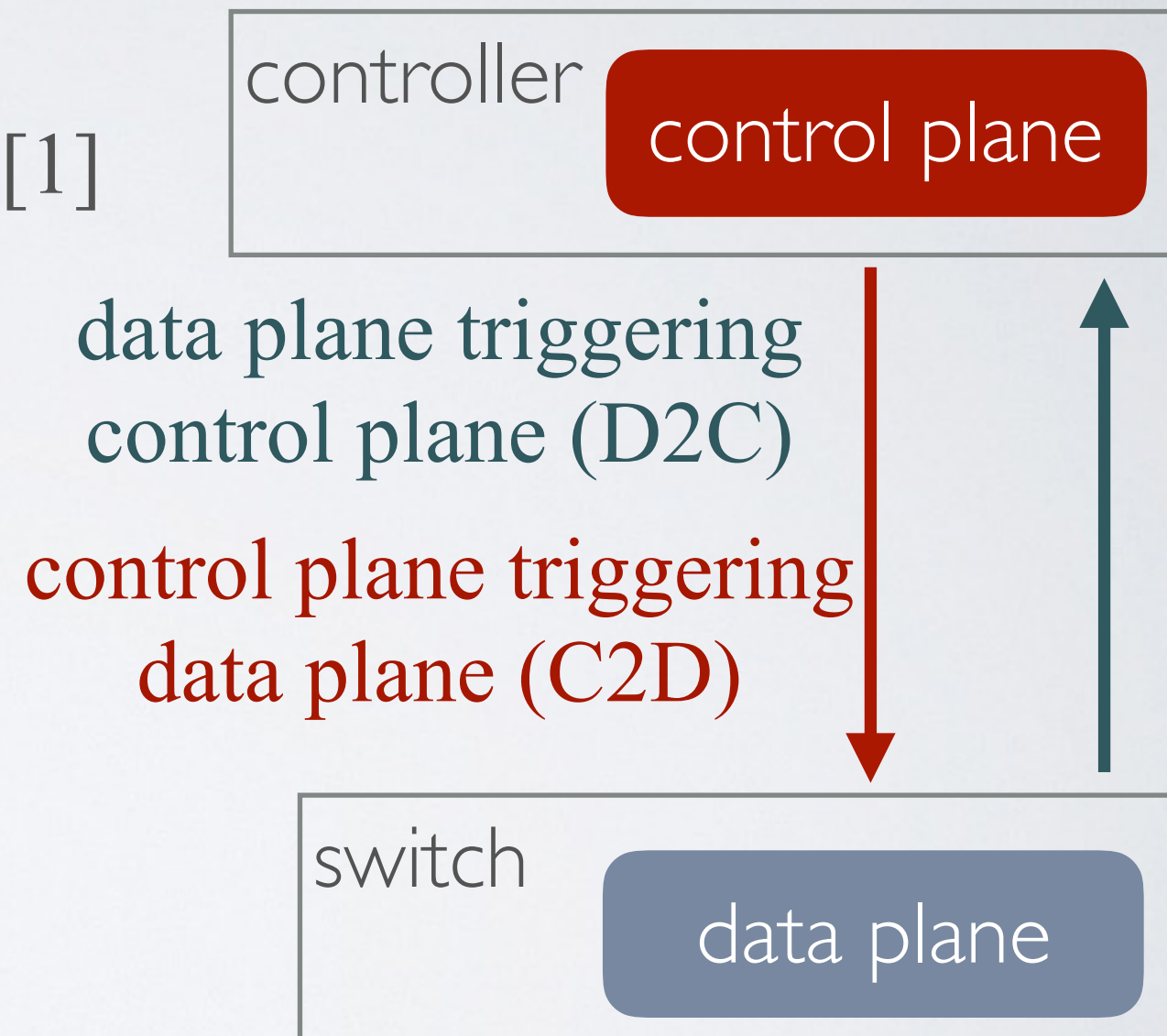
- OpenFlow[2] switch
  - Conformance: OFTest[3], Ryu certification[4]
  - Performance



# Motivation

Open Network Foundation (ONF) [1]

- OpenFlow[2] switch
- Conformance: OFTest[3], Ryu certification[4]
- Performance





# Background - OpenFlow

Open Network Foundation (ONF) [1]

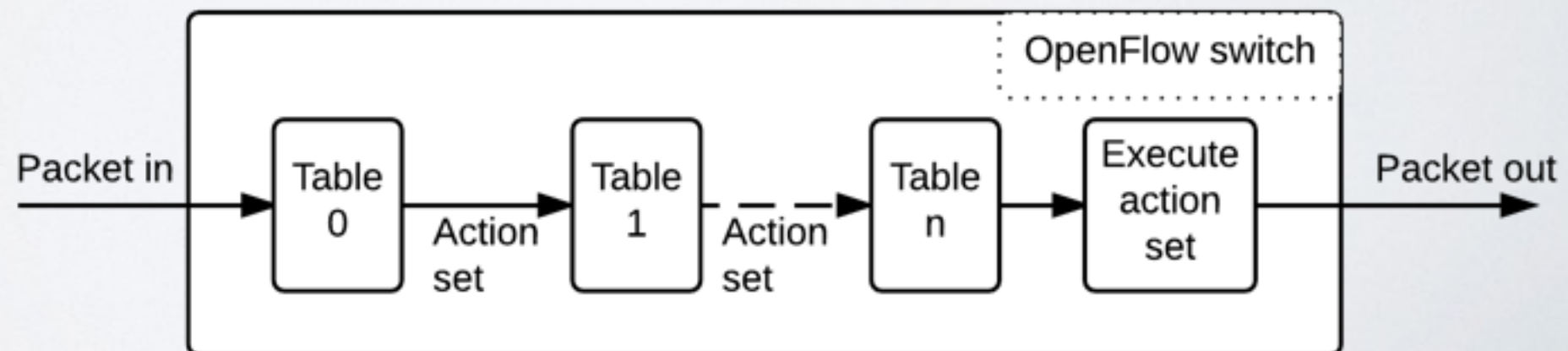
- Standard communications interface.
- Separate control plane and data plane.
- Flow entry

Attributes	Matches	Instructions
priority=1, timeout=5	in_port=1,eth_src,ip_src	write_actions=output:2

# Background - OpenFlow Instructions[2]

Instructions list order by execution order:

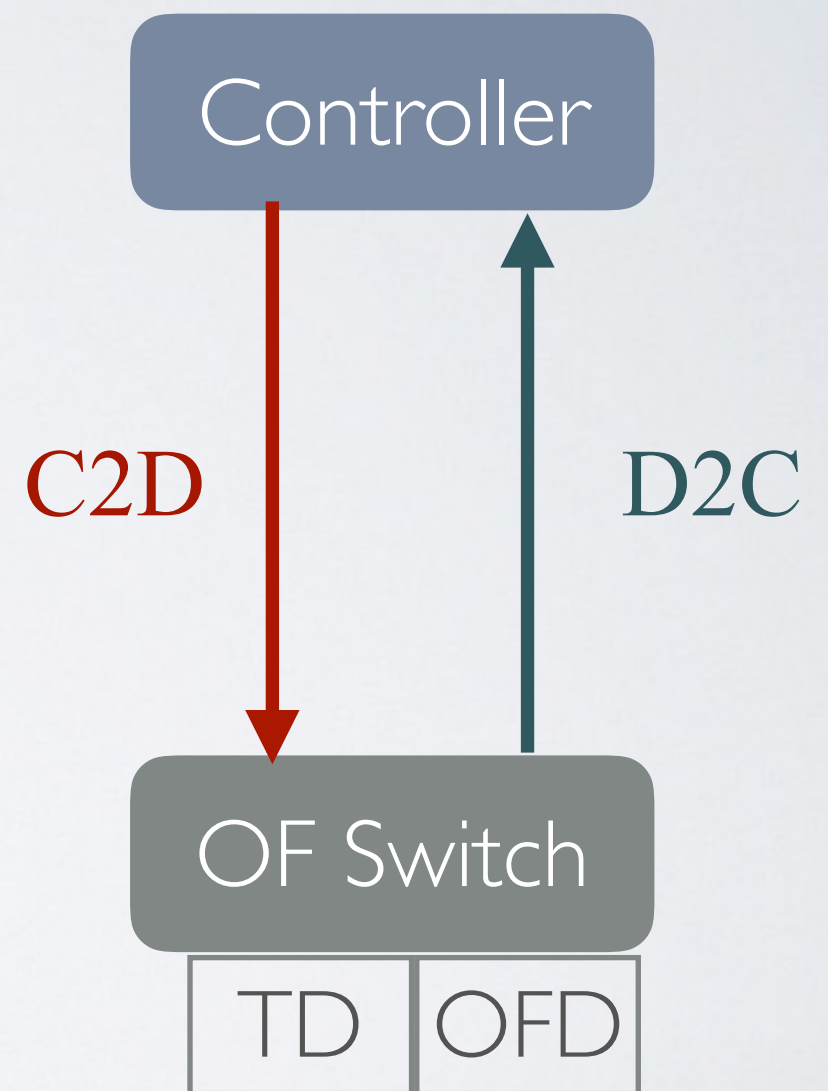
1. Meter
2. Apply-Actions
  - Applies the specific action(s) immediately without change the action set
3. Clear-Actions
4. Write-Actions
5. Write-Metadata
6. Goto-Table



# Background - OpenFlow

## Performance Parameters (1/3)

- Control plane triggering data plane (C2D)
- Data plane triggering control plane (D2C)
- Data plane
  - Traditional (TD)
  - OpenFlow (OFD)



# Background - OpenFlow

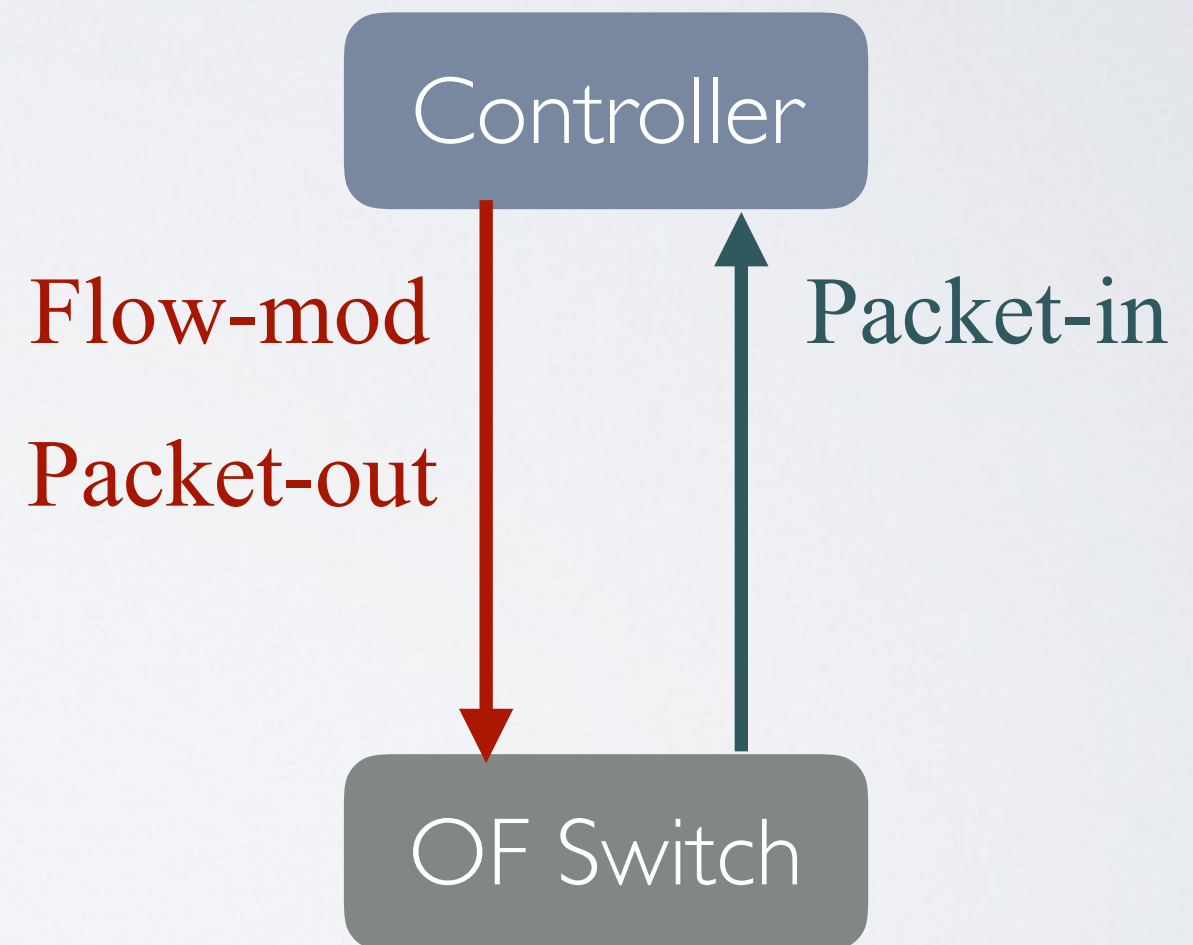
## Performance Parameters (2/3)

C2D

- Flow-mod
- Packet-out

D2C

- Packet-in





# Background - OpenFlow

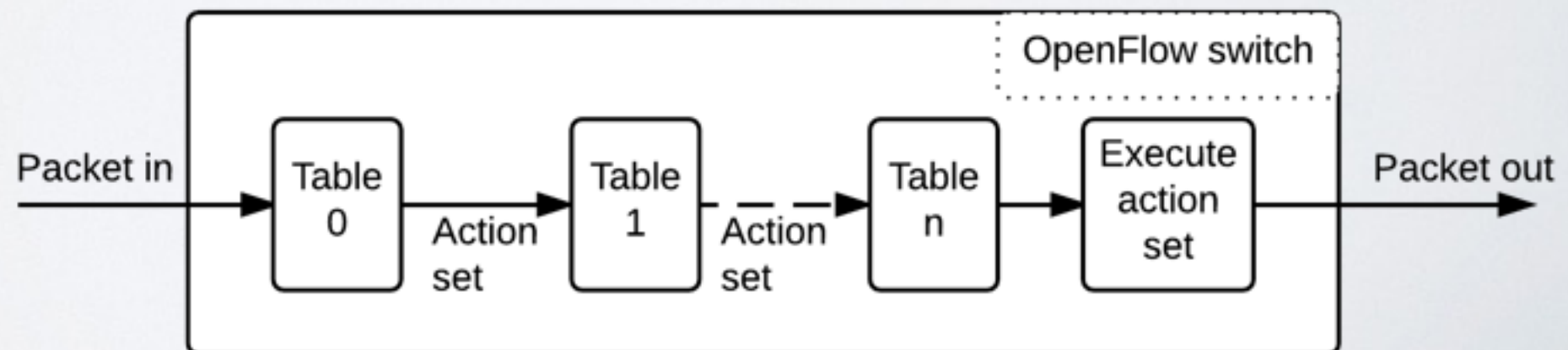
## Performance Parameters (3/3)

TD

- Latency
- Loss
- Buffer size
- Throughput
- Jitter

OFD

- Table lookup
- Multiple table pipeline
- Action
- Timeout
- Table size



# Issues

- How to test the OFD, D2C, C2D parameters.
- Black-box test for some internal parameter.

# Survey - Traditional Data Plane

Parameter	Bianco [7]	Emmerich [8]	Gelberger [9]	Jarschel [10]
Latency	✓	✓	✓	N/A
Loss	✓	N/A	N/A	N/A
Buffer size	N/A	N/A	N/A	✓
Throughput	✓	✓	✓	✓
Jitter	N/A	N/A	✓	N/A

# Survey - OpenFlow Data Plane

Type	Parameter	Spirent [5]	OFLOPS [6]	Bianco [7]	Ours
OFD	Table Lookup	Concept	N/A	Constant	N/A
	Table Pipeline	Concept	N/A	N/A	Time, pipeline gain
	OpenFlow Action	N/A	End-to-end	N/A	Exactly action time
	Timeout	Concept	N/A	N/A	Accuracy
	Table Size	Concept	N/A	N/A	N/A



# Survey - Bidirectional

Type	Parameter	Spirent [5]	OFLOPS [6]	Bianco [7]	Handfield [12]	Ours
D2C	Packet-in rate	Concept	N/A	N/A	N/A	Buffer size
C2D	Packet-out rate	Concept	N/A	N/A	N/A	Buffer size
	Latency of Flow-mod	N/A	Traffic validation	N/A	Impact factors	N/A

# Problem Statement - Notations(1/2)

Category	Notation	Description
Entity	$c$	The controller
	$dut$	The switch under test.
	$N$	The number of tables for $dut$ .
	$H = \{h_n \mid n \geq 2\}$	The set of hosts.
	$CAP = \{cap_c, cap_n \mid n \geq 2\}$	The set of link capacities. $cap_c / cap_n$ is link capacity between $c / h_n$ and $dut$ .
C2D	$CD = POR$	The parameter for C2D. <b>POR</b> means the throughput of packet-in operation in $dut$ .
D2C	$DC = PIR$	The parameter for D2C. <b>PIR</b> means the throughput of packet-in operation in $dut$ .
OFD	$T_{action-set}$	The time of action set execution in $dut$ .
	$T_{table-pipeline}$	The time of table pipeline in $dut$ .
	$buf$	The size of buffer in $dut$ .

# Problem Statement - Notations(2/2)

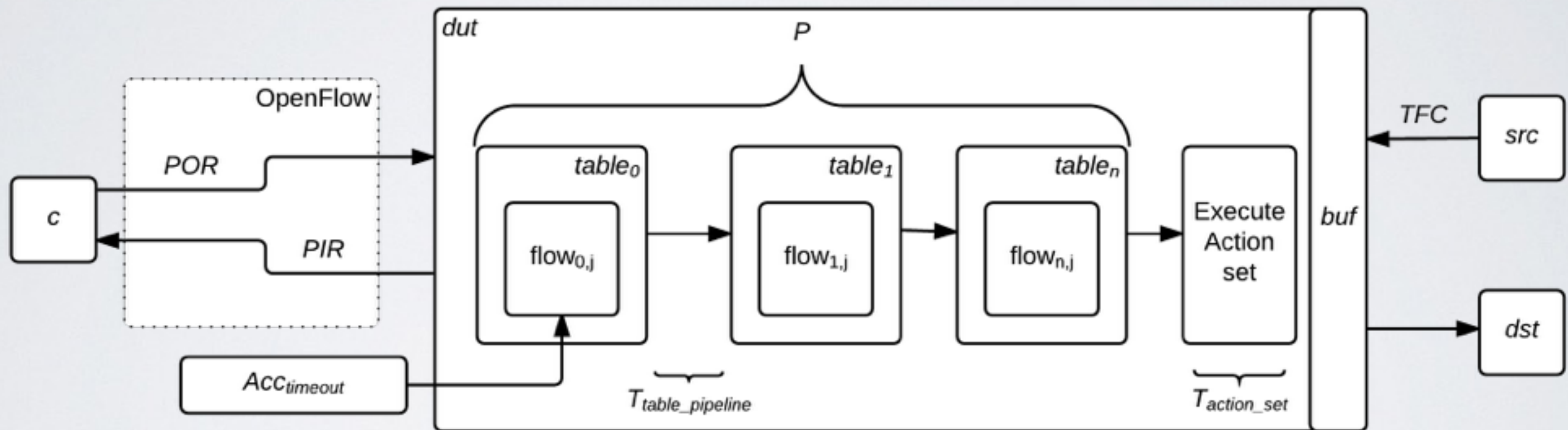
Category	Notation	Description
OFD	$Acc_{hard-timeout}, Acc_{idle-timeout}$	The accuracies of hard-timeout and idle-timeout in <i>dut</i> .
	$P$	The gain of table pipeline in <i>dut</i> .
	$D_{openflow}$	$D_{openflow} = \{T_{action\_set}, T_{table\_pipeline}, Acc_{timeout}, P\}$ The set of parameters for OpenFlow data plane triggering data plane.
Process	$TB = \{table_i \mid 0 < i \leq N\}$	The set of flow tables in <i>dut</i> .
	$F = \{flow_{i,j} \mid 0 < i \leq N, j > 0\}$	The set of flow entries. $flow_{i,j}$ mean the flow entry in $table_i$
	$TFC = \{tfc_y \mid y > 0\}$	The set of traffics which be sent from <i>src</i> to <i>dst</i> . $src \in H, dst \in H. tfc_y = \{pkt_z \mid z > 0\}$
	$T_{table\_lookup}$	The time of looking up the flow table in <i>dut</i>
	$T_{apply\_action}$	The time of executing Apply-Action in <i>dut</i> .
	$T_{idle-timeout}, T_{hard-timeout}$	The timeout value for idle/hard timeout flow entry.
	$T_{idle-expired}, T_{hard-expired}$	The arrival time of flow-removed message for idle/hard timeout flow entry at <i>c</i>
	$T_{idle-duration}, T_{hard-duration}$	The duration of flow-removed message for idle/hard timeout flow entry.

# Problem Statement

- Given:
  - the controller  $c$
  - the set of hosts  $H$
  - the switch under test  $dut$
  - the number  $N$  of tables for  $dut$
  - the set of link capacities  $CAP$
- Objectives:
  - determine  $F, TFC$ 
    - most accurate  $CD, DC, D_{openflow}$
- Constraints:
  - $dut$  not be modified



# Problem Statement - Example



# Approach - Overview

Category	Test Case	Method	Parameters	Output
Mirror-in-processing	Action set	Mirror-first-then-action	OpenFlow action	$T_{action-set}$
	Pipeline time	Mirror-first-then-pipeline	table pipeline	$T_{table\_pipeline}$
Calculated traffic	Buffer size	Burst-until-loss	buffer size, packet-in rate, packet-out rate	$buf, PIR, POR$
	Pipeline gain	Back-to-back-traffic	table pipeline	$P$
Masked entry	Timeout accuracy	Idle-timout-derived-by-hard-timeout	timeout	$Acc_{hard-timeout}, Acc_{idle-timeout}$

# Approach - Action set (1/5)

*Mirror-first-then-action*

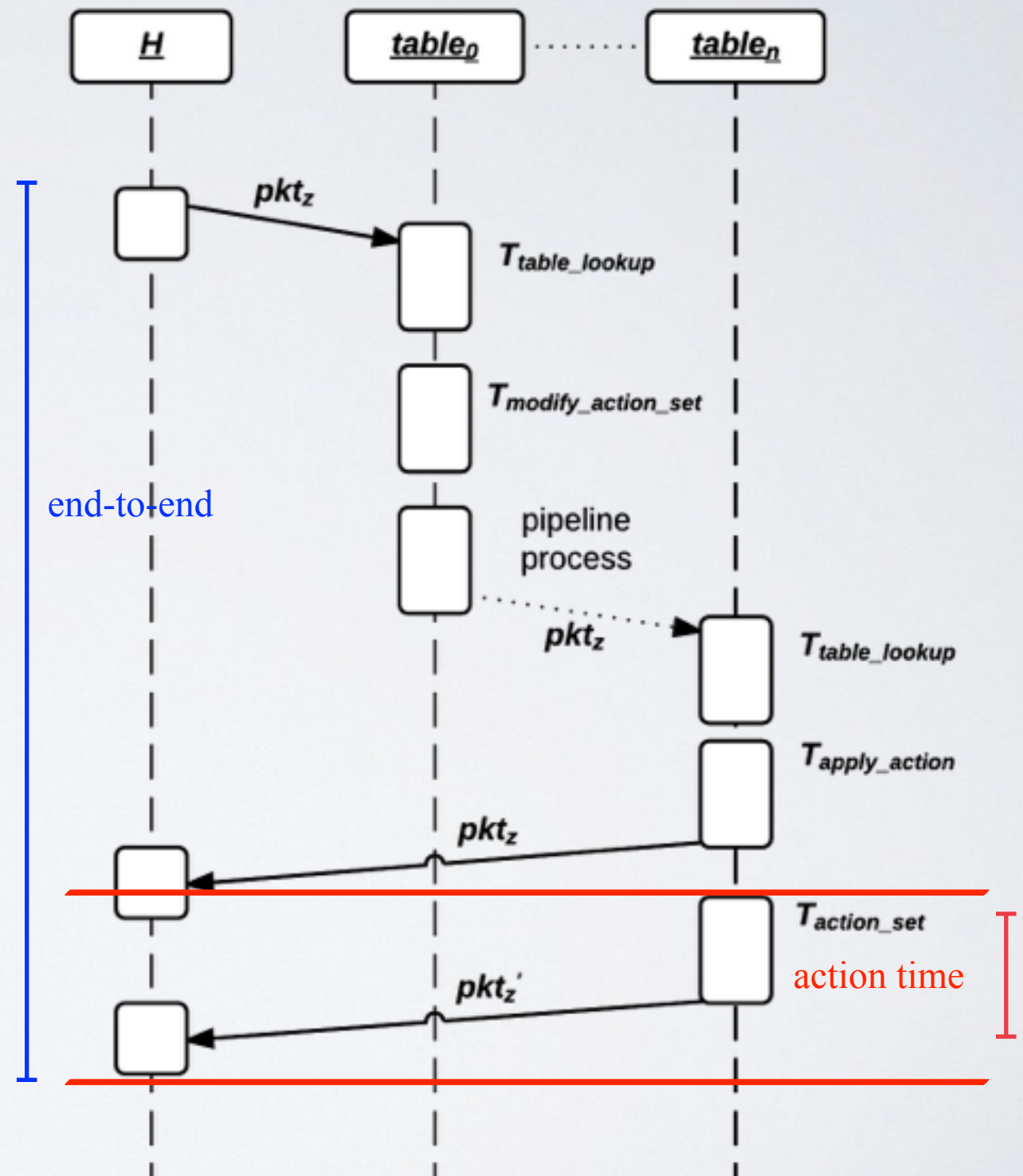
Input:  $tfc_y$ ,  $F$

Output:  $T_{action-set}$

$\alpha$  = time of  $pkt_z$  arrival

$\beta$  = time of  $pkt_z'$  arrival

$$T_{action-set} = \beta - \alpha$$



# Approach - Pipeline time(2/5)

## *Mirror-first-then-pipeline*

Input:  $tfc_y$ ,  $F$

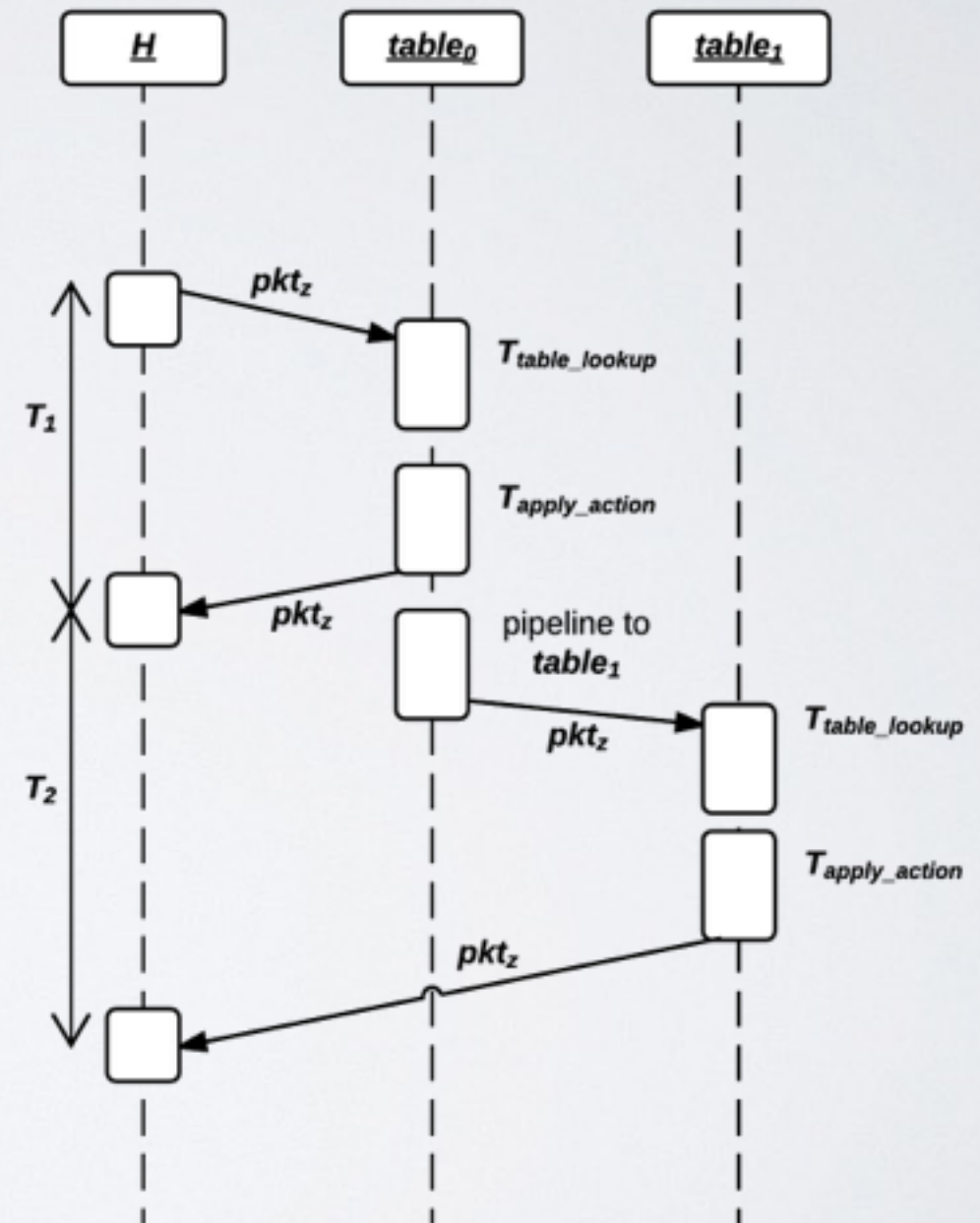
Output:  $T_{table-pipeline}$

$$RTT = (pkt_z / cap_n) * 2$$

$$T_1 = T_{table\_lookup} + T_{apply\_action} + RTT$$

$$T_2 = T_{table-pipeline} + T_{table\_lookup} + T_{apply\_action} + RTT/2$$

$$T_{table-pipeline} = T_2 - T_1 + RTT/2$$





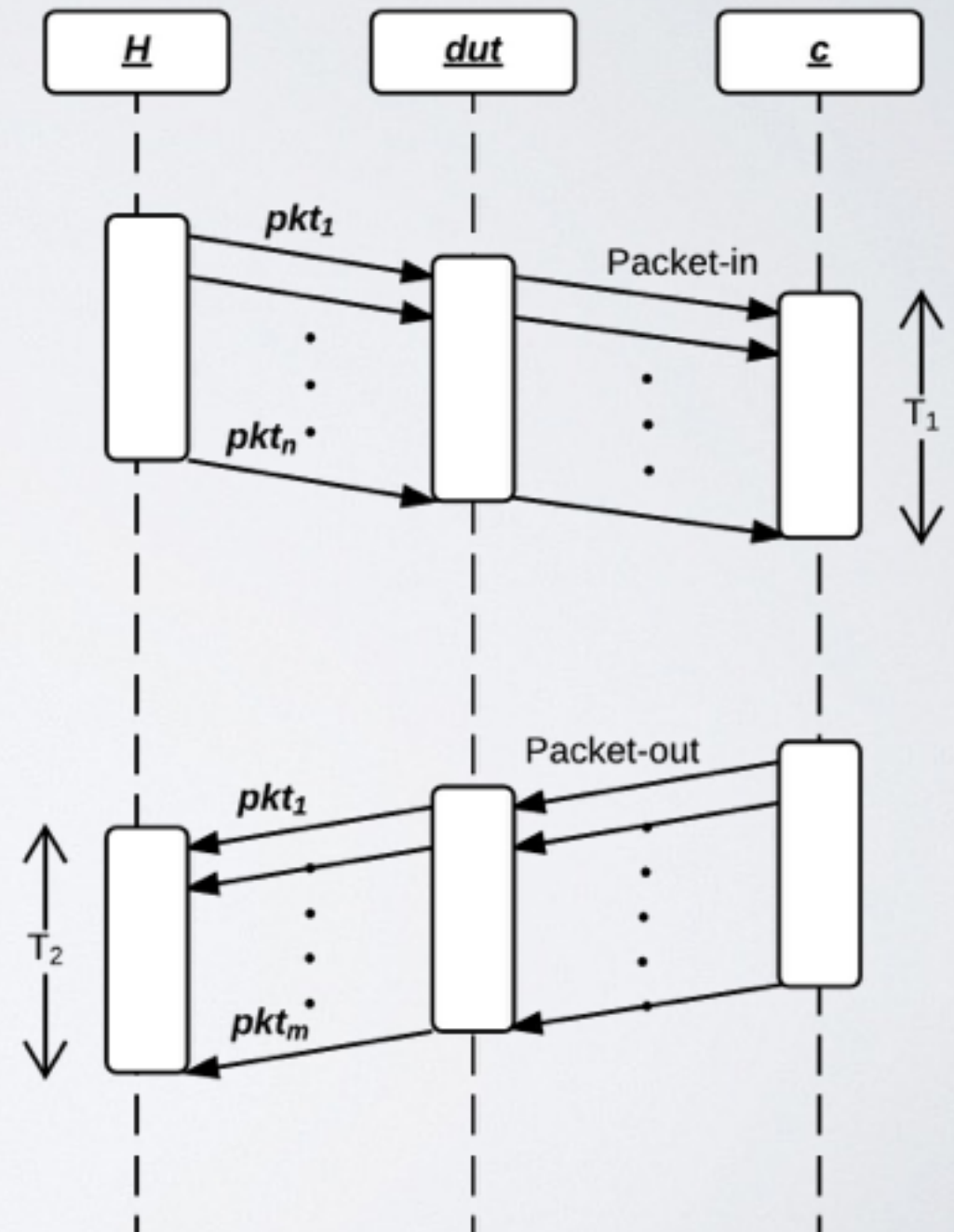
# Approach - Buffer size (3/5)

## *Burst-until-loss*

Input:  $tfc_y$ ,  $N$ ,  $T$

Output:  $buf$ ,  $PIR$ ,  $POR$

- $m < n$
- $buf = m * pkt$
- $PIR = n / T_1$
- $POR = m / T_2$



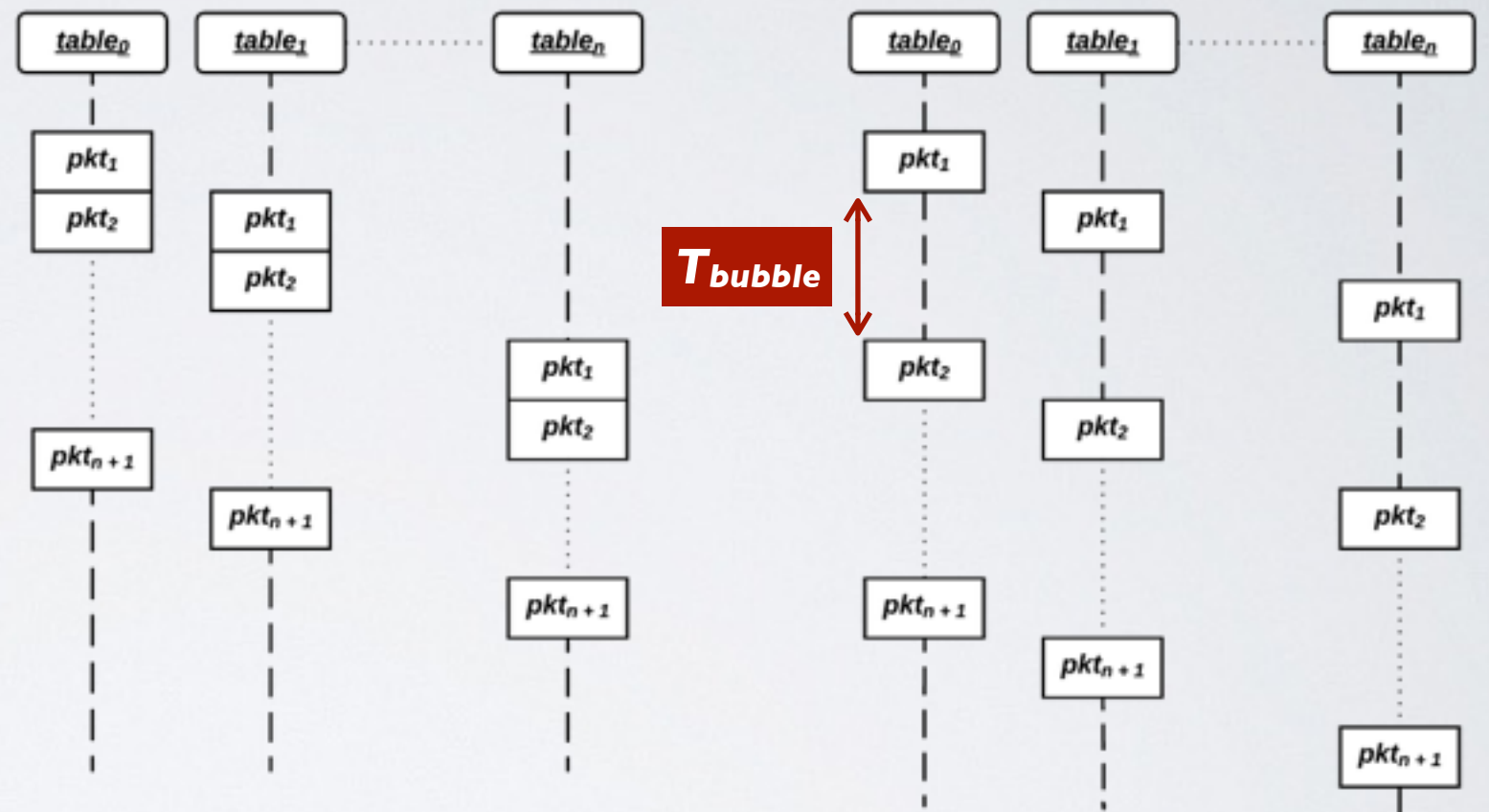
# Approach - Pipeline gain (4/5)

*Back-to-back-traffic*

Input:  $tfc_y$ ,  $F$ ,  $n$ ,  $z$

Output:  $P$

$$P = T_{pkt} / (T_{bubble} + T_{pkt})$$



(a) Full pipeline

(b) Without pipeline

# Approach - Pipeline gain (4/5)

*Back-to-back-traffic*

Input:  $tfc_y, F, n, z$

Output:  $P$

$$P = T_{pkt} / (T_{bubble} + T_{pkt})$$

$$T_{pkt} \cong T_{init} / (n + 1)$$

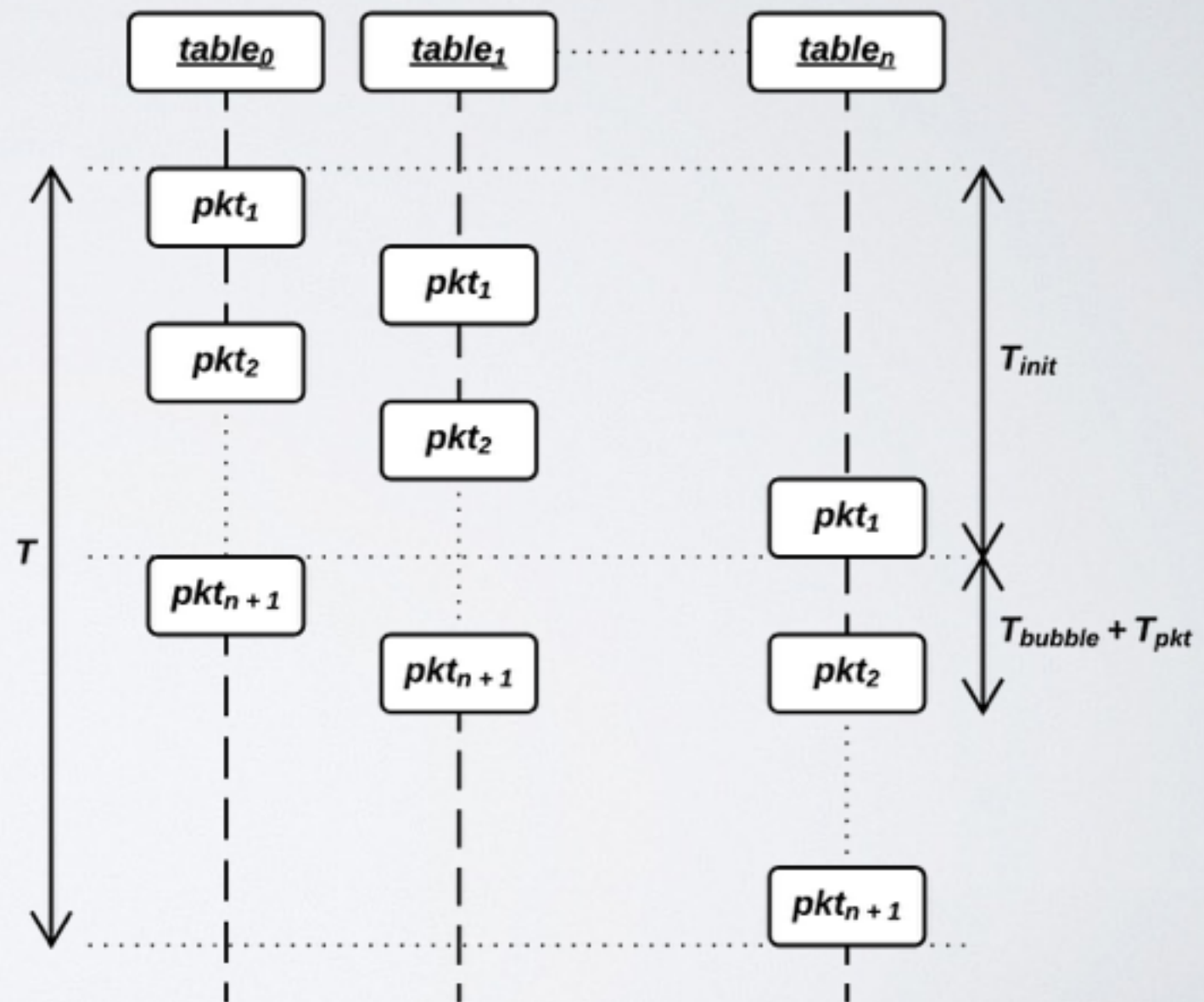
$$T_{bubble} + T_{pkt} \cong (T - T_{init}) / n$$

$\alpha_z$  = time of sending  $pkt_z$

$\beta_z$  = time of  $pkt_z$  arrival

$$T = \beta_{n+1} - \alpha_1$$

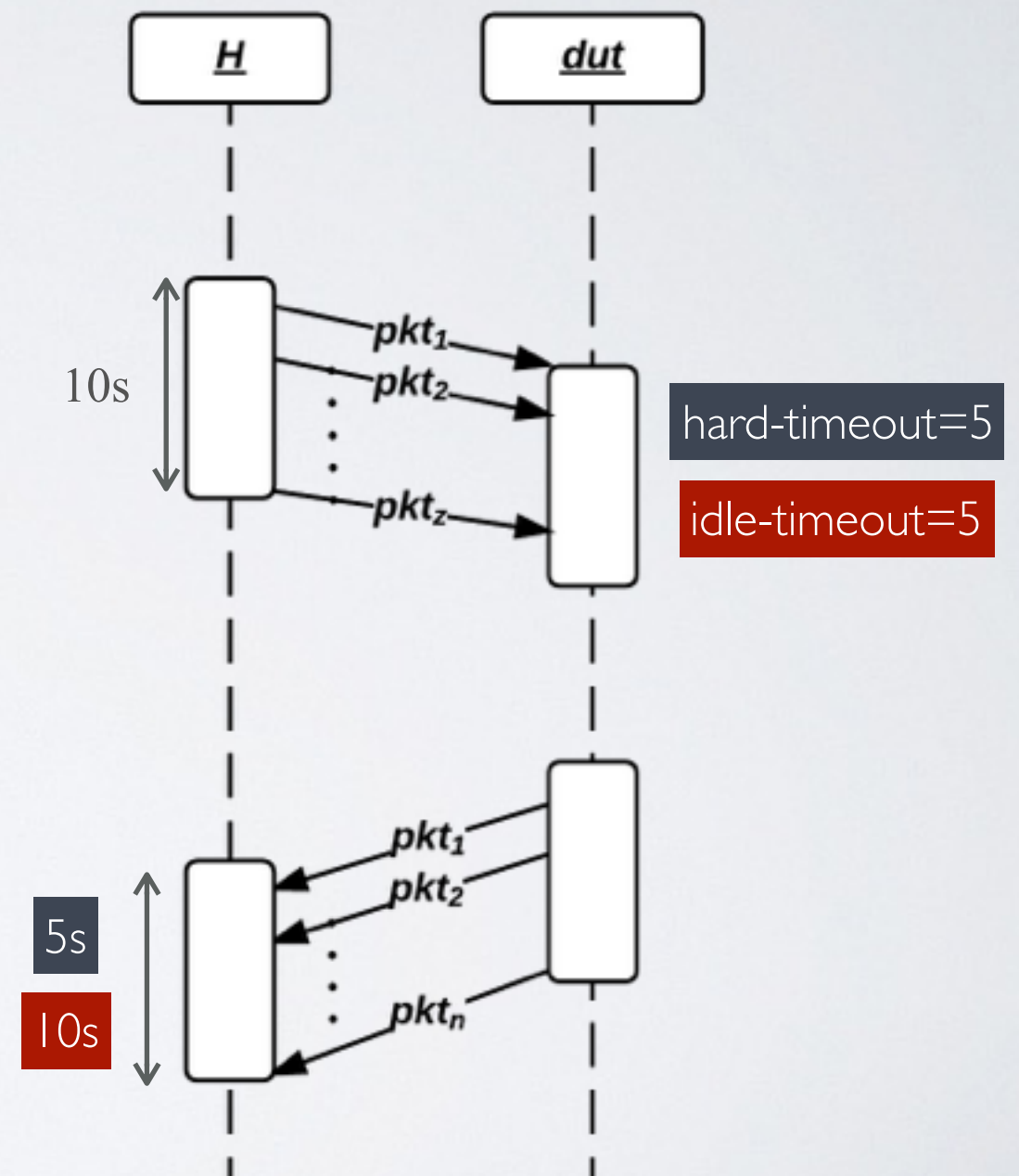
$$T_{init} = \beta_1 - \alpha_1$$



# Approach - Timeout accuracy (5/5)

idle-timeout timer resetting

*Idle-timeout-derived-by-hard-timeout*

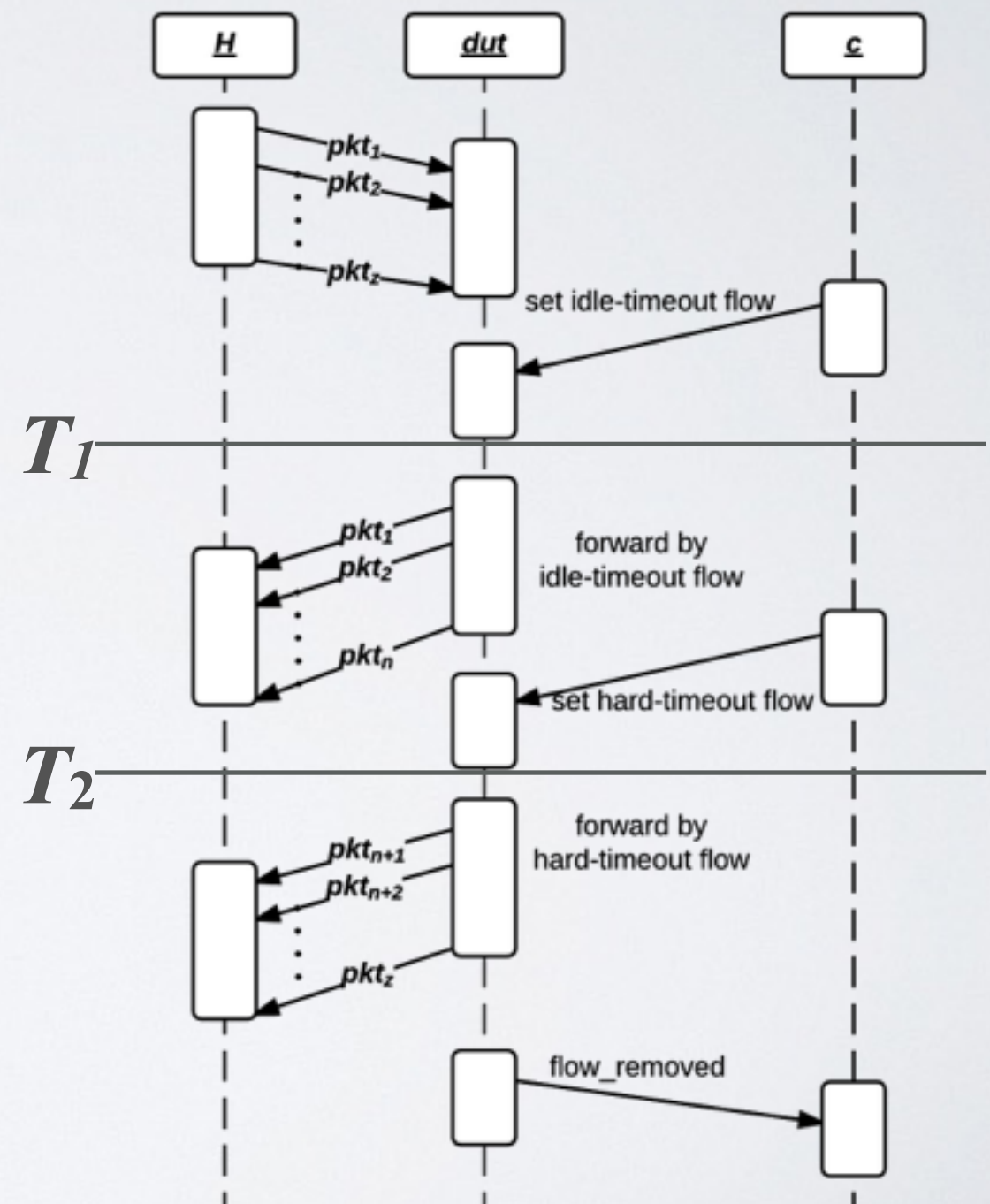




# Approach - Timeout accuracy (5/5)

## Timer synchronization

Time	F
$T_1$	idle-timeout=10, idle-age=0, priority=1, tcp, actions=output:1
$T_2$	idle-timeout=10, <b>idle-age=0</b> , priority=1, tcp, actions=output:1
	hard-timeout=10, hard-age=0, <b>priority=2</b> , tcp, actions=output:2
$T_2 + 1$	idle-timeout=10, <b>idle-age=1</b> , priority=1, tcp, actions=output:1
	hard-timeout=10, <b>hard-age=1</b> , priority=2, tcp, actions=output:2



# Approach - Timeout accuracy (5/5)

*Idle-timout-derived-by-hard-timeout*

Input:  $tfc_y, F$

Output:  $Acc_{timeout}$

$$T_{idle-timeout} = T_{hard-timeout}$$

$\alpha$  = time of  $pkt_z$  arrival,  $\beta$  = time of  $pkt_z'$  arrival

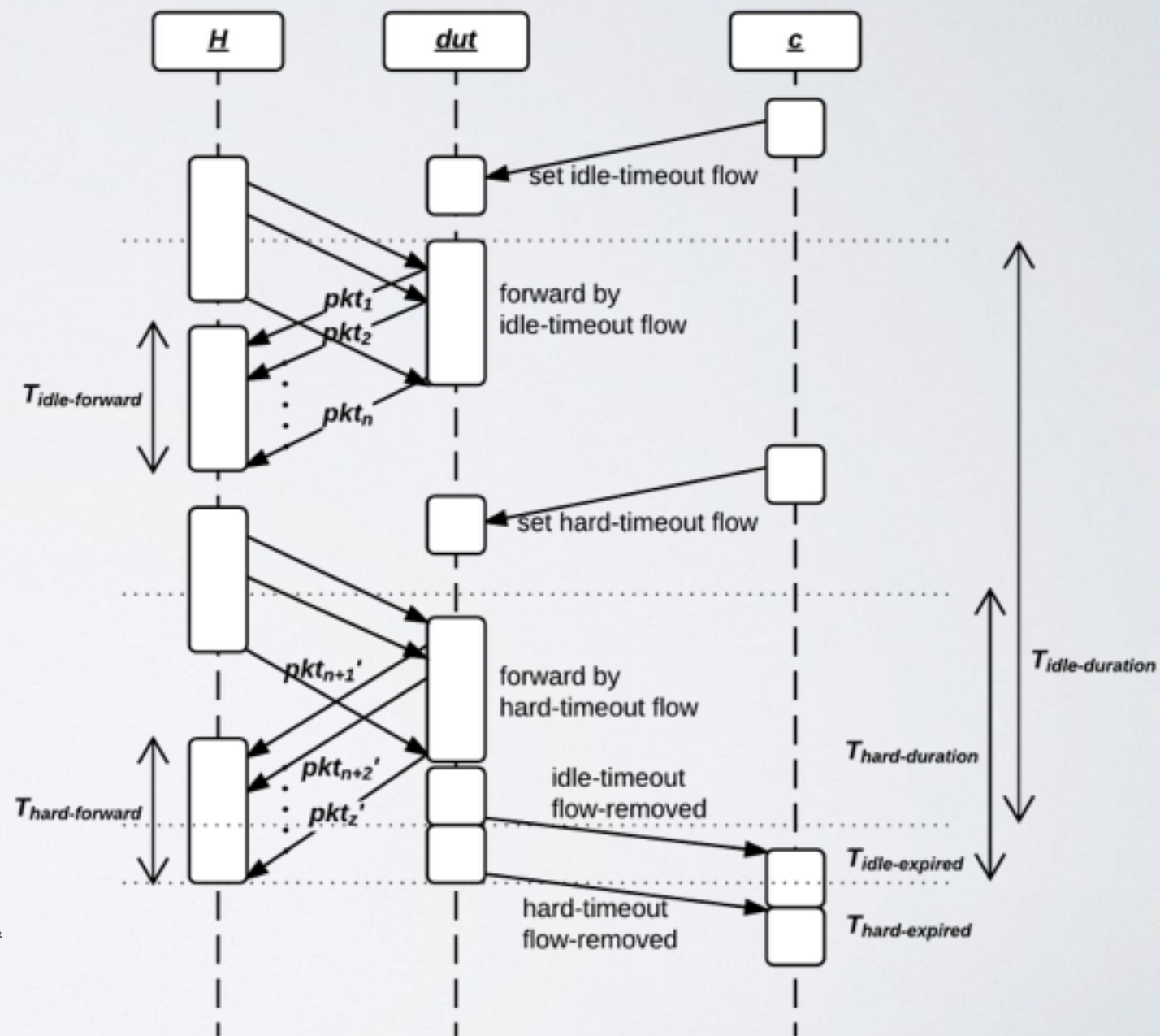
$$T_{idle-forward} = \alpha_n - \alpha_1$$

$$T_{hard-forward} = \beta_z - \beta_{n+1}$$

$$(a) T_{idle-expired} \cong T_{hard-expired}$$

$$Acc_{hard-timeout} = T_{hard-forward} - T_{hard-timeout}$$

$$Acc_{idle-timeout} = T_{idle-duration} - T_{idle-forward} - T_{idle-timeout}$$



(a)  $T_{idle-expired} \cong T_{hard-expired}$

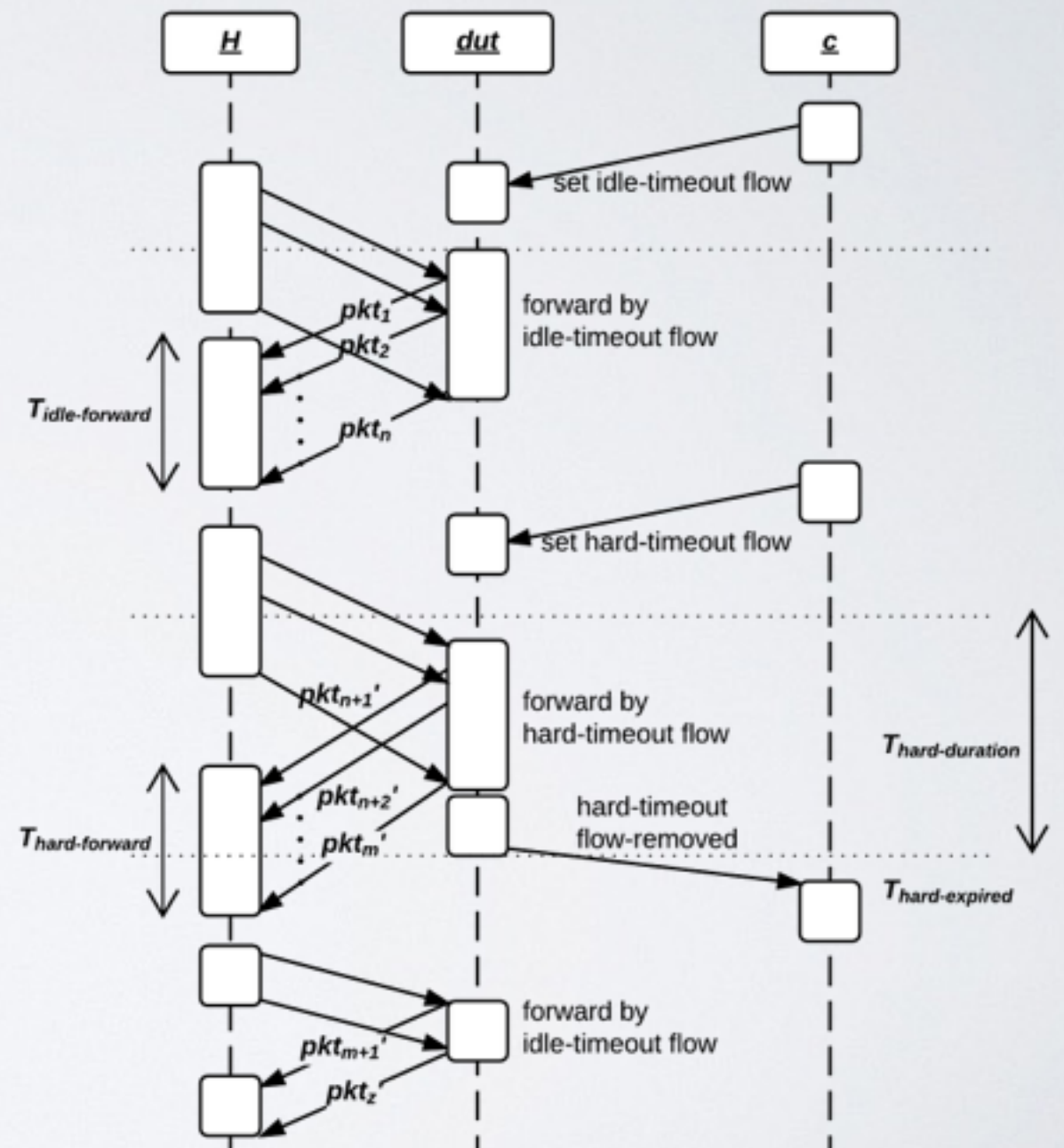
# Approach - Timeout accuracy (5/5)

*Idle-timout-derived-by-hard-timout*

(b)  $T_{idle-expired} > T_{hard-expired}$

Increase  $T_{hard-timeout}$

Reset hard-timeout flow entry



(b)  $T_{idle-expired} > T_{hard-expired}$



# Implementation

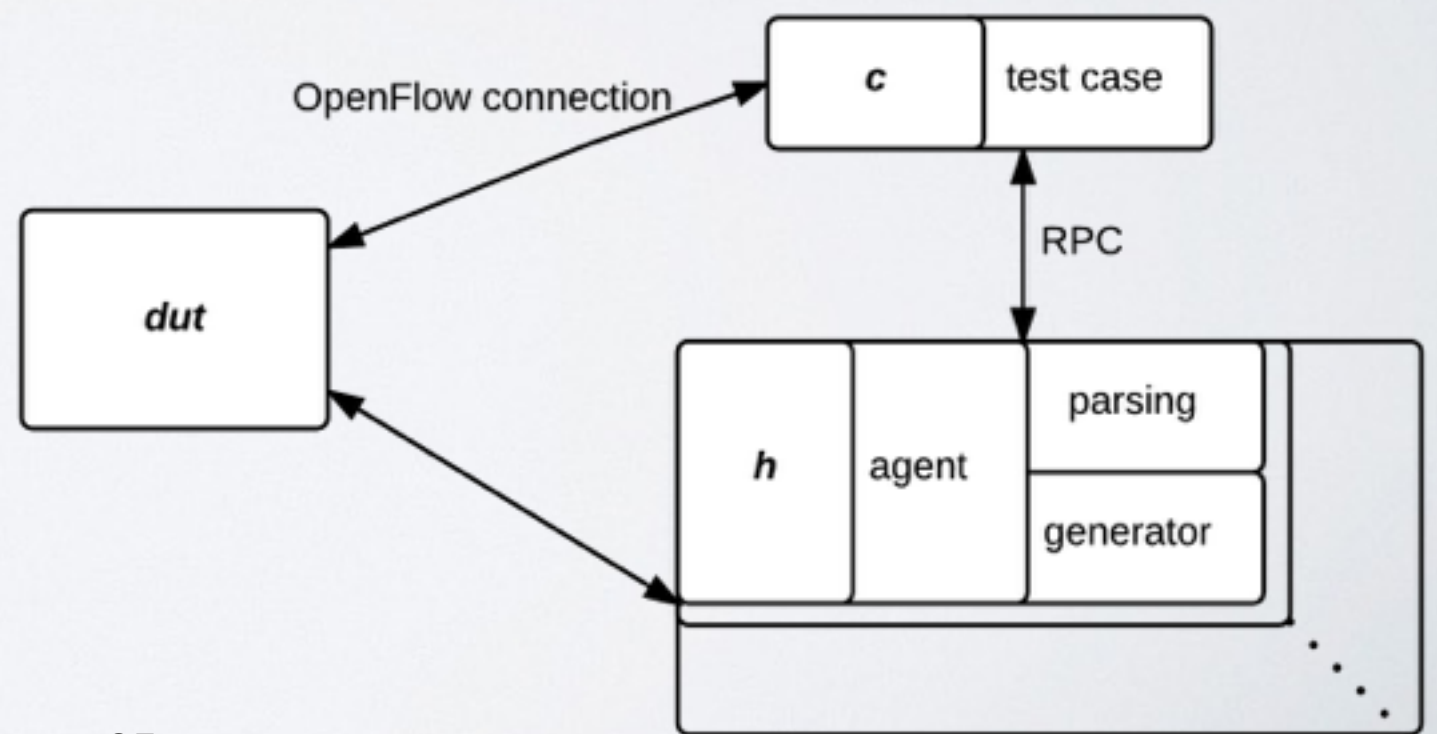
- Tools

- OF controller: Ryu v3.2.1
- Traffic generator: Ostinato v0.7
- Software switch: Open vSwitch v2.3.2

Component	core	clock rate	OS	kernel
Controller	4	3.3 GHz	Ubuntu 14.04	3.13.0-24
Host	4	3.2 GHz	Ubuntu 14.04	3.13.0-24
Switch	2	3.1 GHz	Ubuntu 14.04	3.13.0-24

- OFBench architecture

- test cases
- hosts with agent
  - traffic generator, parsing





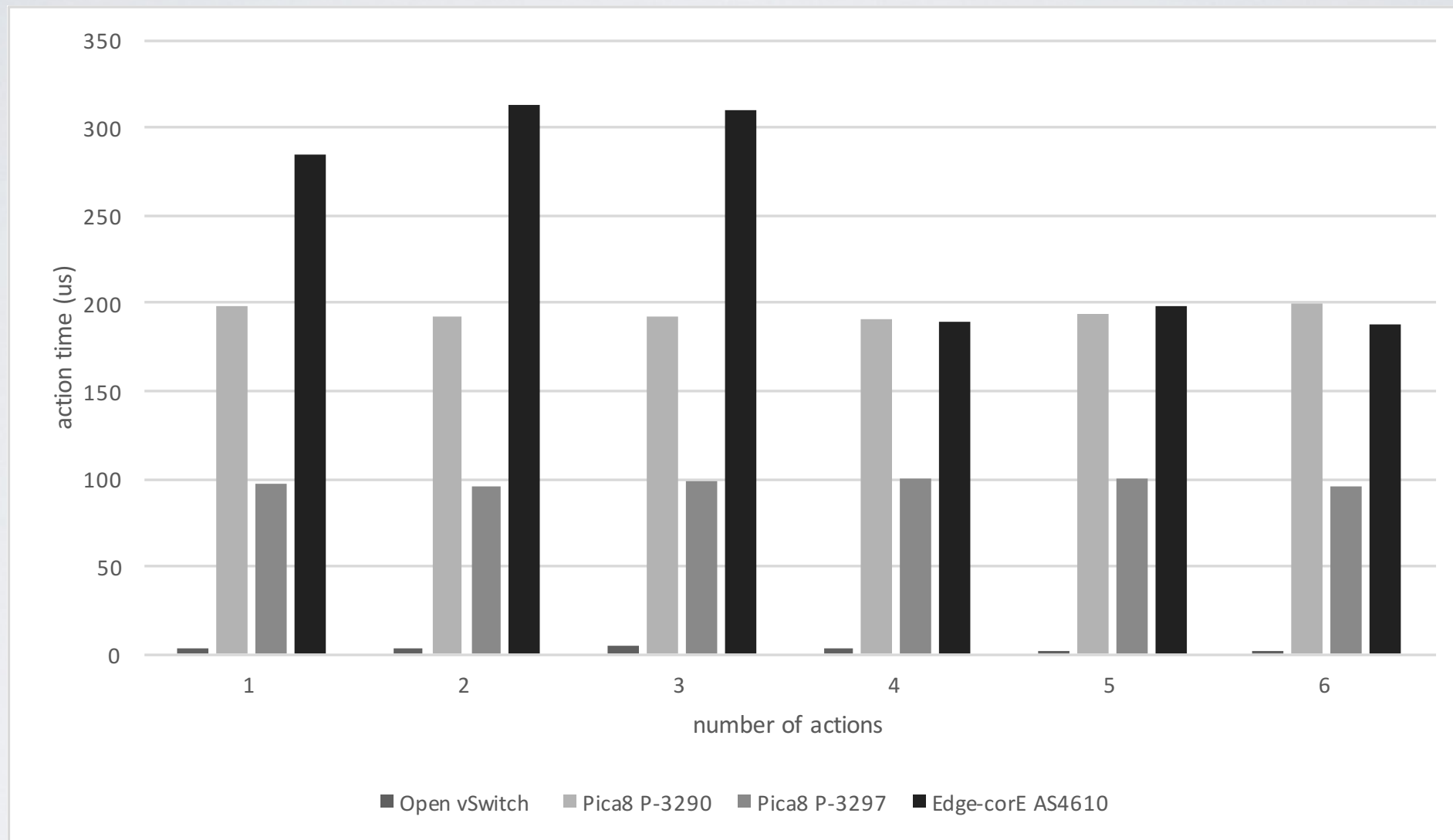
# Experiment

- Testbed
- Configuration
  - *pkt*: frame size, transmission rate
  - switch: hardware, software
- Issues to study
  - Effect of different configuration

# Switch Specifications

Switch	CPU	core	clock rate	Memory	buffer	OS version
Pica8 P-3290	MPC8541	1	1 GHz	512 MB	4 MB	v2.6.1
Pica8 P-3297	P2020	2	1.33 GHz	2 GB	4 MB	v2.6.1
Edge-corE AS4610-30T	ARM Cortex A9	2	1 GHz	2 GB	N/A	v2.6.4
Centec V350	e500v2	1	533 MHz	2 GB	N/A	v3.1(11), 1.alpha

# Results - Action time



frame size: 1024 bytes

packet-per-sec: 1000

duration: 10s

actions: [ip\_src, ip\_dst, eth\_src, eth\_dst, udp\_src, udp\_dst]

# Results - Action time - White box

End-to-end	ours	white box
315 us	2~3 us	0.6~0.8us

- Tool: ftrace
- tracer: function\_graph
- function: set\_ip\_addr

frame size: 1024 bytes

packet-per-sec: 1000

duration: 10s

actions: [ip\_src, ip\_dst]



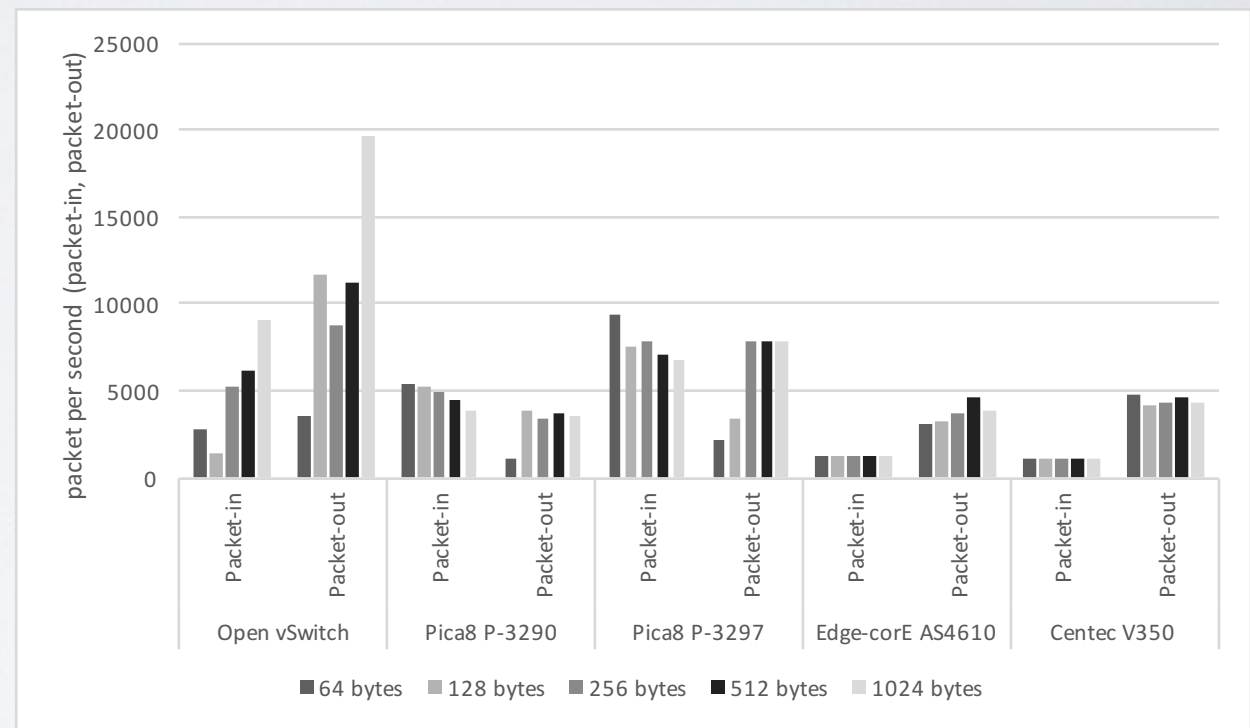
# Result - Buffer size

Buffer size (byte)

frame size	ovs	pica8 P-3290	pica8 P-3297	Edge-core	Centec V350
64	16384	8256	16192	32768	342464
128	N/A	N/A	N/A	N/A	N/A
256	N/A	N/A	N/A	N/A	N/A
512	N/A	N/A	N/A	N/A	N/A
1024	N/A	N/A	N/A	N/A	N/A

TX rate: 1Gbps  
duration: 5~10s

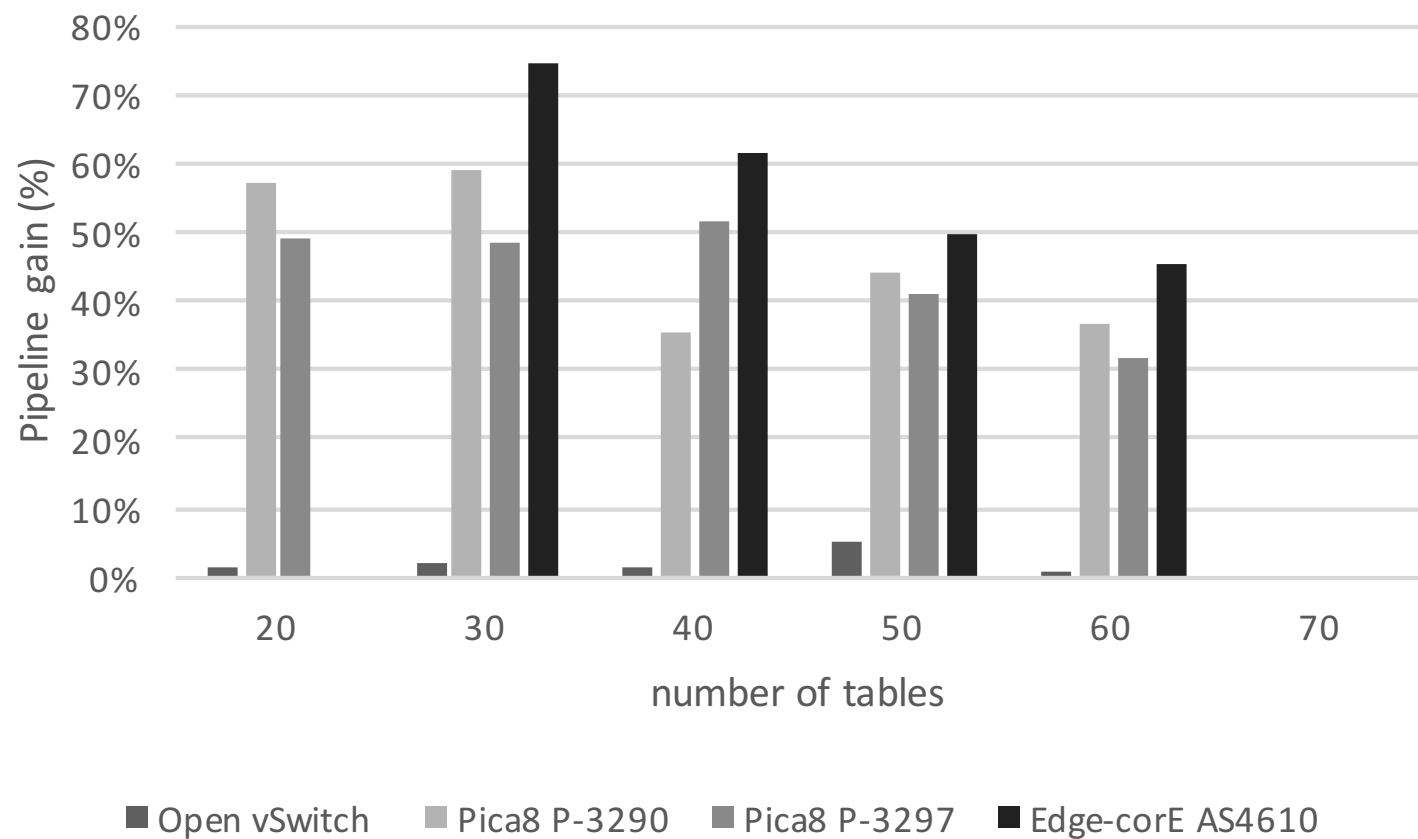
CVE-2016-2074 [15], buffer overflow



# Results - Pipeline Gain

2 ports

48 ports



loading	packet operation time	Pipeline gain
↑	↑	↑

# Result - Timeout Accuracy

Switch	Acc <sub>idle-timeout</sub>	Acc <sub>hard-timeout</sub>
Open vSwitch	2%	0%
Pica8 P-3290	-16.24%	4.7%
Pica8 P-3297	24.64%	3.3%
Edge-corE AS4610	10%	0%
Centec V350	17.64%	1.66%

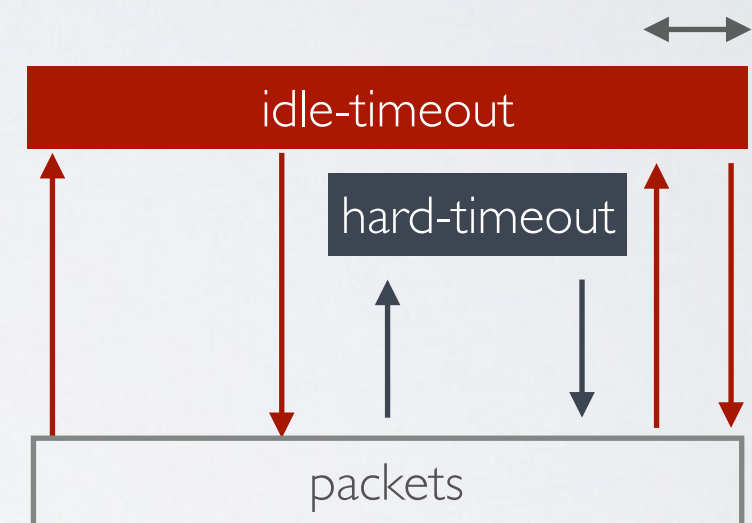
hard-timeout: 3~7

idle-timeout: 5

frame size: 64 byte

packet-per-sec: 10000

Pica8 switches not reset the timer



# Conclusion

## Method evaluation

- 7 test parameters → 6 insights
- The hardware switch have  $\pm 20\%$  deviation for idle timeout.
- The Packet-in rate for hardware switch is approximately 20 times higher than software switch
- The hardware switches reach 40~60% pipeline gain under handling the 1Gbps traffic.

## Switch evaluation

- Burst traffic and packet modification: Pica8 P-3297
- Multi-table handling: Edge-core AS4610

## Issues for switch implementation

- Apply-Action
- Burst Packet-in → crash
- idle-timeout timer not reset properly



# Future Work

- Loading
- Unconsidered variables: Queuing time

# Reference

1. Open Network Foundation. <https://www.opennetworking.org/about/onf-overview>.
2. “OpenFlow Switch Specification,” vol. 3, pp. 1–164, 2013.
3. OFTest. <http://www.projectfloodlight.org/oftest/>.
4. Ryu certification. <https://osrg.github.io/ryu/certification.html>.
5. Spirent, “OpenFlow Performance Testing,” 2015.
6. C. Rotsos, N. Sarrar, S. Uhlig, R. Sherwood, and A. W. Moore, “OFLOPS: An open framework for OpenFlow switch evaluation,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7192 LNCS, pp. 85–95, 2012.
7. A. Bianco, R. Birke, L. Giraudo, and M. Palacin, “OpenFlow switching: Data plane performance,” *IEEE International Conference on Communications*, 2010.
8. P. Emmerich, D. Raumer, F. Wohlfart, and G. Carle, “Performance characteristics of virtual switching,” *2014 IEEE 3rd International Conference on Cloud Networking, CloudNet 2014*, pp. 120–125, 2014.

# Reference

9. A. Gelberger, N. Yemini, and R. Giladi, “Performance analysis of Software-Defined Networking (SDN),” in *Proceedings - IEEE Computer Society’s Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, MASCOTS*, pp. 389–393, 2013.
10. M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll, and P. Tran-Gia, “Modeling and performance evaluation of an OpenFlow architecture,” *2011 23rd International Teletraffic Congress (ITC)*, pp. 1–7, 2011.
11. C. Rotsos, G. Antichi, M. Bruyere, P. Owezarski, and A. W. Moore, “OFLOPS-Turbo: Testing the next-generation OpenFlow switch,” *IEEE International Conference on Communications*, pp. 5571–5576, 2015.
12. R. B. Handfield and K. McCormack, “What You Need to Know About SDN Flow Tables,” *Supply Chain Management Review*, no. September, pp. 29–36, 2015.
13. V. Tanyingyong, M. Hidell, and P. Sjödin, “Improving PC-based OpenFlow switching performance,” *Architectures for Networking and Communications Systems (ANCS), 2010 ACM/IEEE Symposium on*, pp. 8–9, 2010.
14. D. Y. Huang, K. Yocum, and A. C. Snoeren, “High-fidelity switch models for software- defined network emulation,” *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking - HotSDN ’13*, p. 43, 2013.
15. CVE-2016-2074: MPLS buffer overflow vulnerabilities in Open vSwitch. <http://openvswitch.org/pipermail/announce/2016-March/000082.html>.