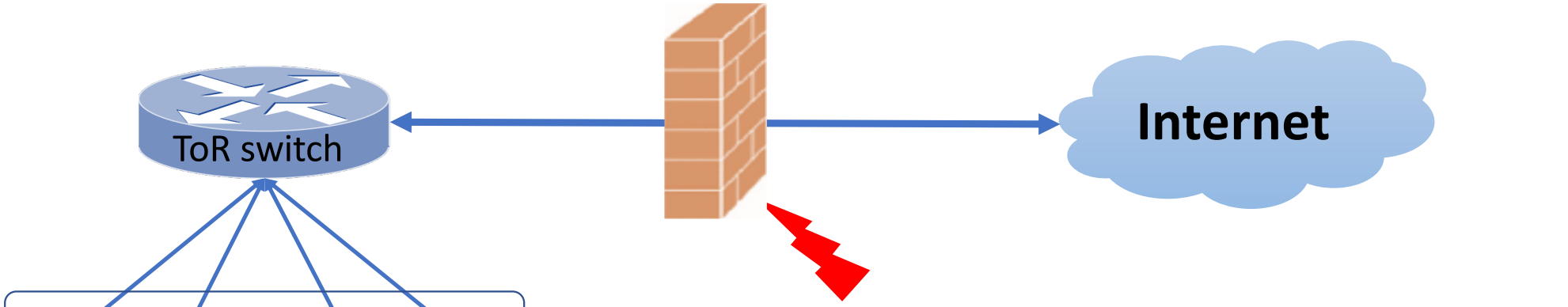
# CoFilter: A High-Performance Switch-Accelerated Stateful Packet Filter for Bare-Metal Servers

**Jiamin Cao**

Ying Liu, Yu Zhou, Chen Sun, Yangyang Wang, Jun Bi

# Stateful packet filter for bare-metal servers




**A rack with Bare Metal Servers (BMS)**

Per-packet inspection by tracking connection states and dropping illegal packets

Bare-metal server (BMS):

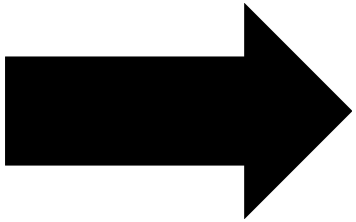
- Excellent single-tenant environment
- Raw processing power
- High reliability

# Status quo: stateful packet filter in bare-metal servers

- Provide realiable security guarantee
- Satisfy the stringent performance requirements of many appliacations

| Stateful packet filter solutions | Performance | Cost |
|---|---|---|
| Dedicated hardware | High performance | **Expensive** |
| Software solutions | **Performance penalty** | Relative cheap |

# CoFilter: A High-Performance Switch-Assisted Stateful Packet Filter for Bare-Metal Servers

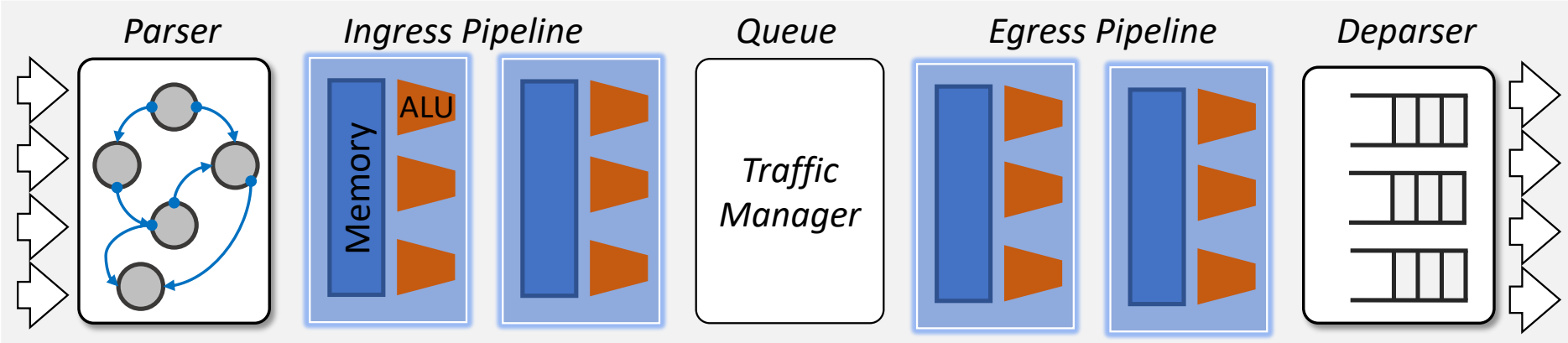



# Programmable switch

*Flexibility* ⇒ ***Implement stateful logic***

*High performance* ⇒ ***Satisfy performance requirements***

*Low cost* ⇒ ***Deployable***



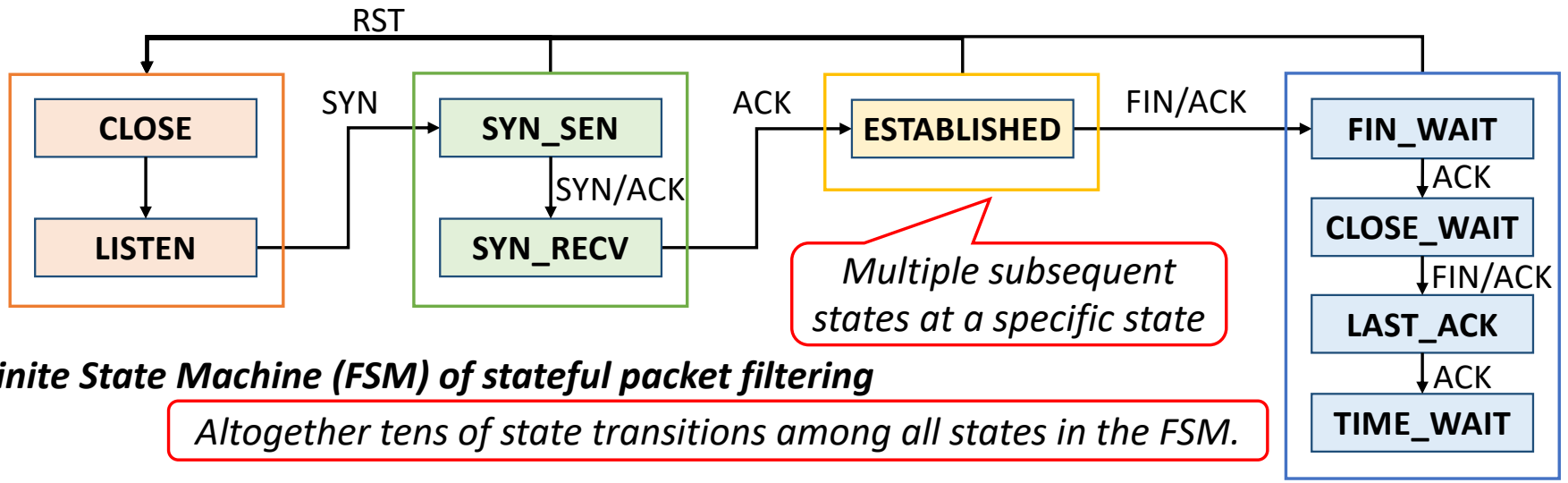

*[RMT@SIGCOMM'13]*

# CoFilter: A High-Performance Switch-Assisted Stateful Packet Filter for Bare-Metal Servers

**C1: Complexity of stateful packet filtering logic vs. limited programmability of programmable ASICs**

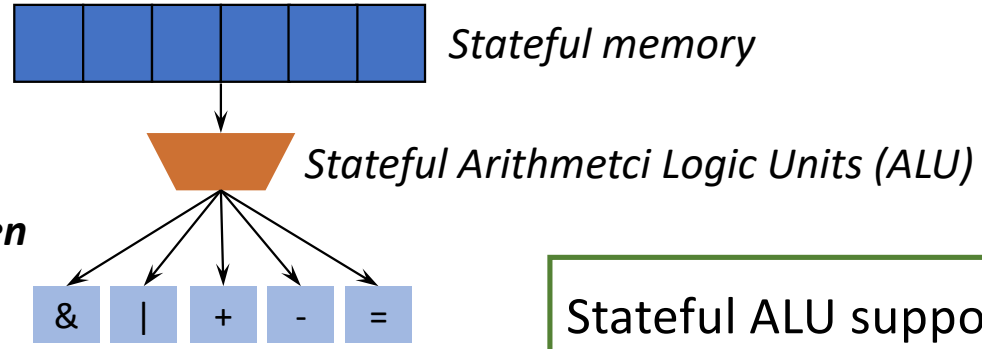
**C2: Scalability requirement for tracking massive connections vs. limited memory space in switching ASICs**

C1

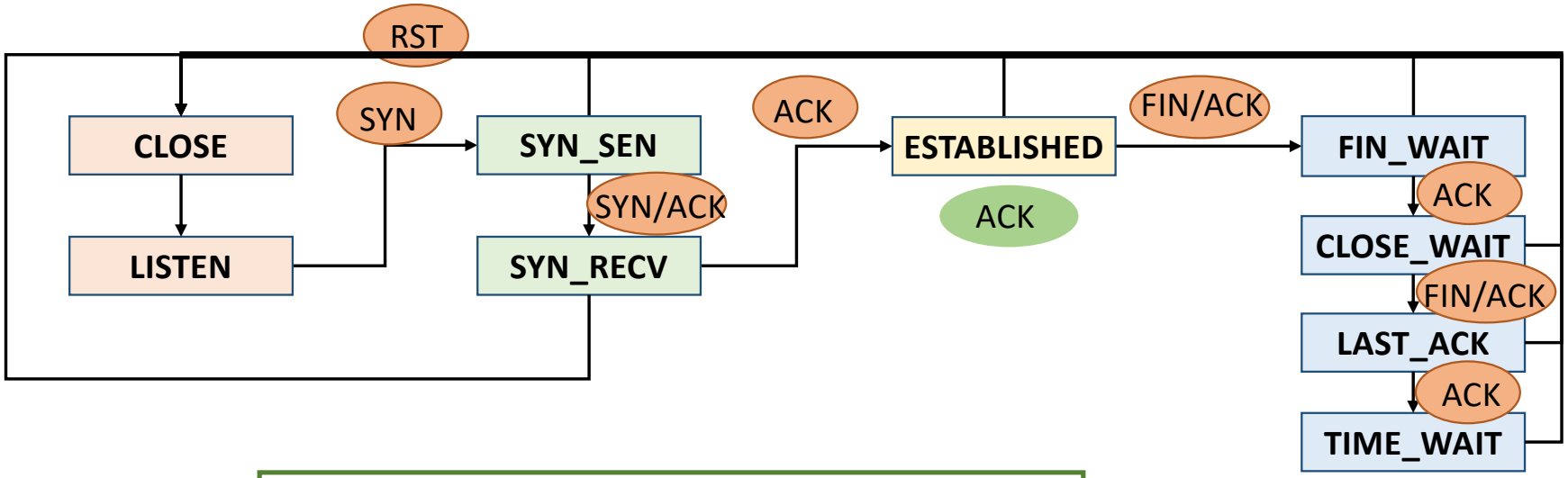


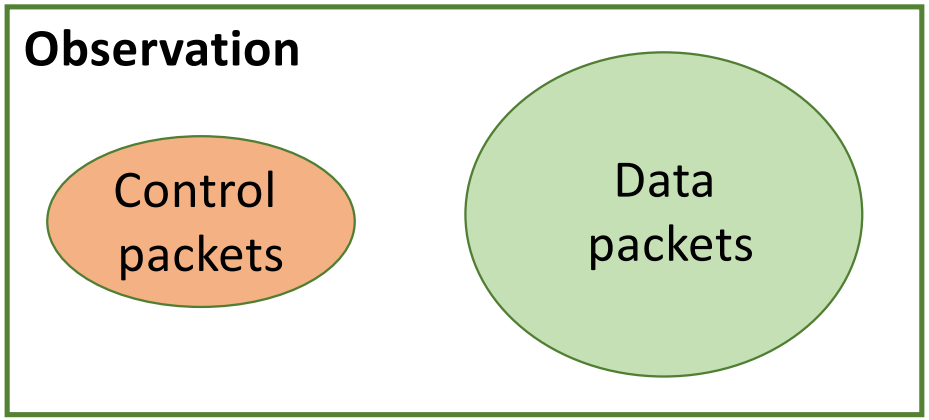

***Finite State Machine (FSM) of stateful packet filtering***

*Altogether tens of state transitions among all states in the FSM.*




Stateful ALU supports very few branch operations

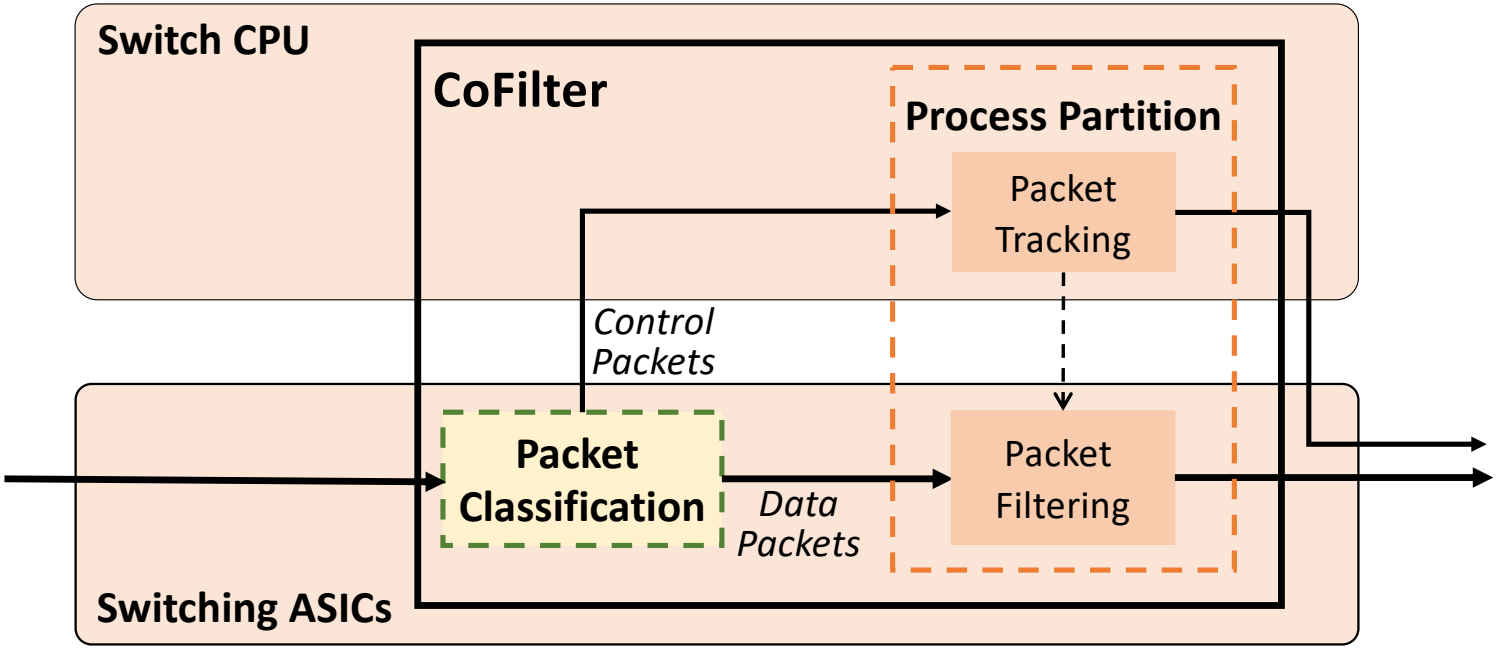


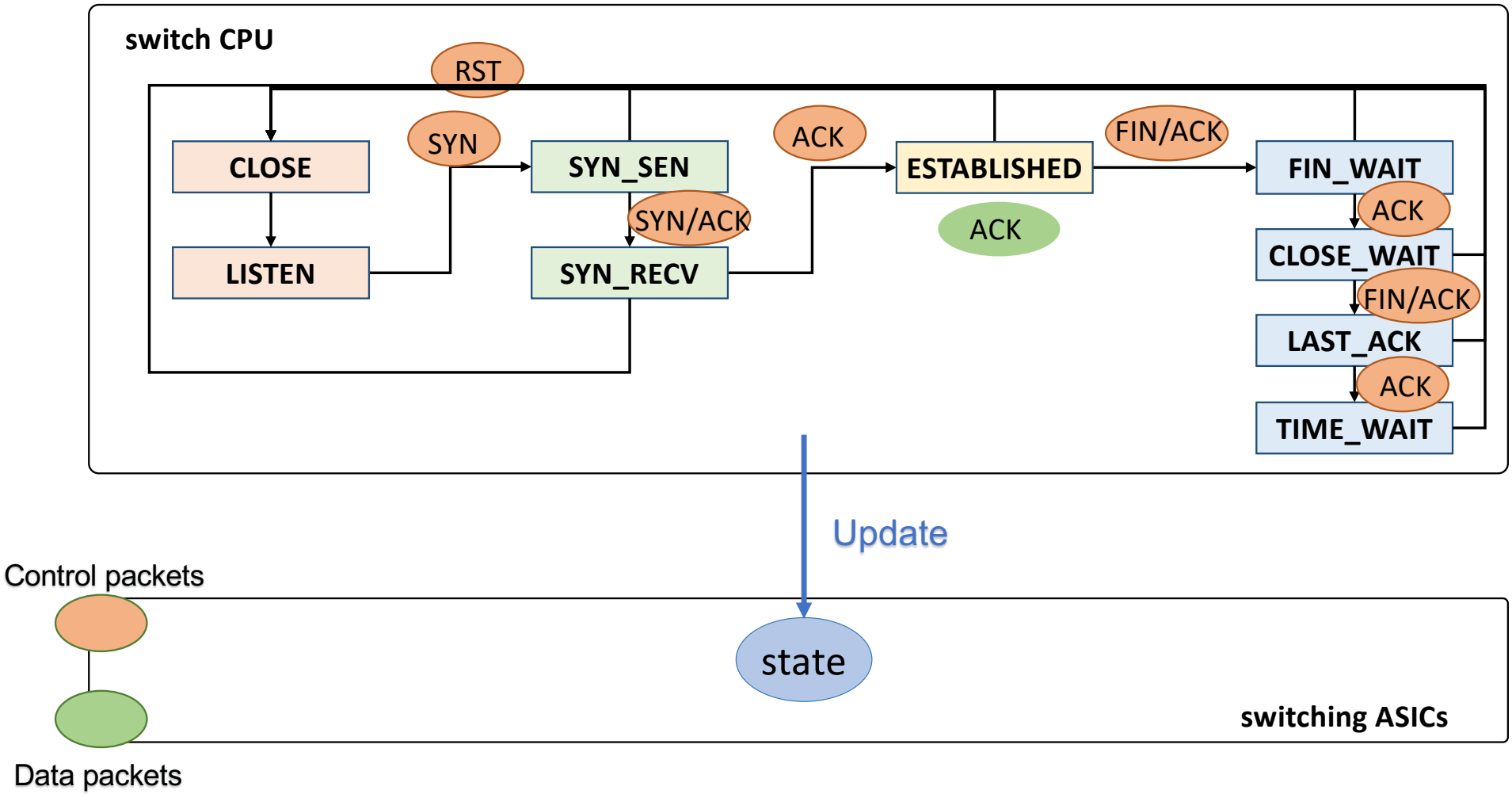




Control pakcets: trigger state transition

Data packets: have no impact on connection state

# Process partition




switch CPU
RST
CLOSE
SYN
SYN_SEN
ACK
ESTABLISHED
FIN/ACK
FIN_WAIT
ACK
CLOSE_WAIT
FIN/ACK
LAST_ACK
ACK
TIME_WAIT
LISTEN
SYN/ACK
SYN_RECV
ACK
Update
Control packets
state
switching ASICs
Data packets

C2

**Scalability requirement for tracking massive connections**

*VS*

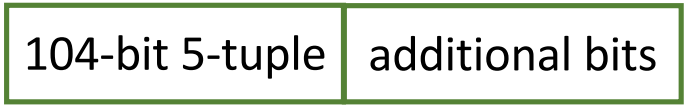
**Limited memory space in programmable ASICs**

**Hash?** Hash incurs hash collision

**Exact match?**

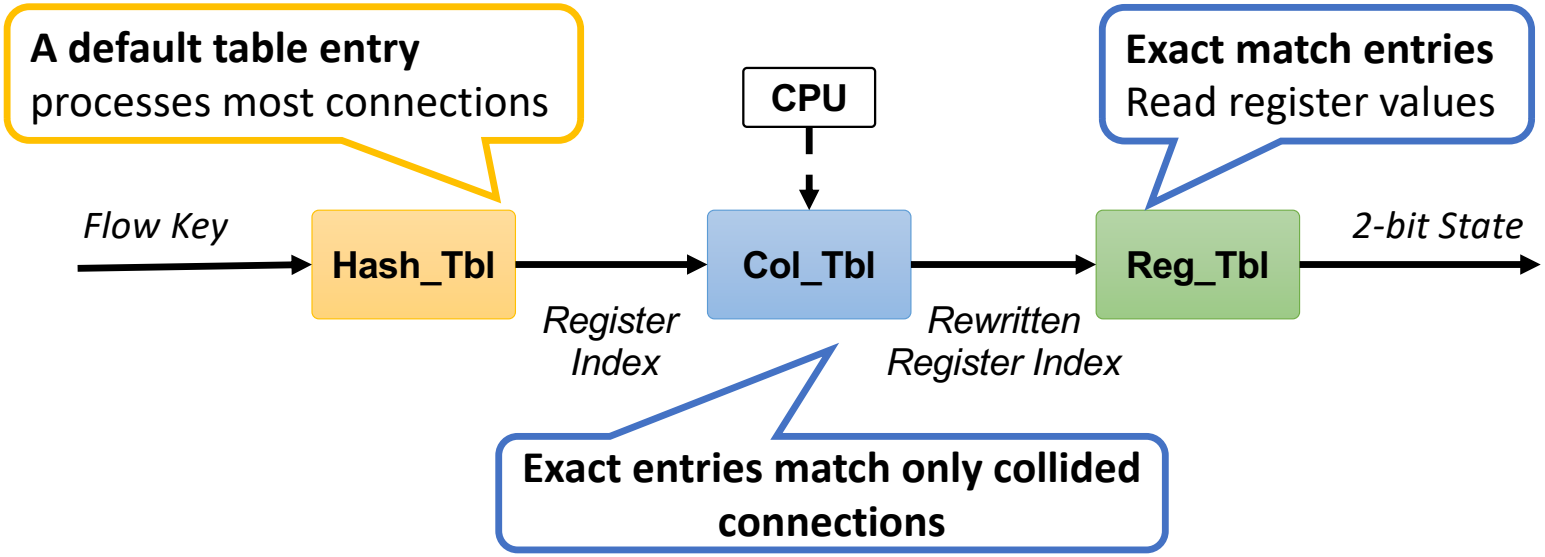
*Exact match-action tables to map each 5-tuple to the index of a register:*

*One connection:*



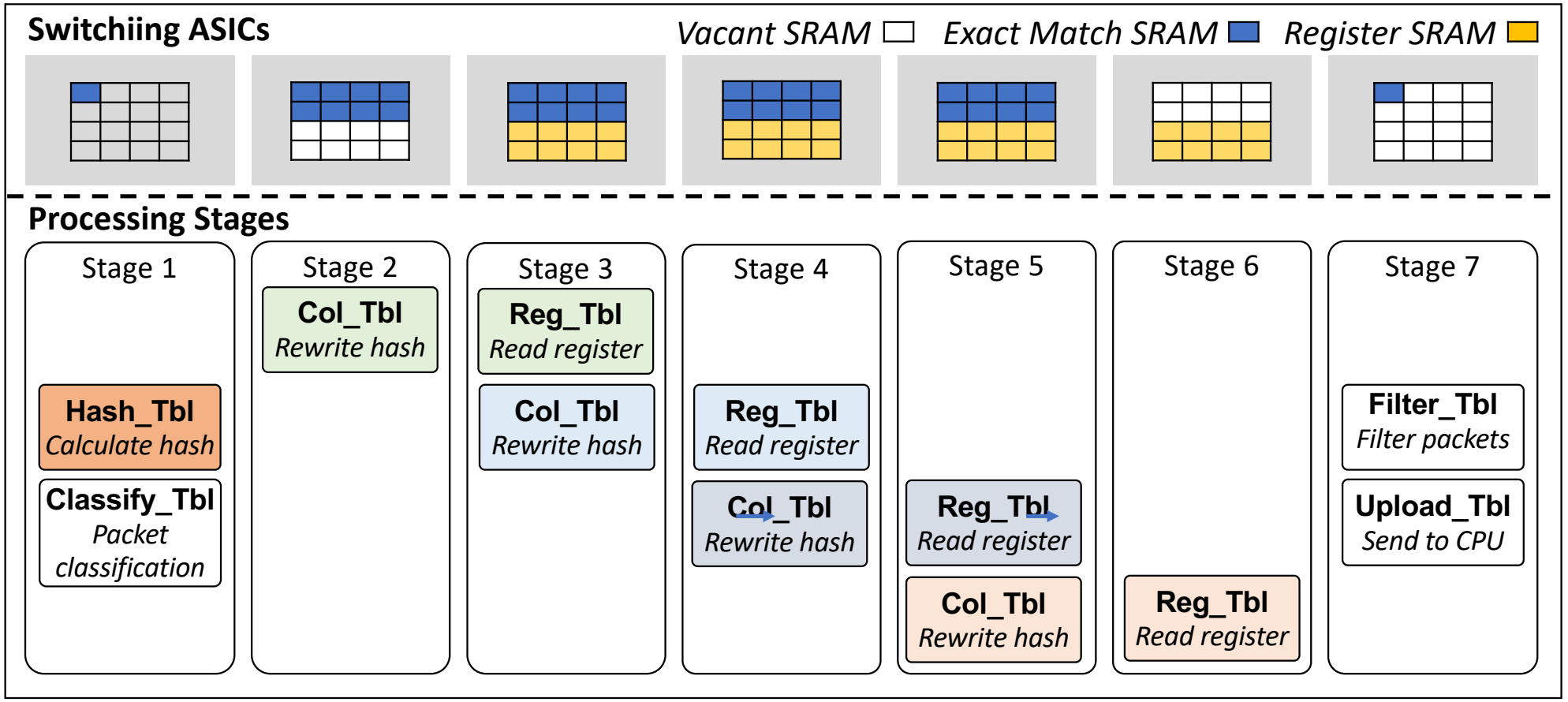

*Ten million connection: hundreds of MB SRAM*

# Hash compression and collision settlement

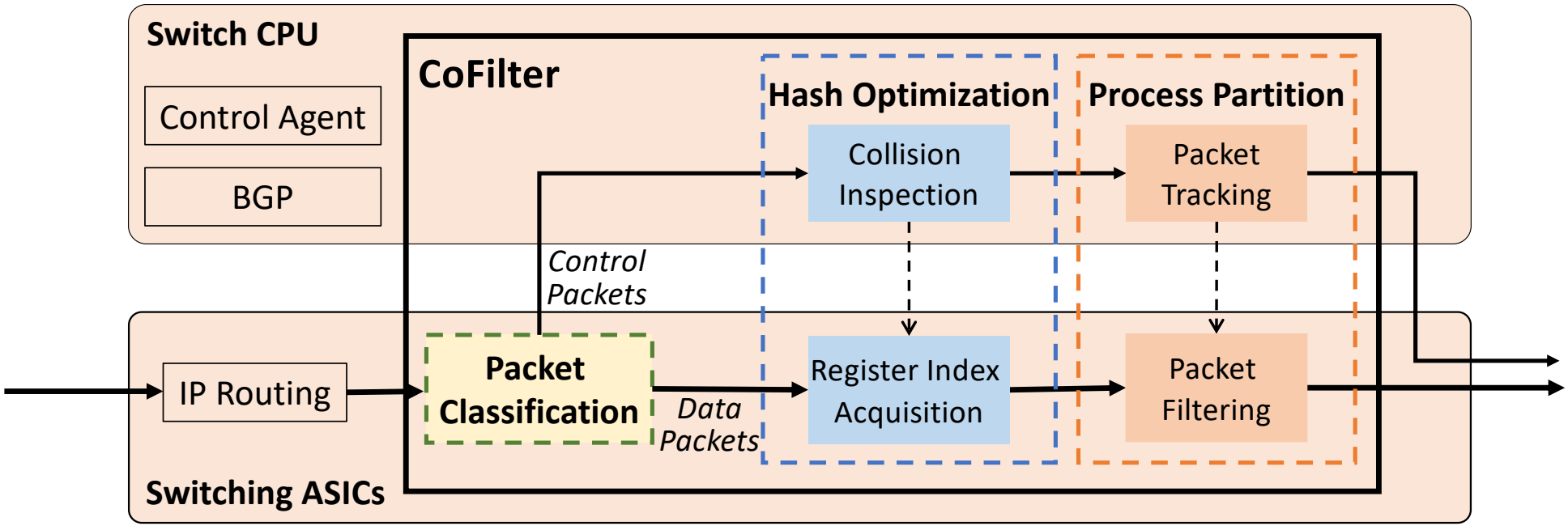



A CPU-assisted three-phase hash collision settlement scheme on programmable ASICs

# Cross stage table placement

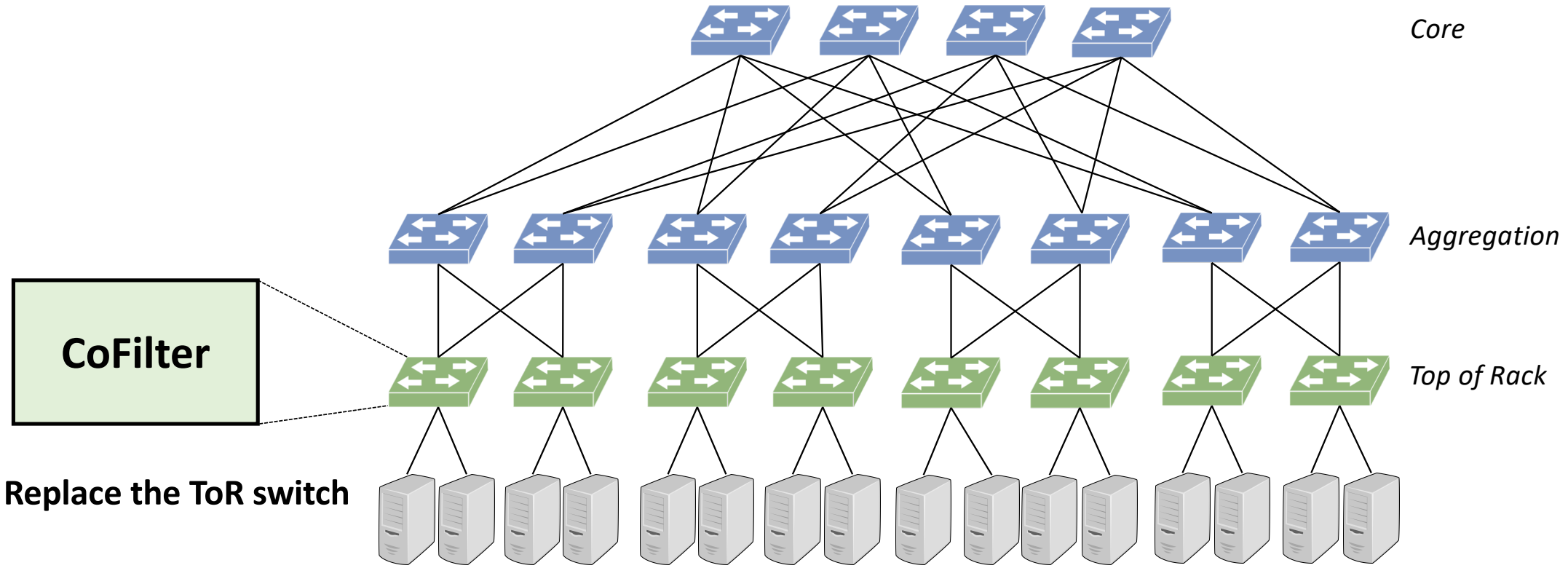



**Make full use of fixed per-stage resources**

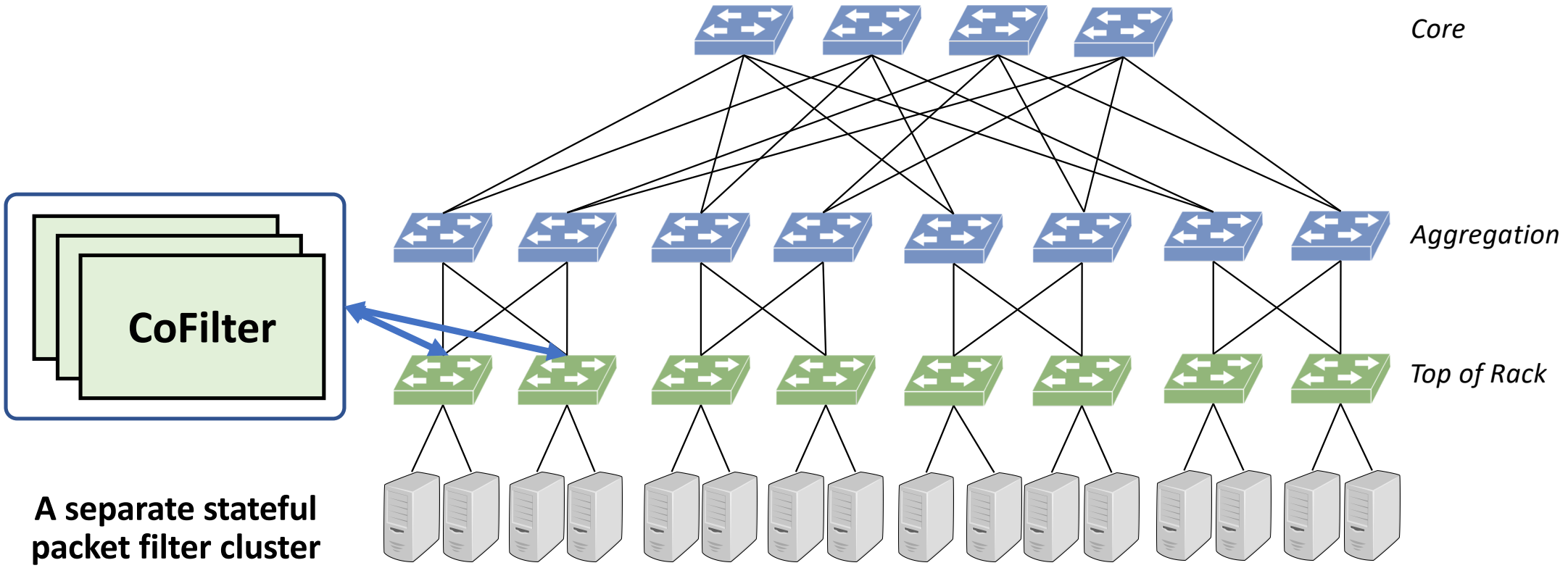



Overall architecture of CoFilter

# Typical deployment scenario - 1

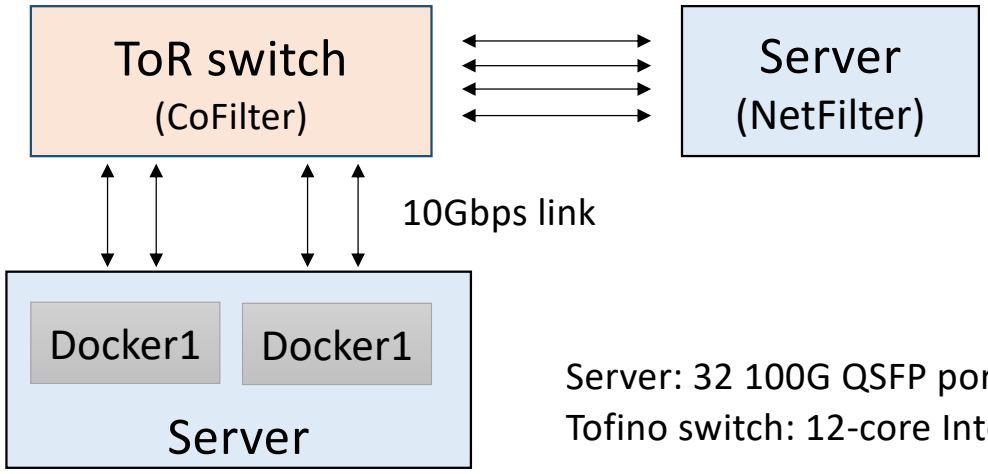


# Typical deployment scenario - 1



# Evaluation

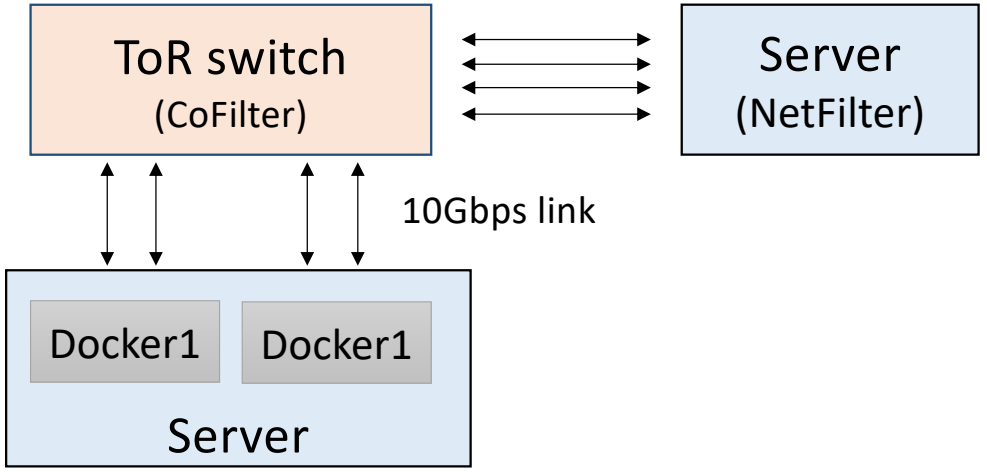
- **Implementation**
  - About 400 lines of P4 code to configure ASICs
  - About 200 lines of C code on switch CPU on top of ConnTrack
- **Evaluation setup**
  - Testbeds
  - Four realistic flow distributions to generate traffic
    - DCTCP, VL2, FACEBOOK CACHE, FACEBOOK HADOOP




Server: 32 100G QSFP ports, 4 Intel Pentium 1.60GHz CPU cores
Tofino switch: 12-core Intel Xeon E5-2620 2.40GHz CPUs

# Evaluation

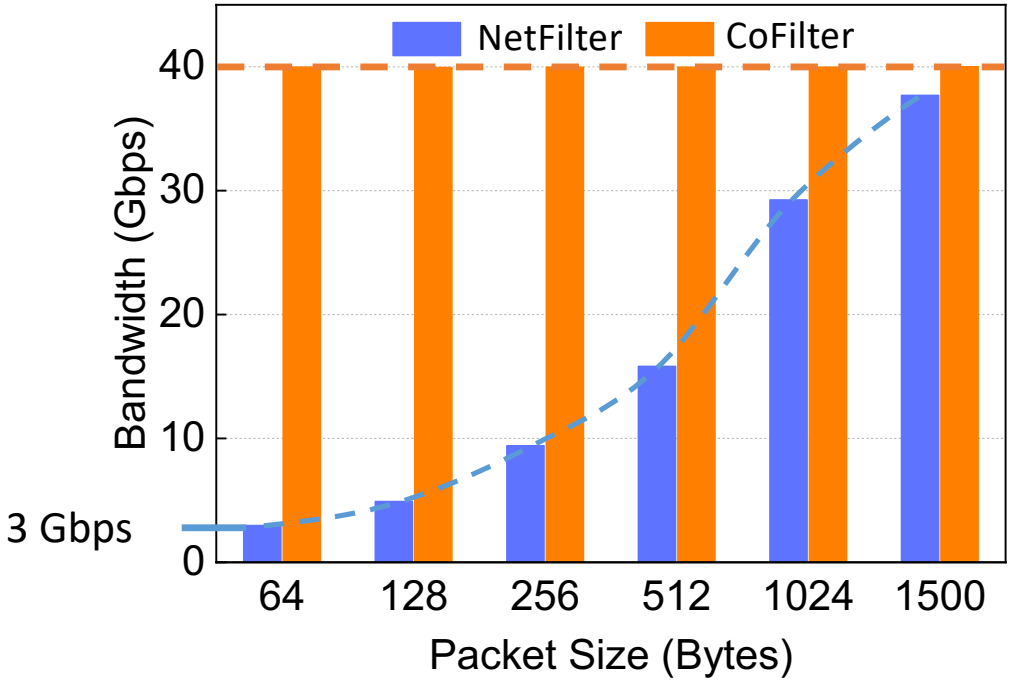
- **Metrics**
  - Micro benchmark
    - Throughput
    - Data/control packet delay
  - End-to-end
    - Flow completion time
  - Scalability
    - ASIC resource usage
    - ASIC capacity
    - CPU resource usage



# Result highlight

- Forwarding packets at line rate (13x throughput of NetFilter)
- Keeping packet delay at 1us
- Great scalability and accommodates over ten million connections with only 16MB SRAM.
- Freeing a significant quantity of CPU cores

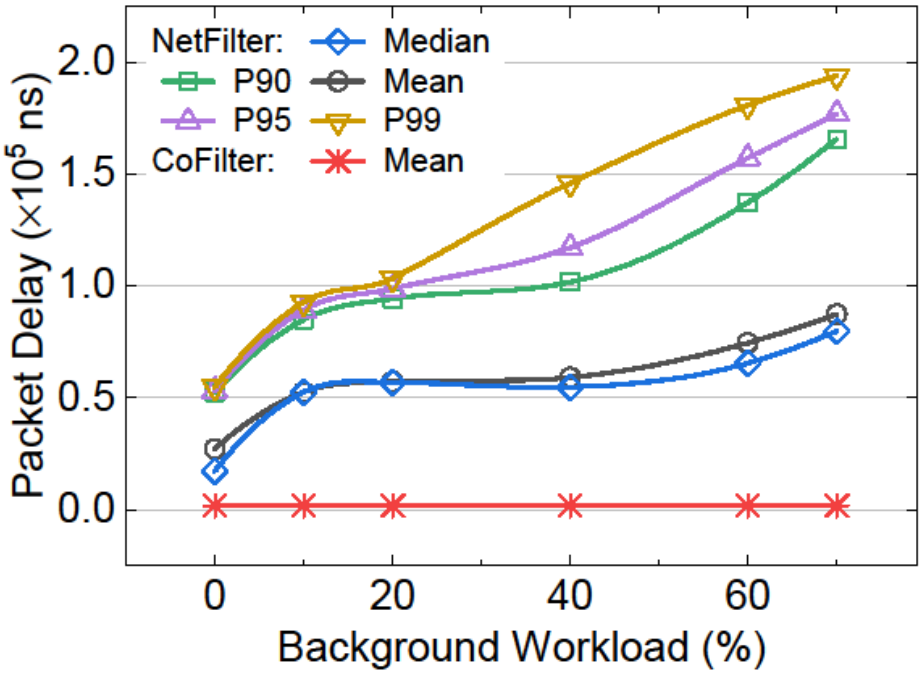
# Throughput

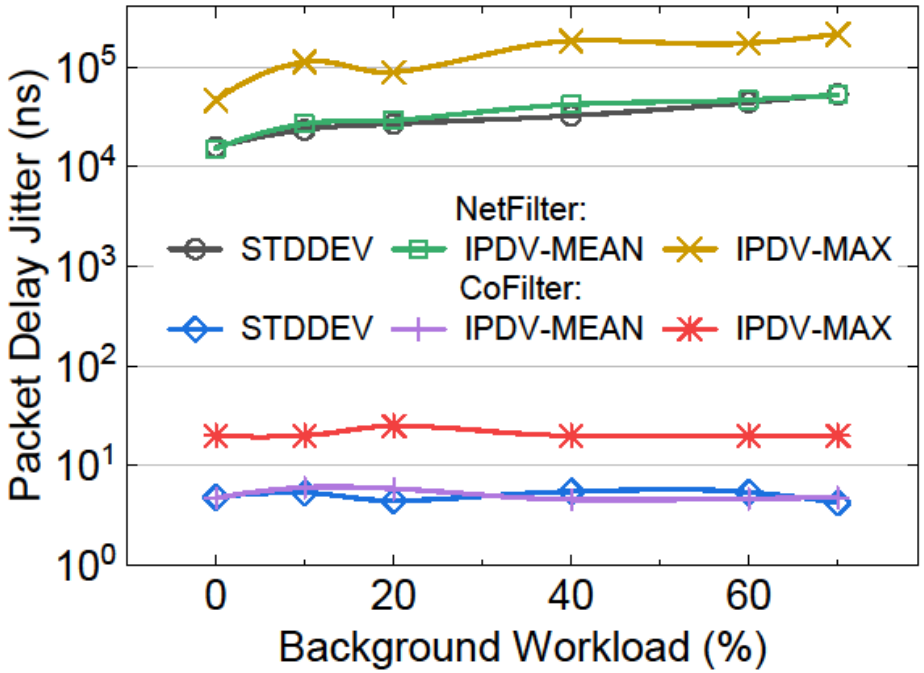



- CoFilter can always forward data packets at line rate (40Gbps)
- NetFilter reduces throughput sharply with smaller packet sizes.
- For 64-byte packets, NetFilter achieves only 3Gbps, 13x smaller than CoFilter.

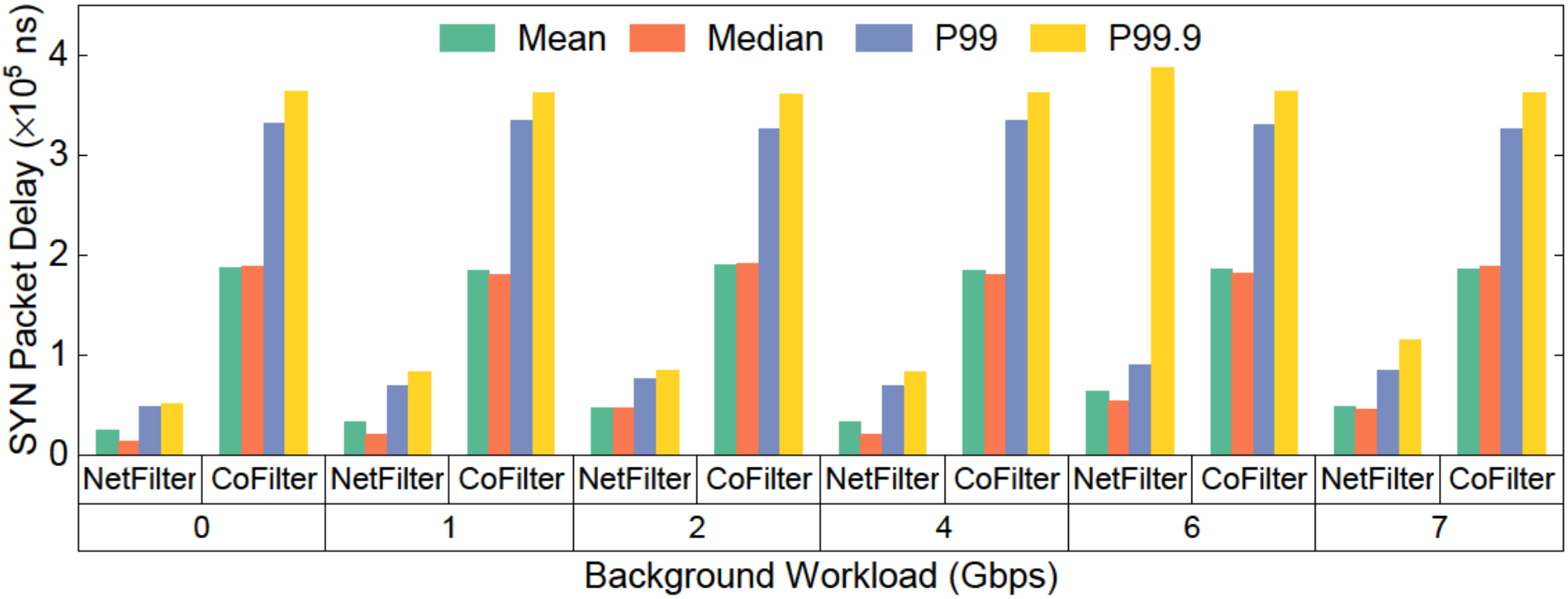
# Data packet delay

IPDV: instantaneous packet delay variation
STDDEV: standard deviation



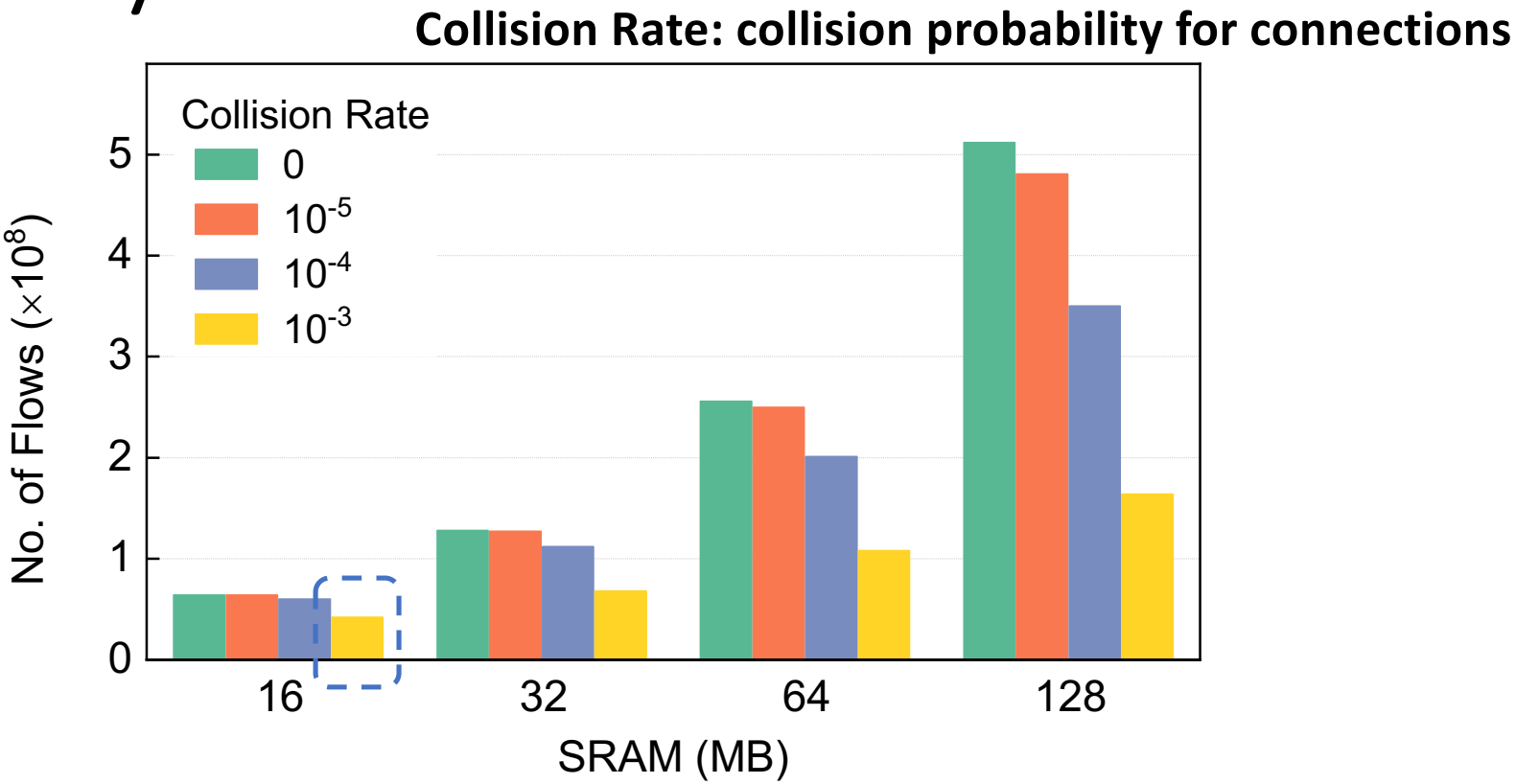



# Control packet delay




- CoFilter has larger SYN packet delay compared with NetFilter.
- CoFilter keeps the delay at a constant value.
- As the workload increases from 0 to 7Gbps, the gap is smaller.

# ASIC capacity

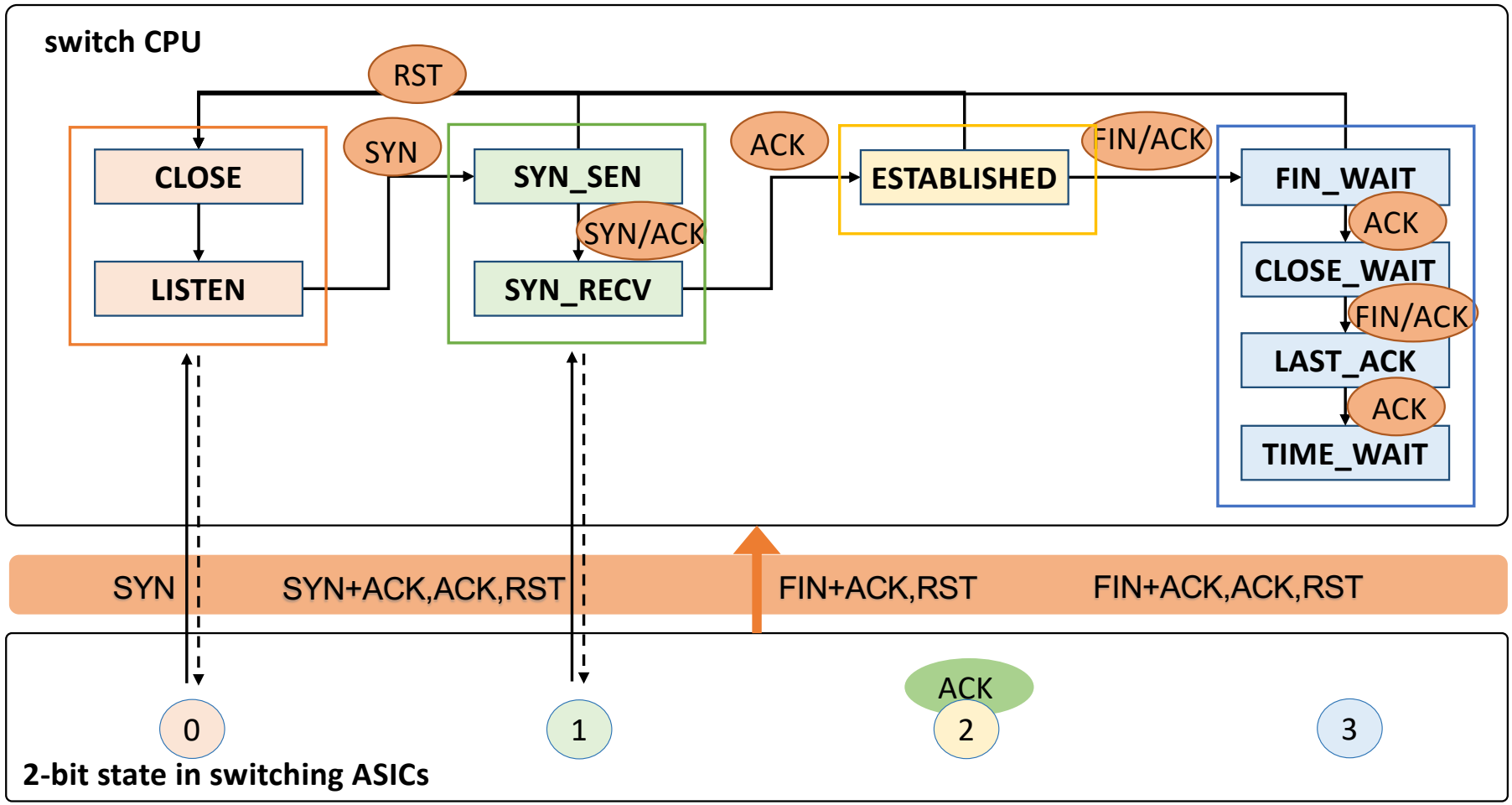



With the collision rate = $10^{-3}$, CoFilter can store more than $10^{7}$ connections with 16MB SRAM

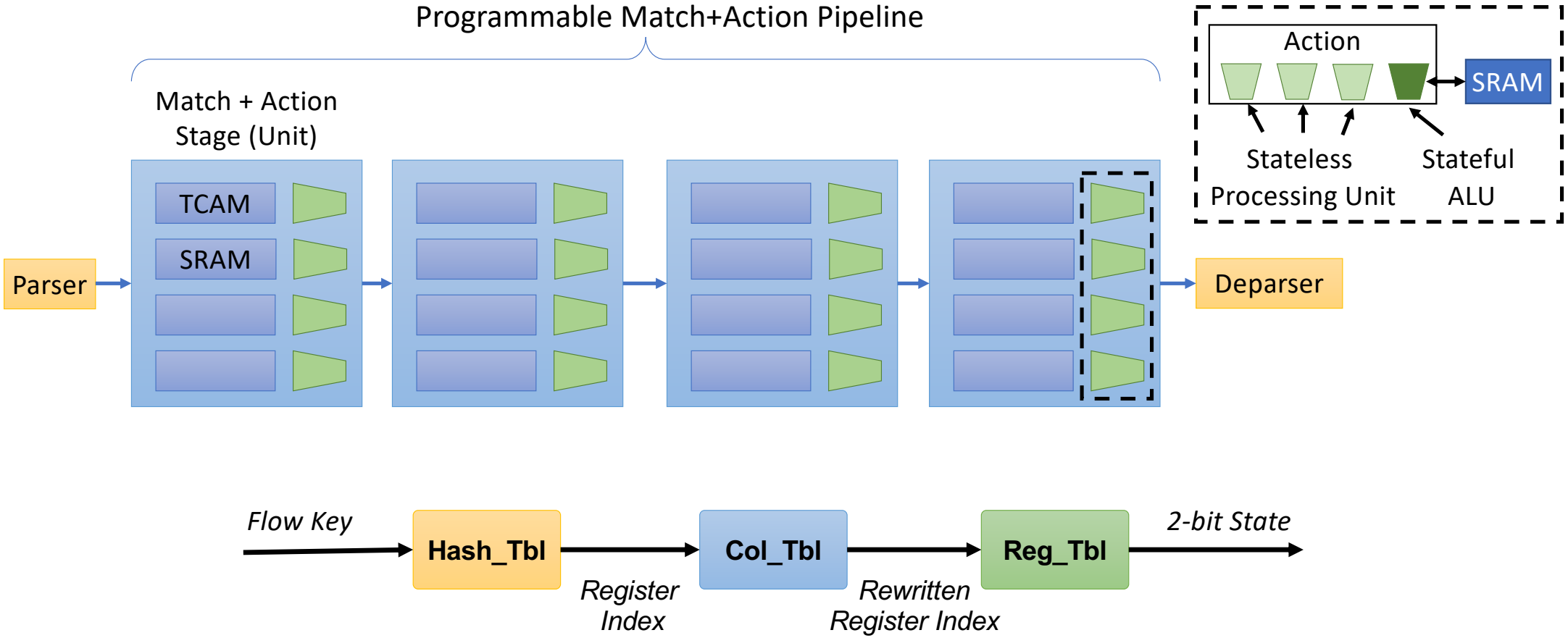
# Conclusion & Future Work

- **CoFilter uses programmable switches to meet these requirements and proposes a co-design between programmable ASICs and switch CPU**
  - process partition
  - hash optimization
- **CoFilter inherits the advantages of programmable ASICs**
  - high throughput, low packet delay, low cost
  - high scalability, high connection capacity, low switch CPU usage
- **Future work**
  - Considerations of security of switch CPU, e.g., the switch CPU can be vulnerable to denial-of-service attacks such as TCP SYN flood

# Thanks!




# 2-bit State compression in ASICs

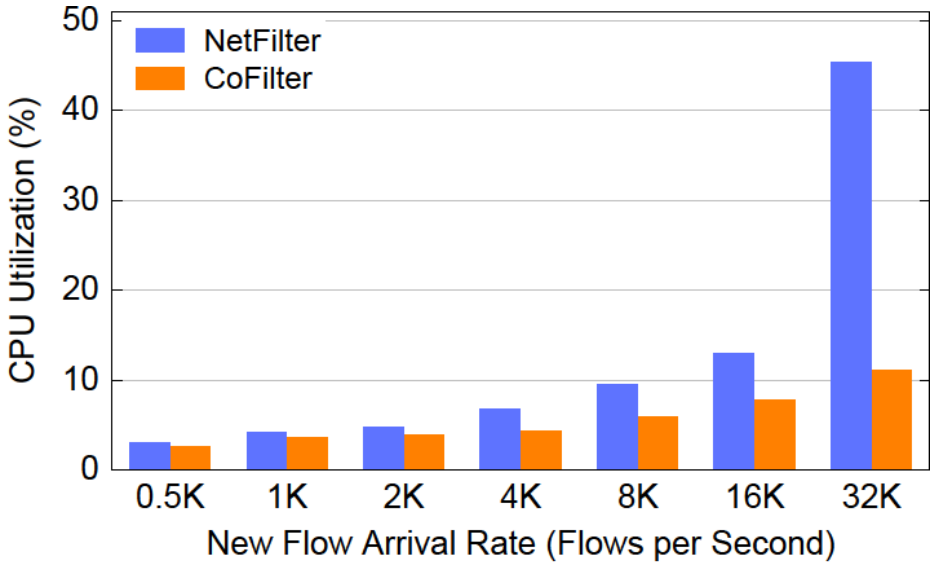


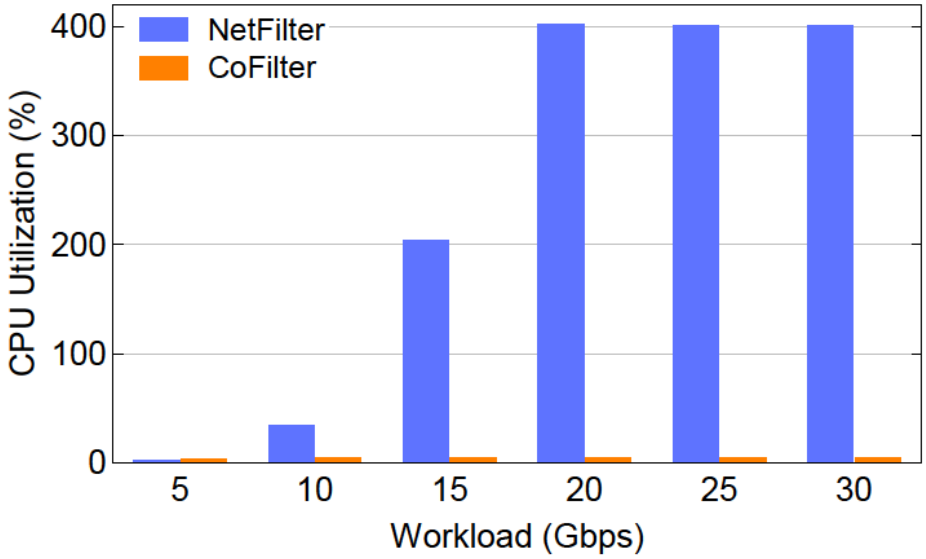

**Programmable switch architecture posses two restrictions on our design**

*Think over the resource occupation of each table and place them in proper stages*

*Sequential operations must be implemented in multiple stages in sequential order*

# CPU usage




The server CPU of NetFilter becomes a considerable bottleneck at high network speeds and new flow arrival rates, while CoFilter can significantly save CPU.