What you need to know about (Smart) Network Interface Cards

Georgios P. Katsikas, Tom Barbette, Marco Chiesa,
Dejan Kostić, and Gerald Q. Maguire Jr.

KTH Royal Institute of Technology
Stockholm, Sweden

Why are NICs worth studying?

amazon web services

Google Cloud

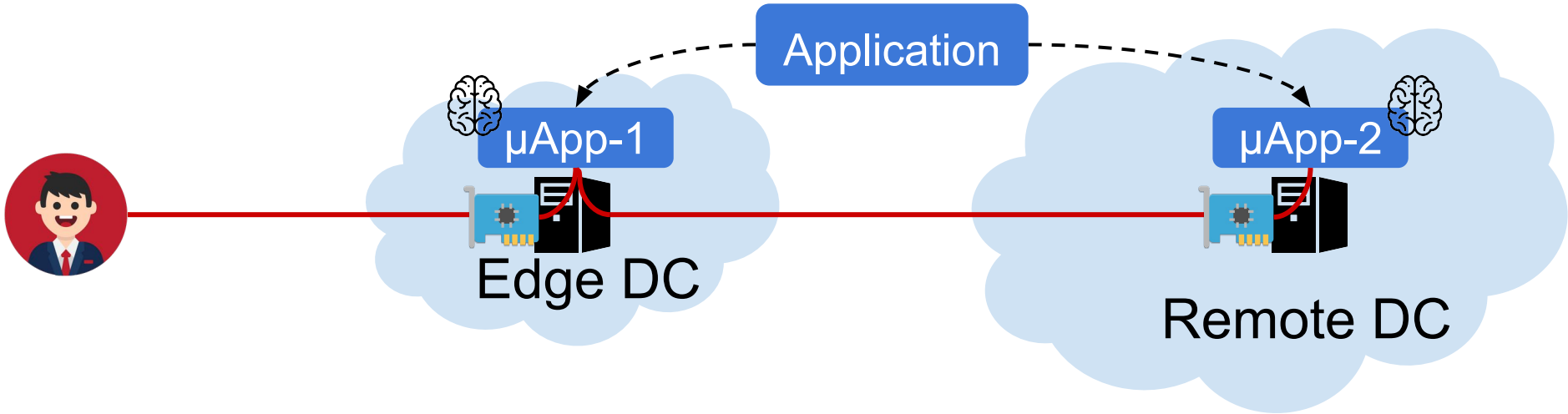
Microsoft Azure

1000s of racks/site

100.000s of servers/site

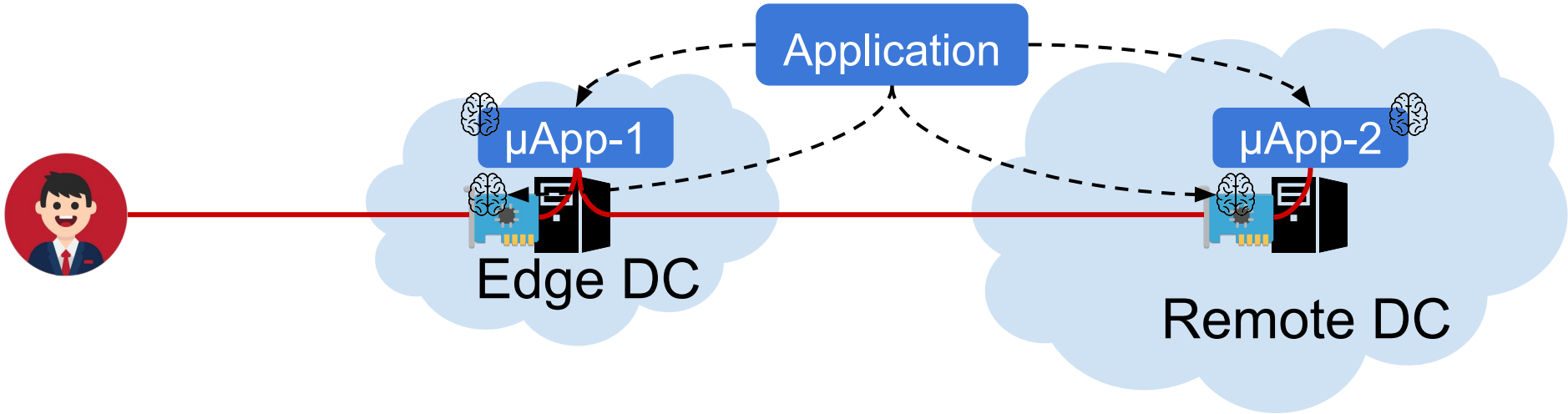
At least 1 NIC/server

Why are NICs worth studying?



Why are NICs worth studying?

NICs increasingly realize portions of the application logic



Main Question

Facts

Today's NICs operate at multi-hundred Gbps (200 Gbps and raising..)

The available time to process small packets at 100 Gbps is **only a few nanoseconds per packet**

Question

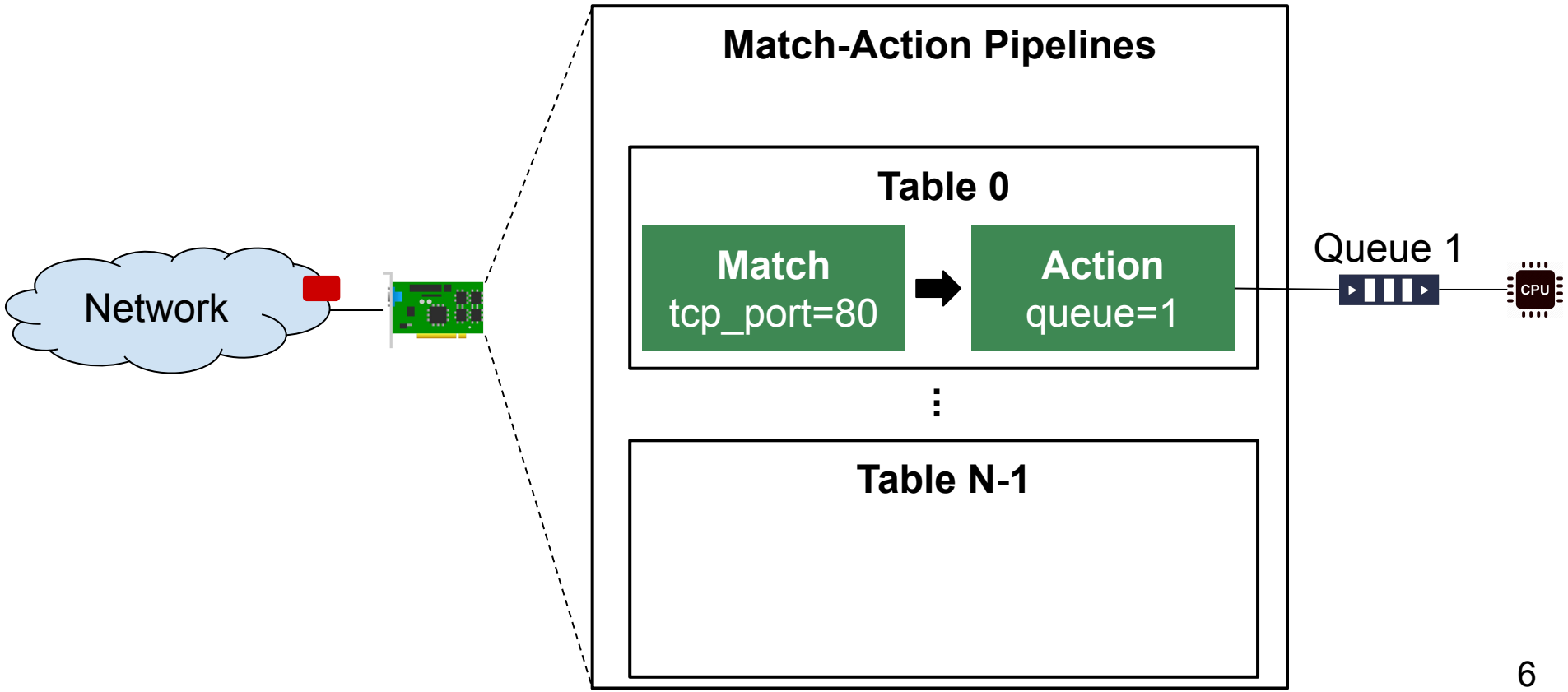
Are today's NICs ready to sustain **both**:

- ❑ high-speed packet processing

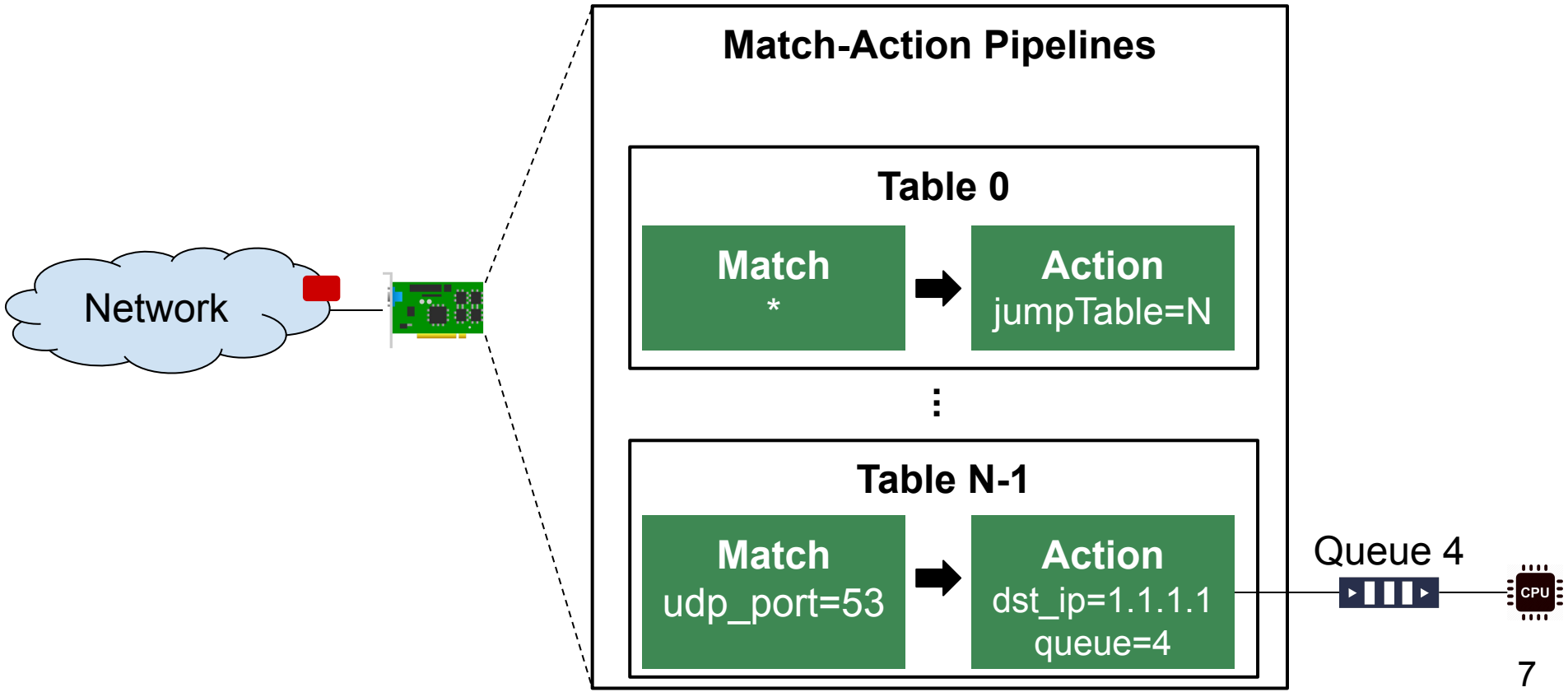
and

- ❑ parts of the application logic?

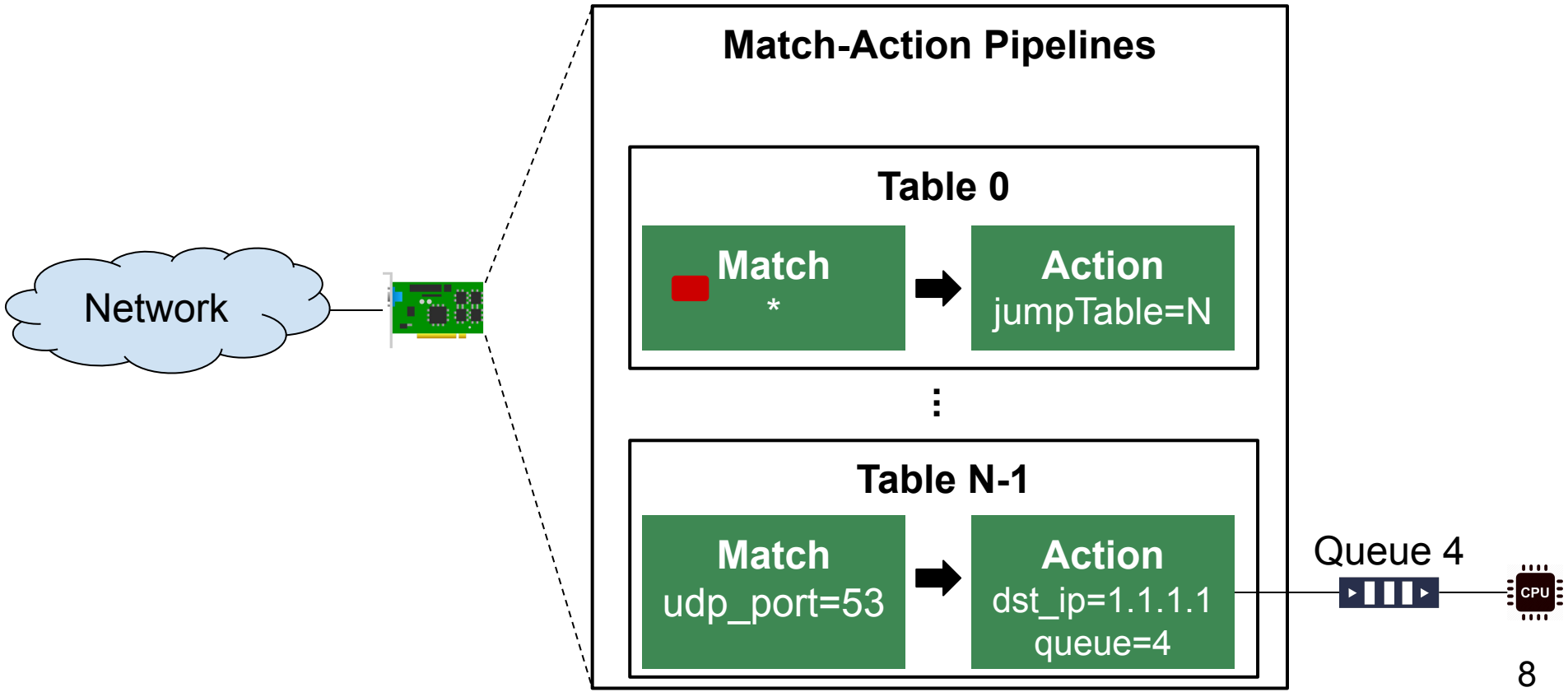
How do NICs perform processing?



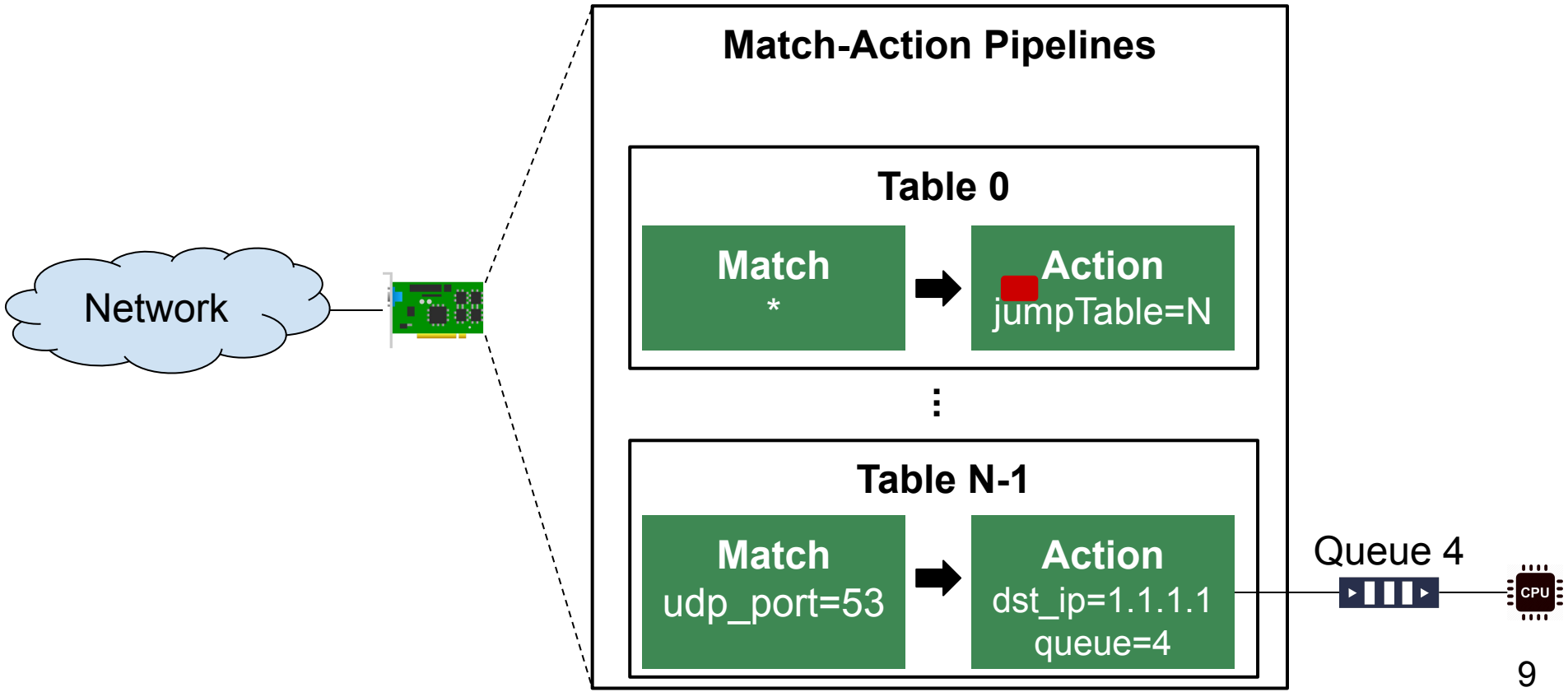
How do NICs perform processing?



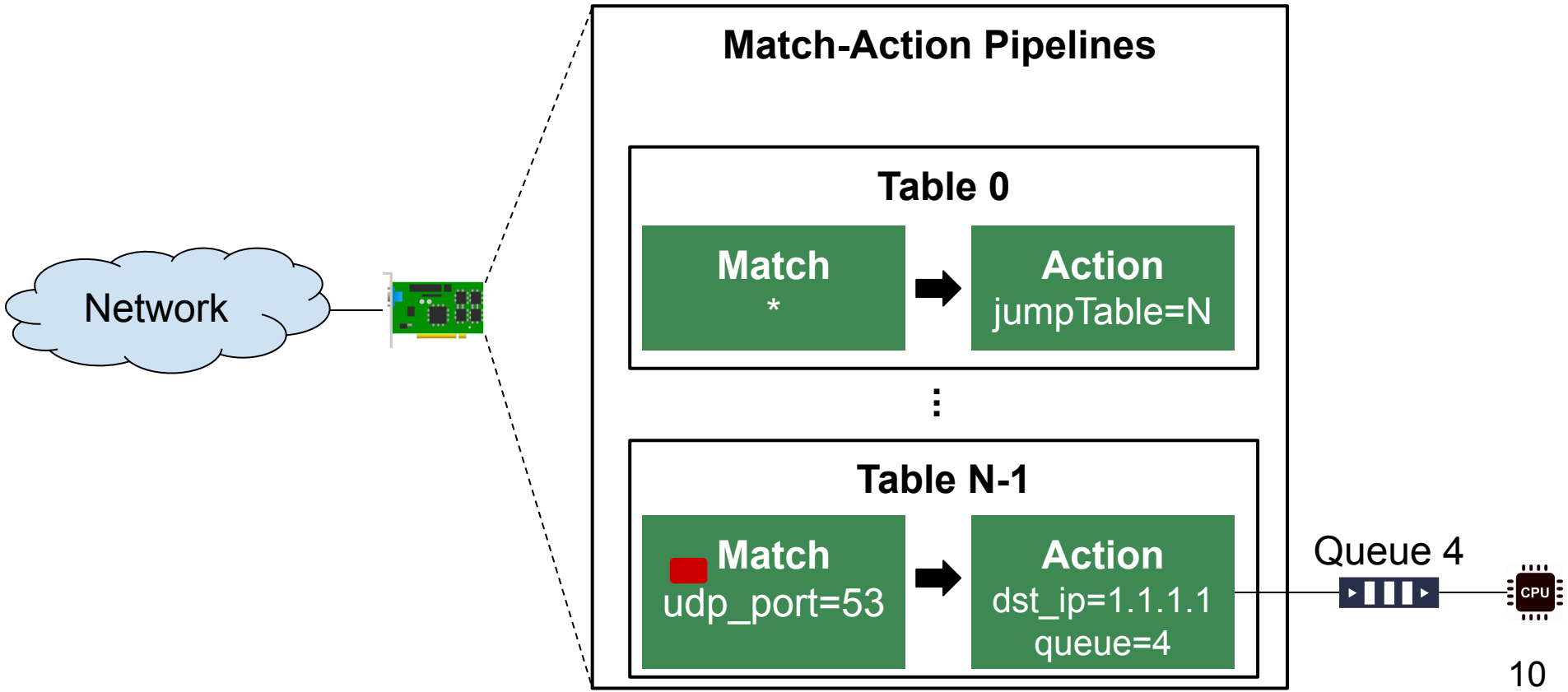
How do NICs perform processing?



How do NICs perform processing?



How do NICs perform processing?



Motivation

- Study the performance of modern NIC classifiers
  - Focus on widely deployed packet classification operations
  - Investigate whether traffic processing is affected by runtime modifications of the packet classifier's ruleset

Research questions

Q1	Does the number of rules and/or tables affect the performance of the NIC?
Q2	Do updates to the classifier affect the performance of the NIC?
Q3	Do rule insertion/deletion operations perform the same for all types of rules?
Q4	How much does it take to update an existing rule?

Research questions

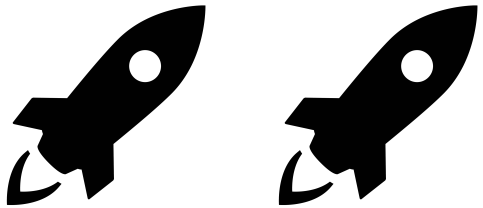
Q1 Does the number of rules and/or tables affect the performance of the NIC?

Q2 Do updates to the classifier affect the performance of the NIC?

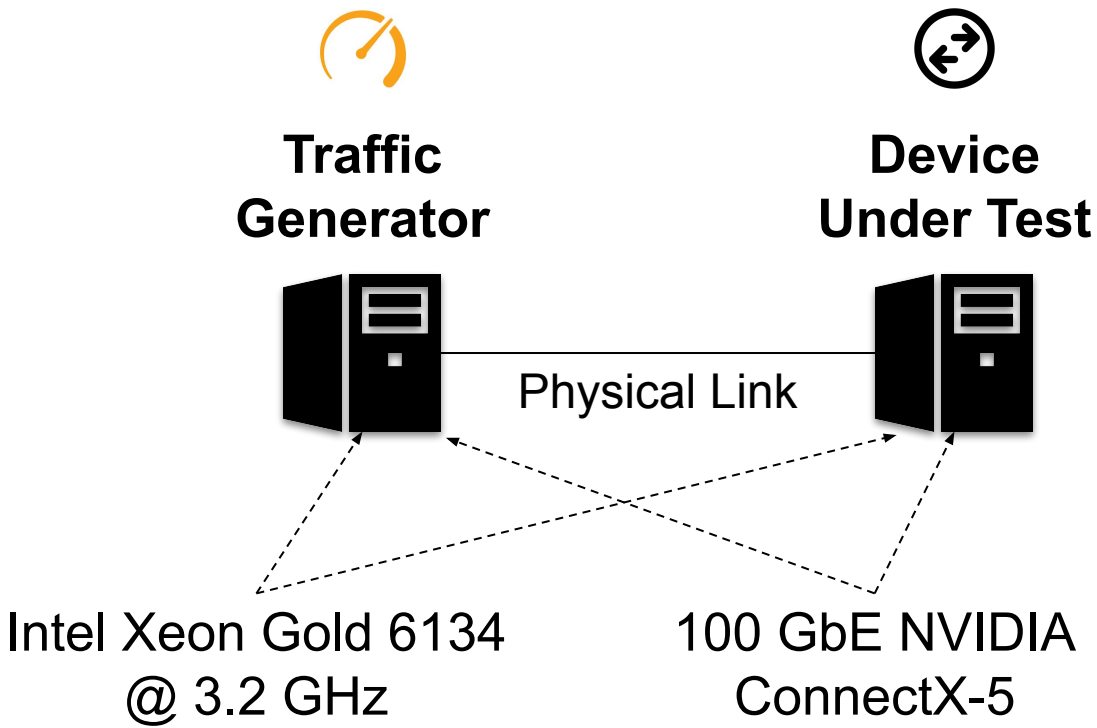
Q3 Do rule insertion/deletion operations perform the same for all types of rules?

Q4 How much does it take to update an existing rule?

Performance Evaluation



Testbed



More results with additional NVIDIA NICs

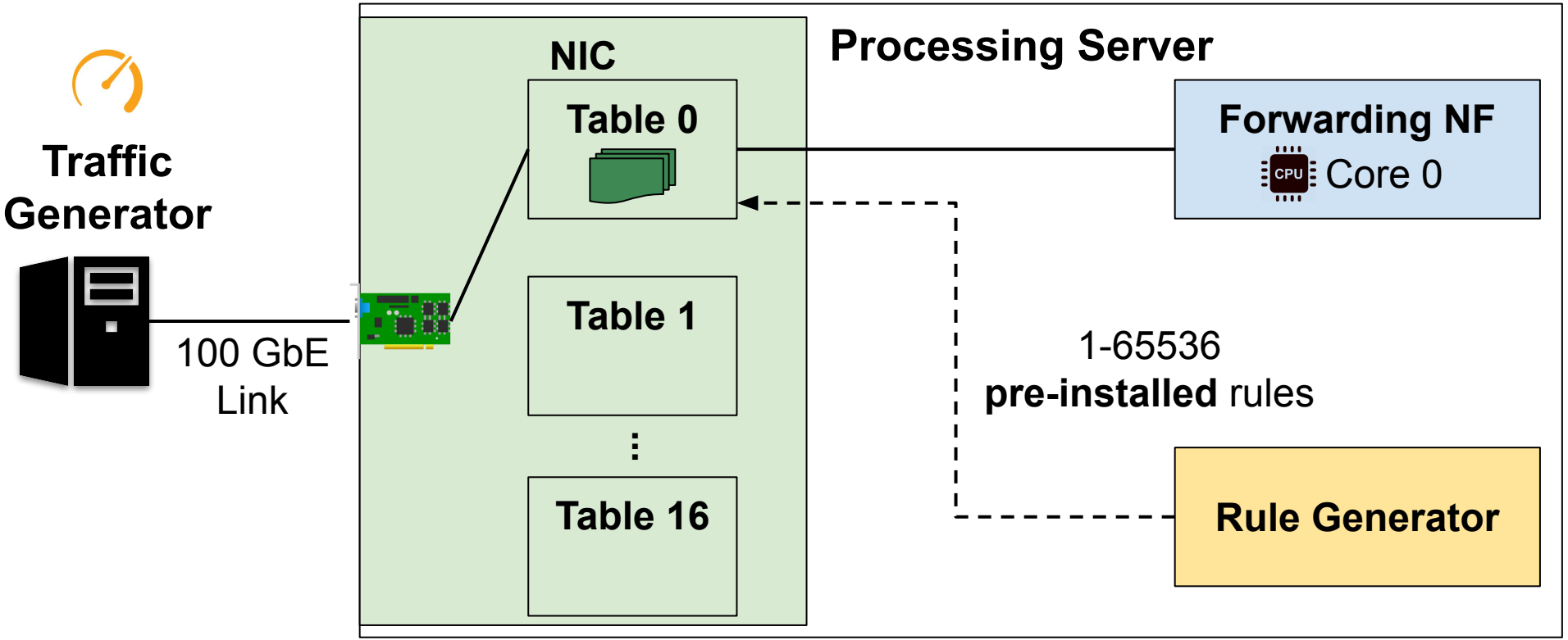
100 GbE ConnectX-4

200 GbE ConnectX-6

100 GbE Bluefield

Scenario 1

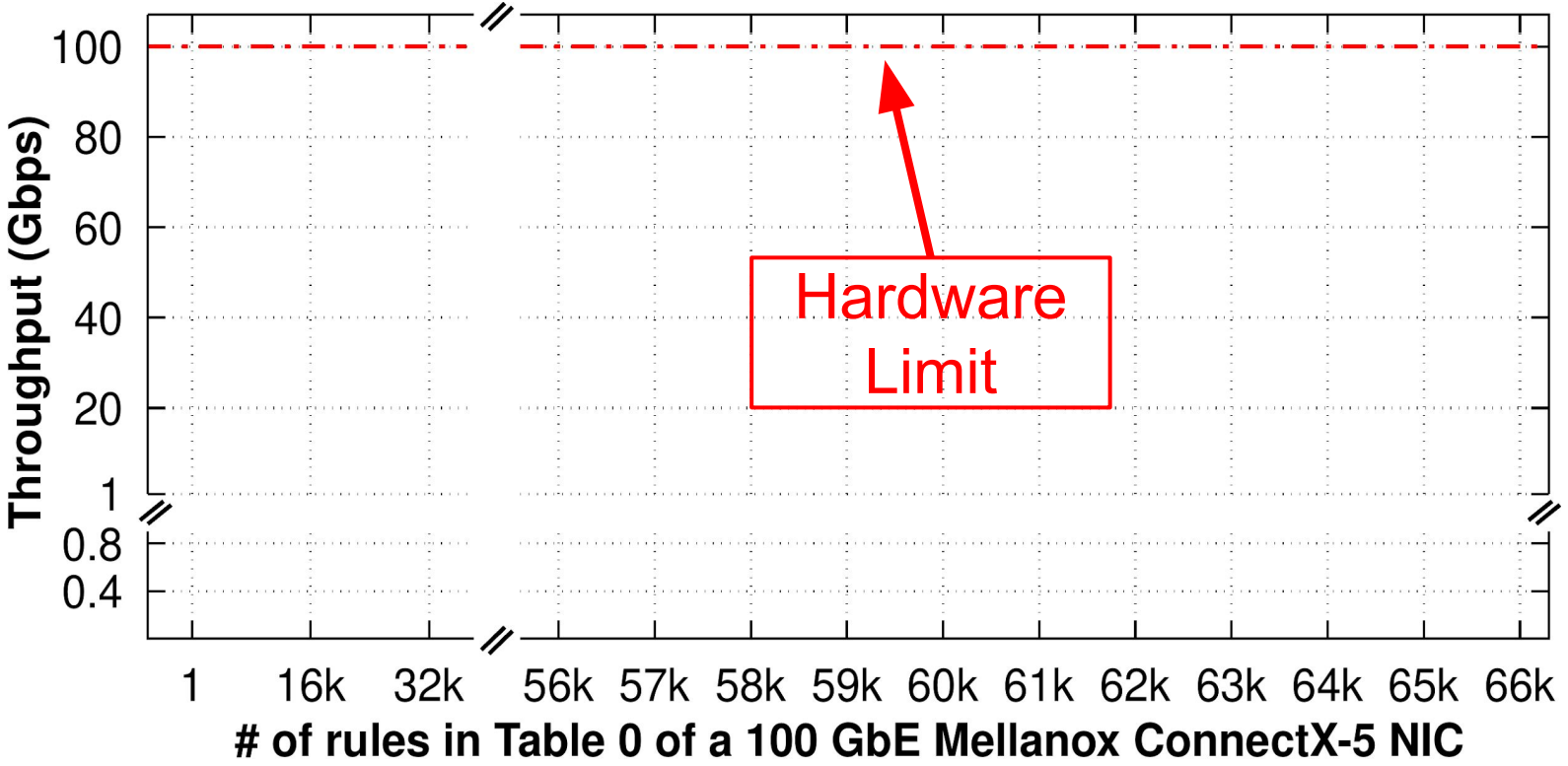
Scenario 1 - Topology



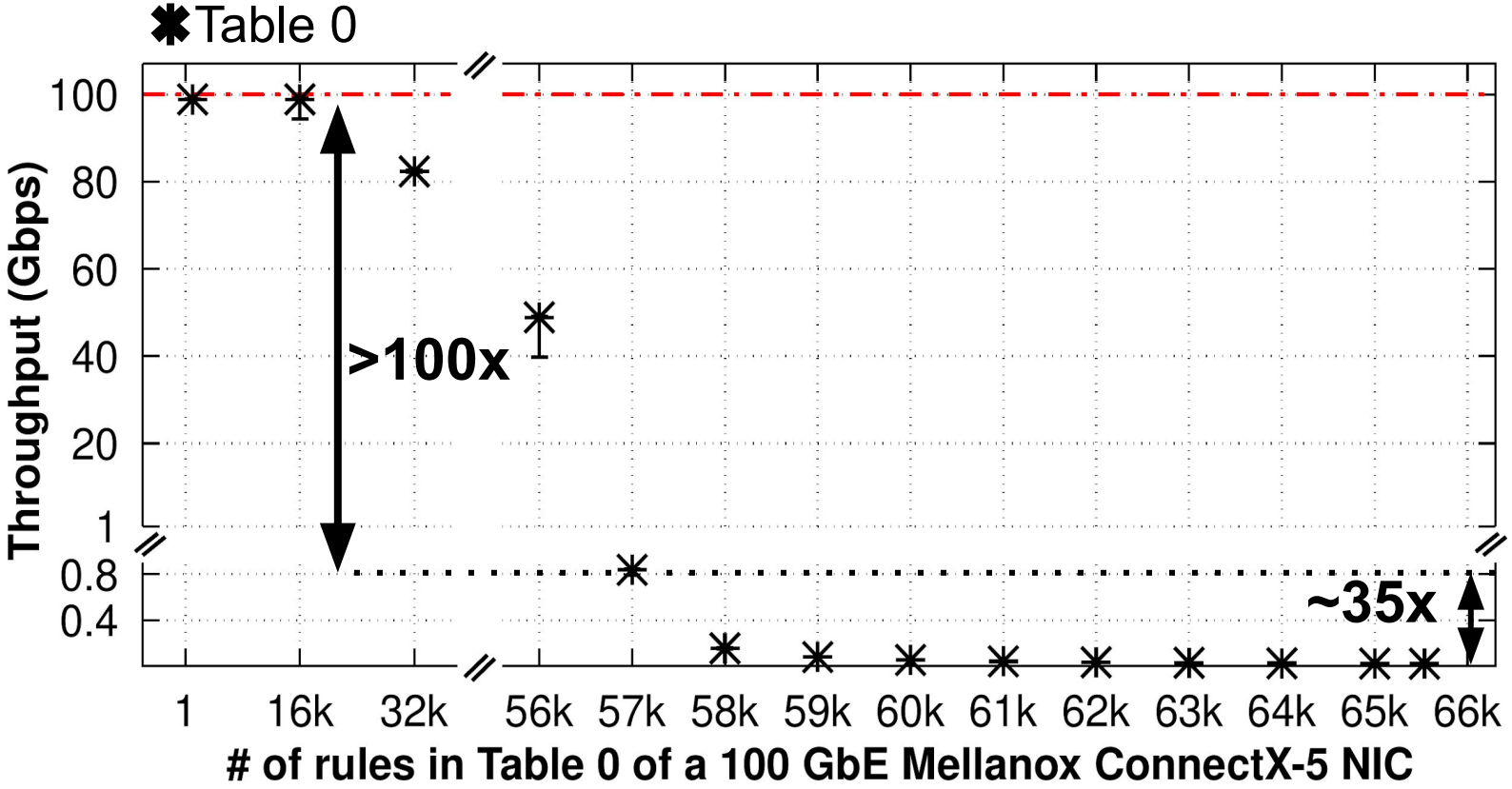
Scenario 1 - Traffic Characteristics

Flows	10k - Proto **UDP**
Rate	**100 Gbps**
Frame length	1500 bytes
Ruleset sizes	1 rule - 65536 rules
How many rules match traffic?	**1**
NIC tables used	**Table 0**
Who installs the rules at the DUT?	Core 0 **pre-installs** rules
Who processes traffic?	Core 0
What do we measure?	Throughput and latency

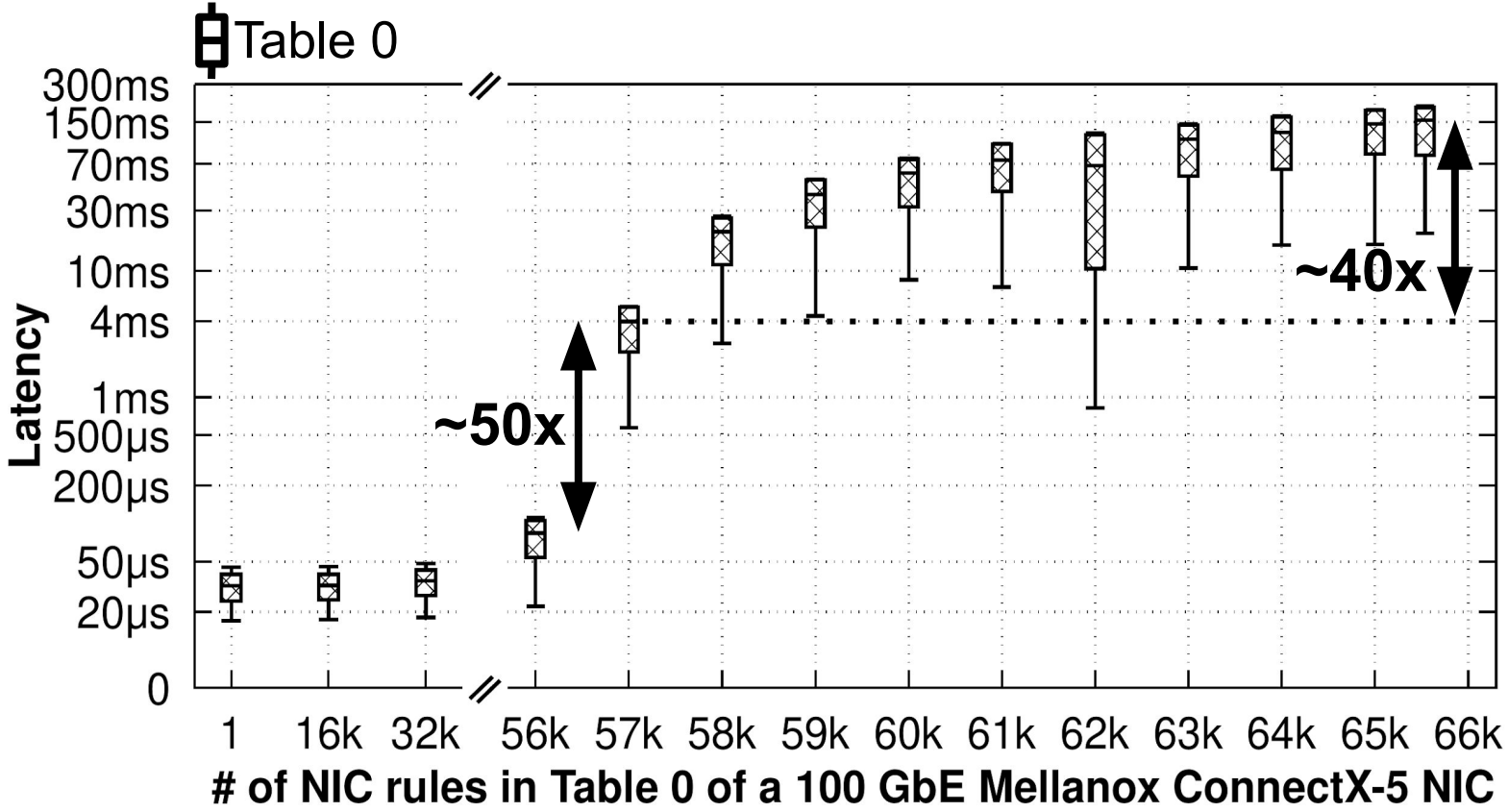
Scenario 1 - Results



Scenario 1 - Results



Scenario 1 - Results



Scenario 1 - Findings

Finding

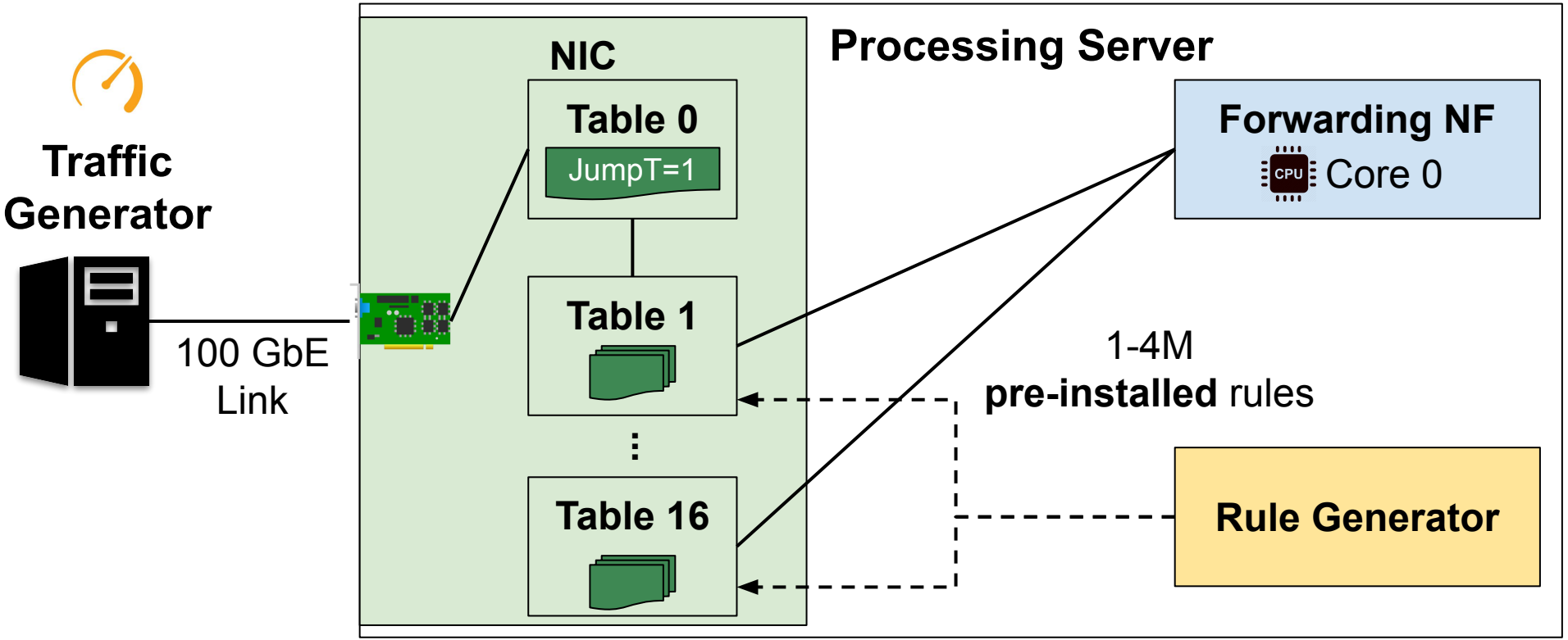
- Table 0 does not yield the expected performance at 85% of its occupancy and above

Implications

- Throughput drops from 100 Gbps to 20 **Mbps**
- Latency increases by several orders of magnitude

Scenario 2

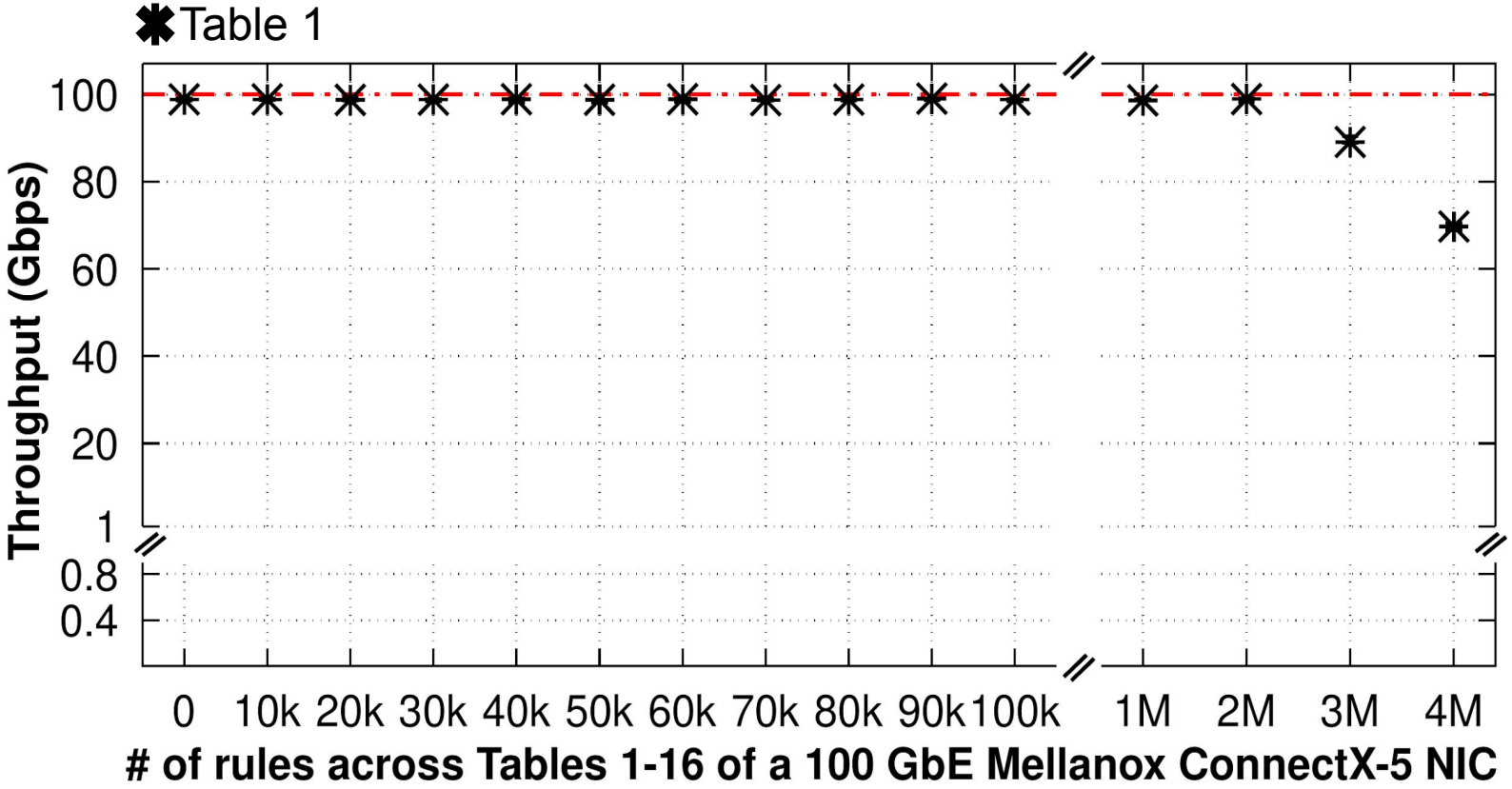
Scenario 1


Traffic Generator
100 GbE Link
NIC
Table 0
JumpT=1
Table 1
Table 16
Processing Server
Forwarding NF
CPU
Core 0
1-4M
pre-installed rules
Rule Generator

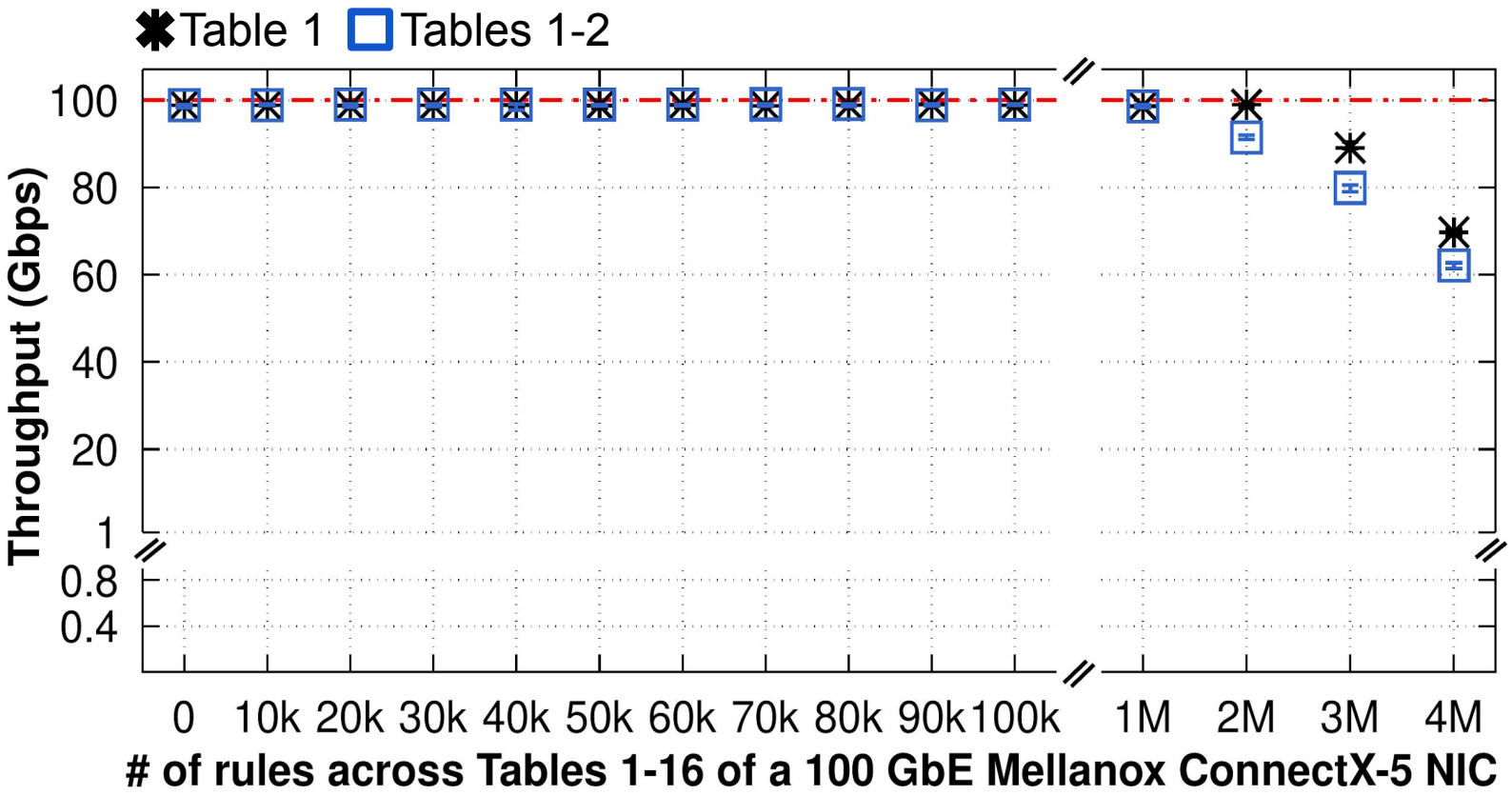
Scenario 1- Traffic Characteristics

Flows	10k - Proto **UDP**
Rate	**100 Gbps**
Frame length	1500 bytes
Ruleset sizes	1 rule - 4M rules
How many rules match traffic?	**1**
NIC tables used	**Tables 1-16**
Who installs the rules at the DUT?	Core 0 **pre-installs** rules
Who processes traffic?	Core 0
What do we measure?	Throughput and latency

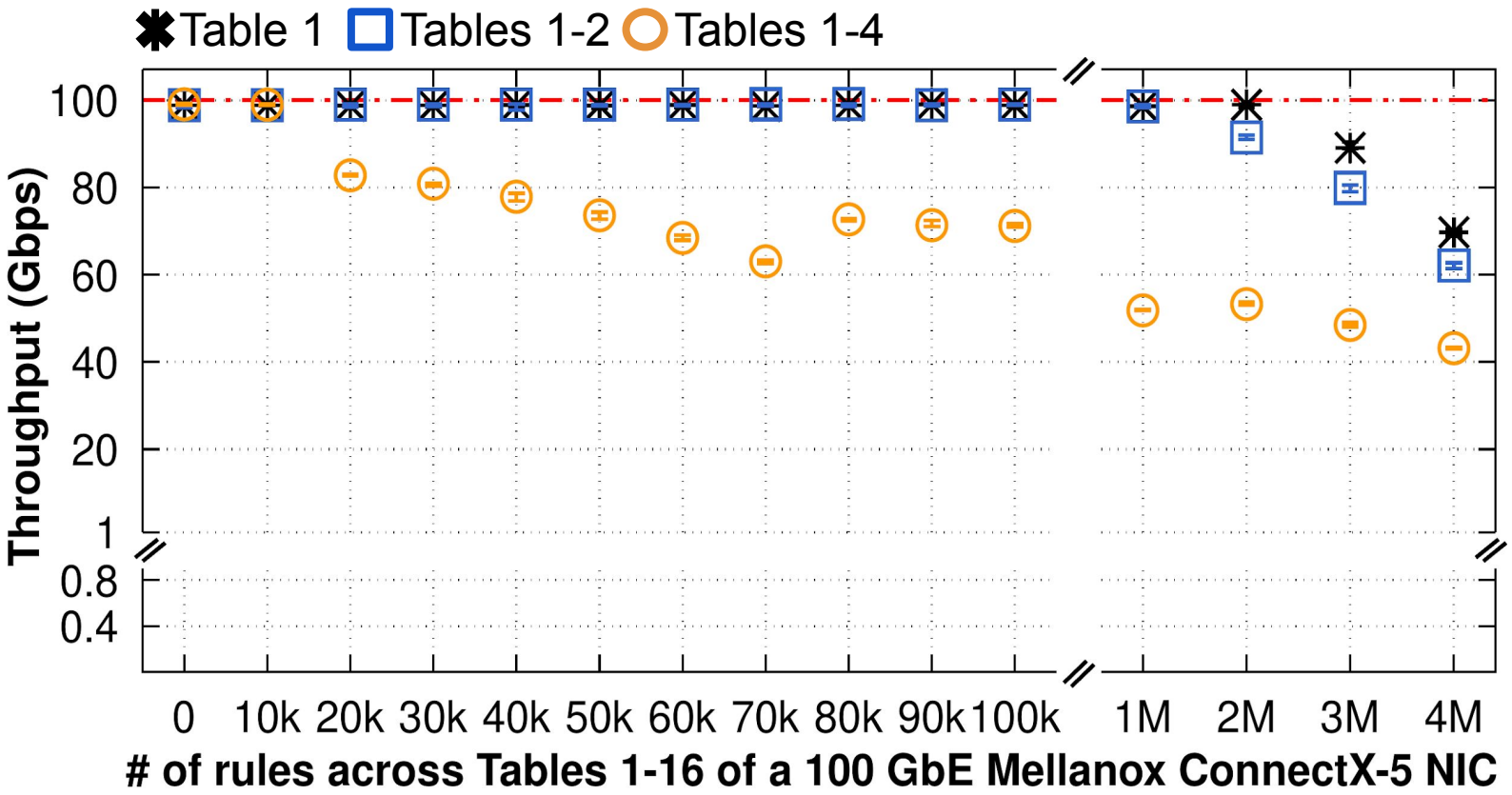
Scenario 1- Results



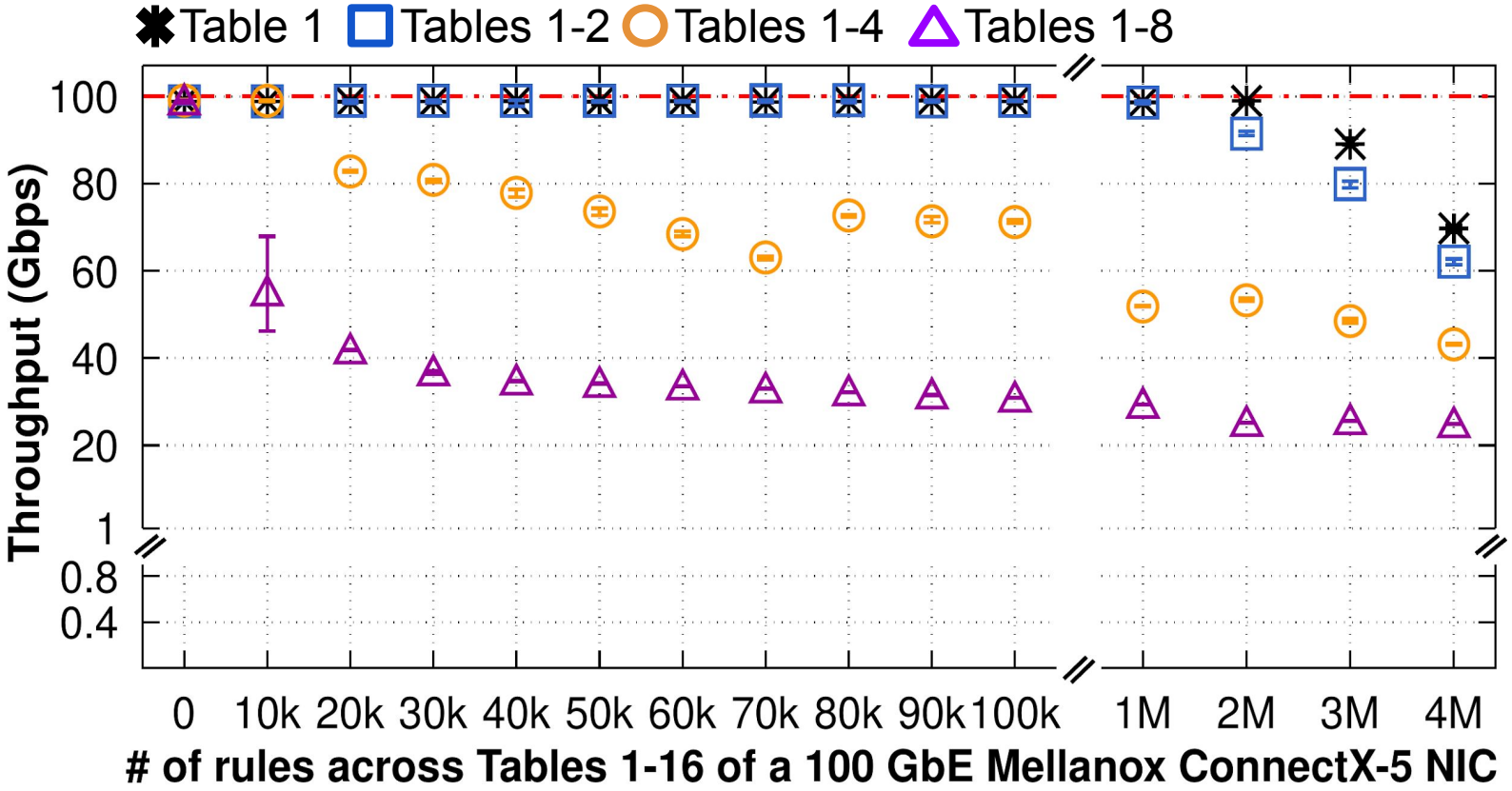
Scenario 1- Results



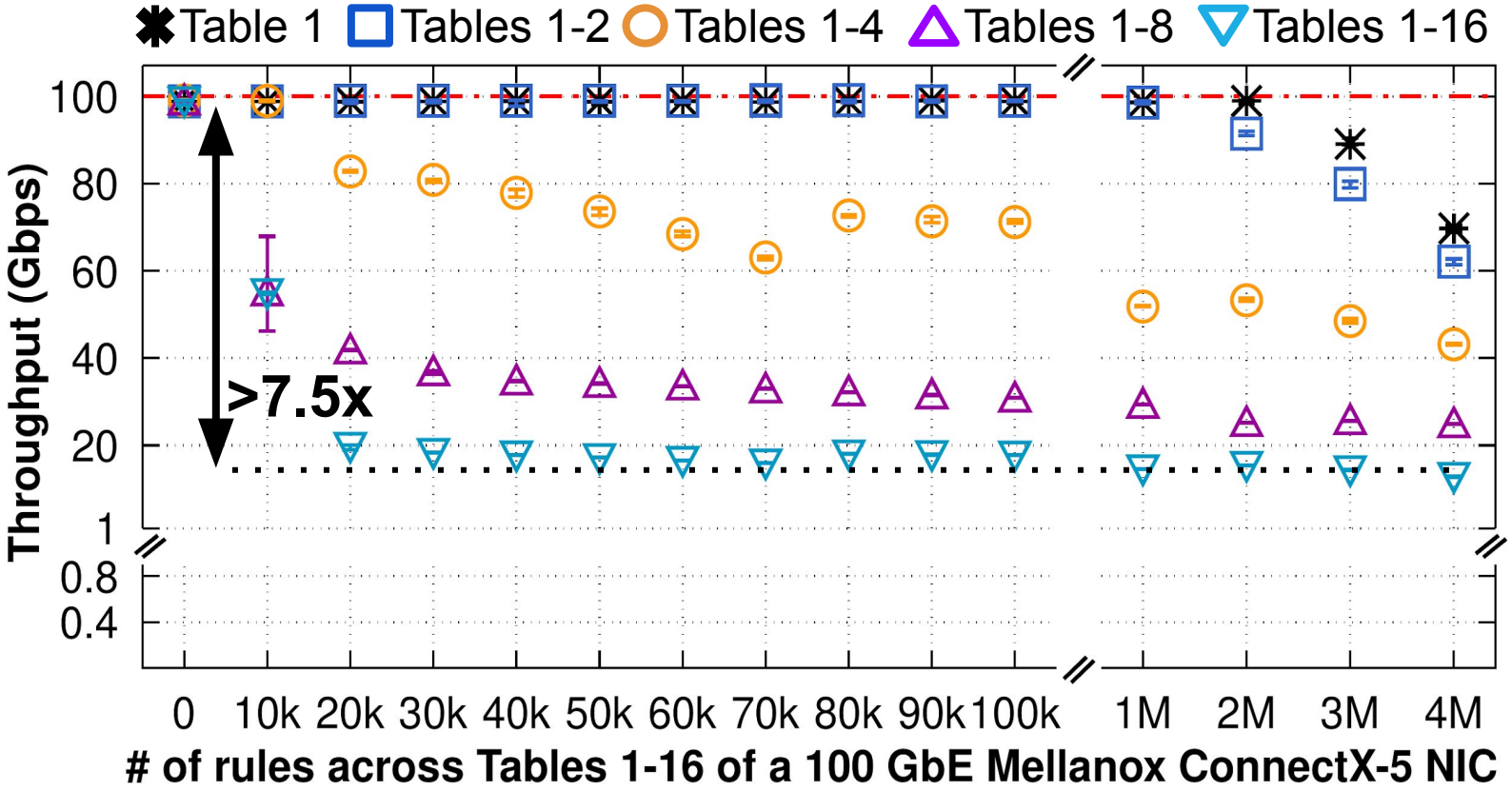
Scenario 1- Results



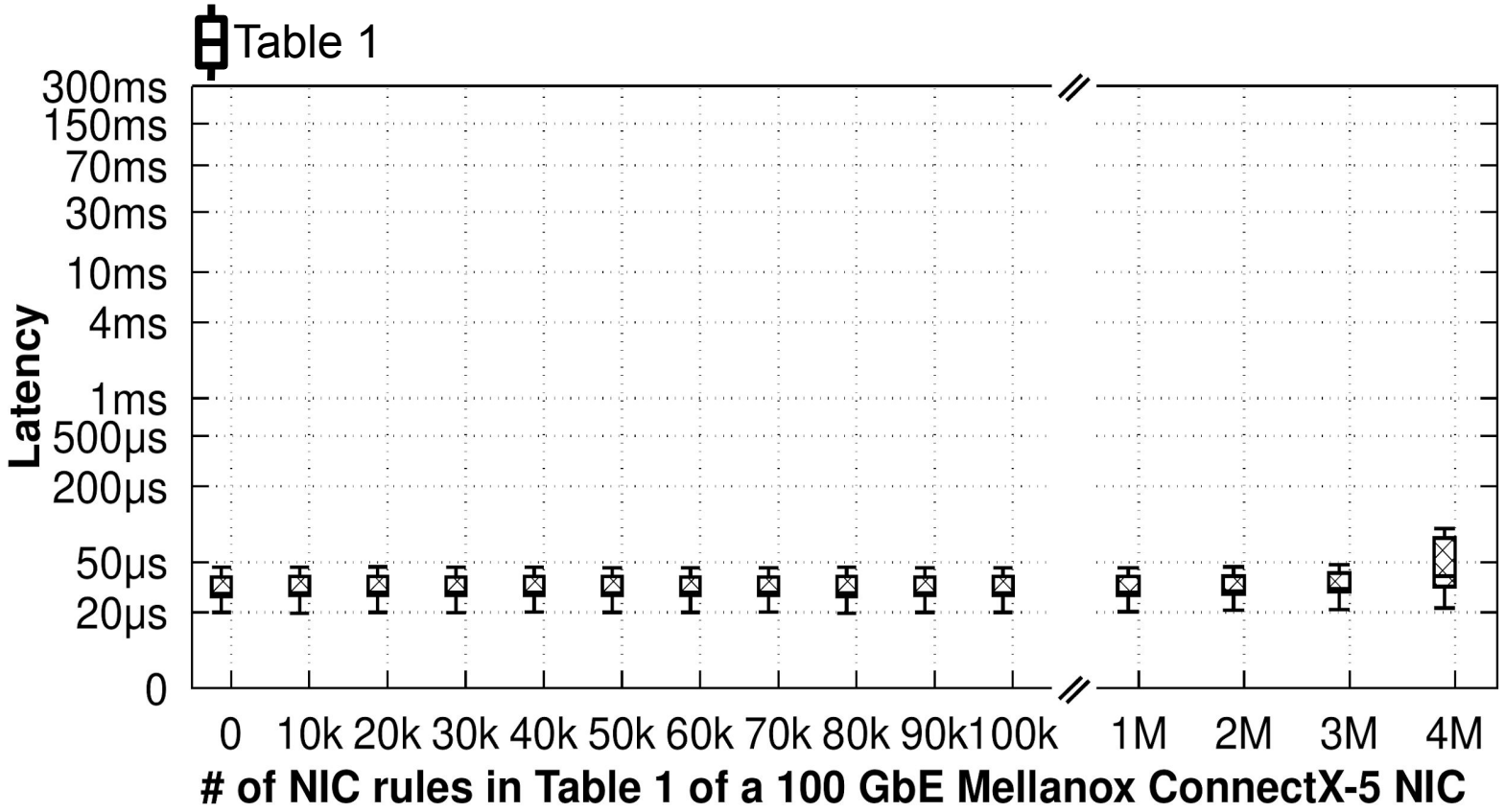
Scenario 1- Results



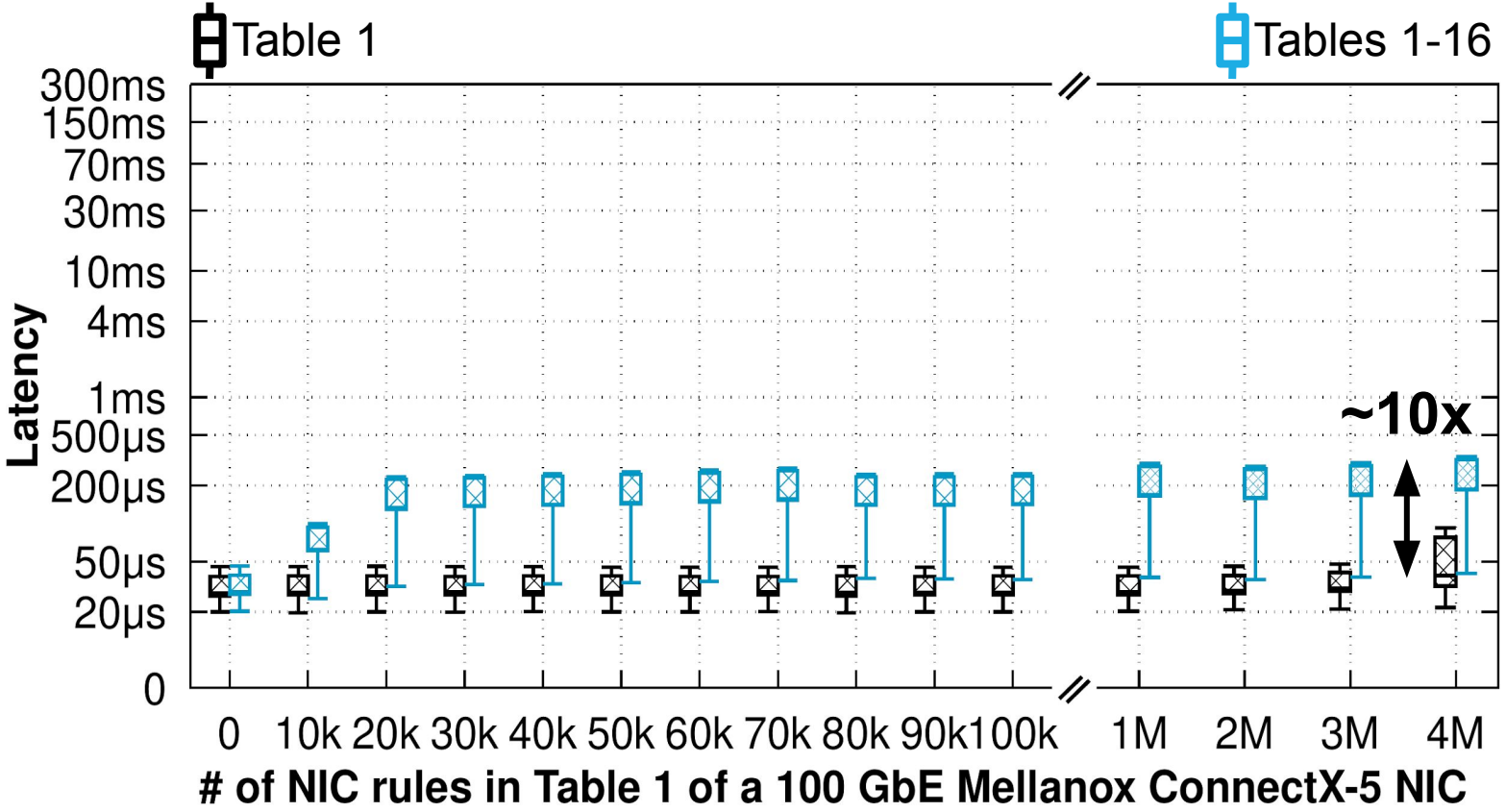
Scenario 1- Results



Scenario 1- Results



Scenario 1- Results



Scenario 1 - Finding

Finding

- Uniformly spreading rules across a chain of NIC tables incurs performance penalty

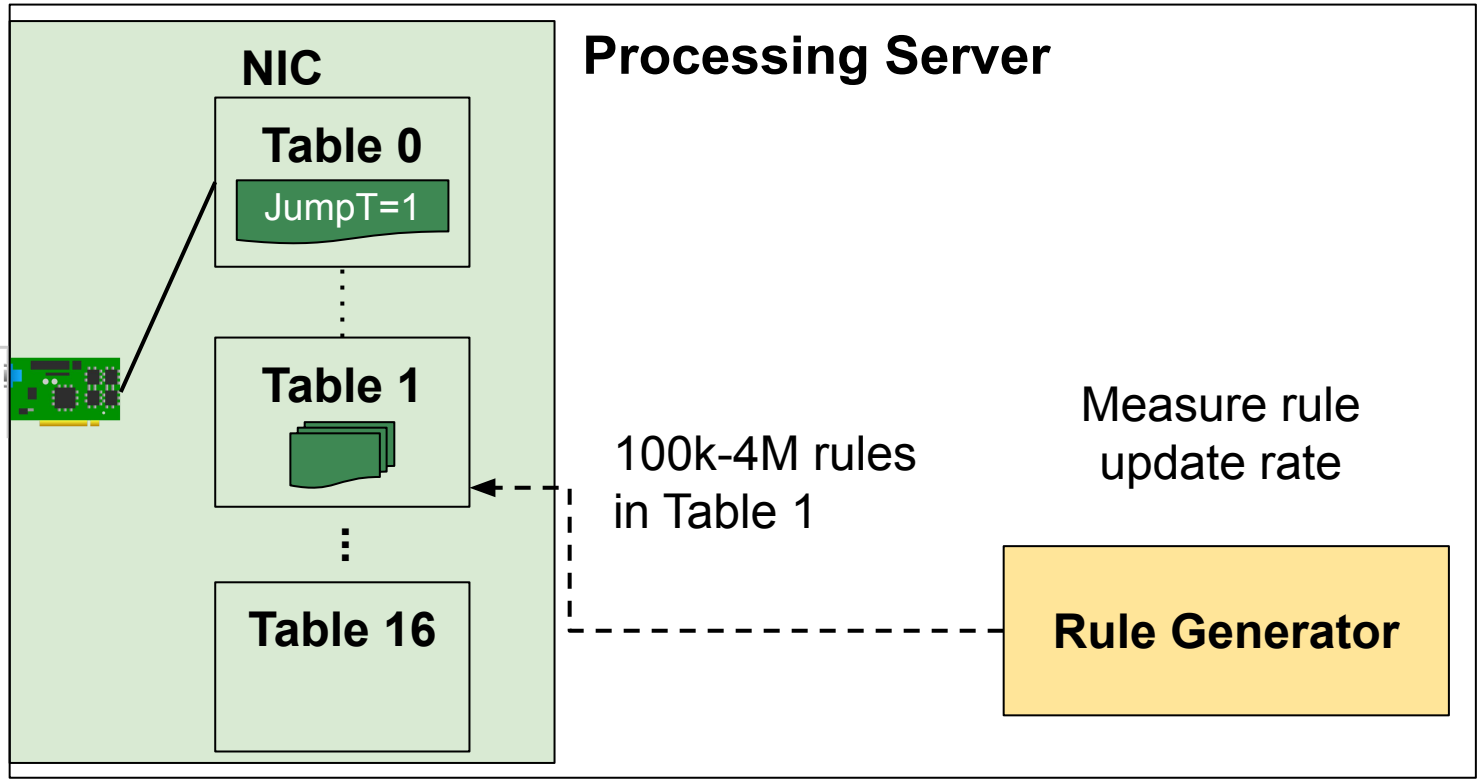
Implications

Using 16 tables:

- Throughput drops from 100 Gbps to 13 Gbps
- Latency increases by 10x

Scenario 2

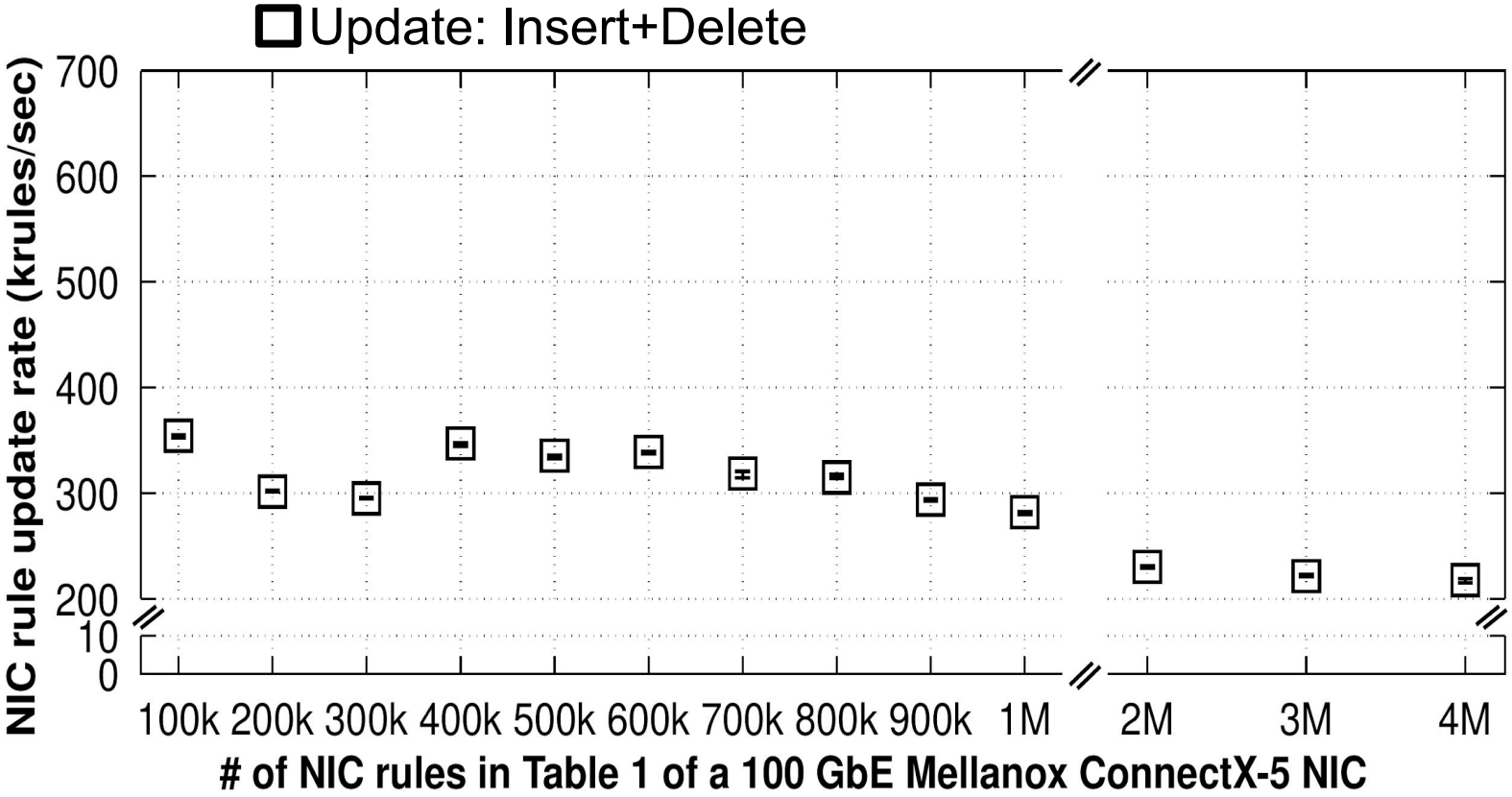
Scenario 2 - Topology



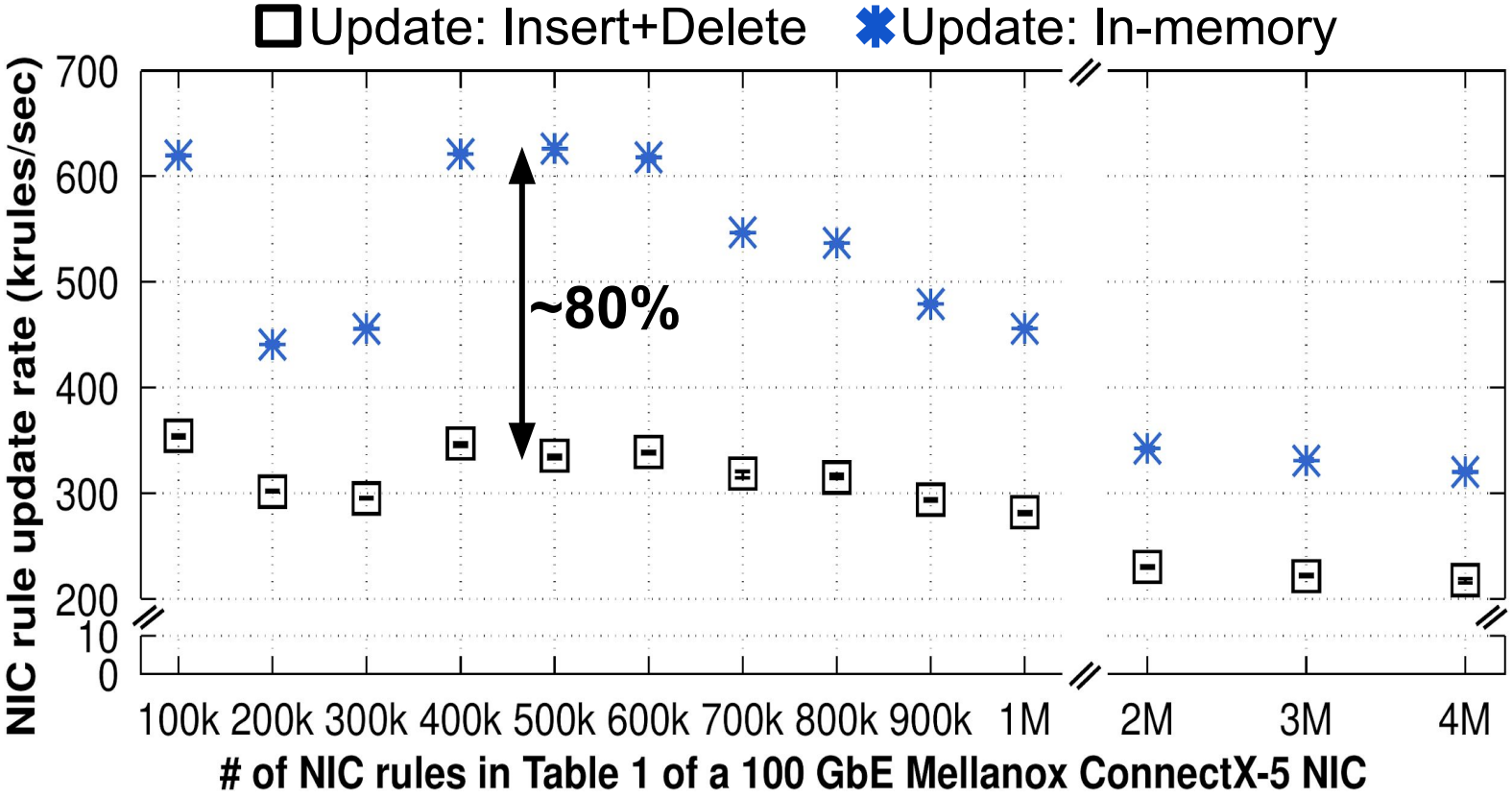
Scenario 2 - Setup

Traffic	No traffic
System	Modified version of DPDK v20.11 flow-perf
Table occupancy at the DUT	100k - 4M rules
Rules' type	Exact matches on IPv4
NIC table used	Table 1
How do we update a rule?	• Insert+Delete • Direct (**our new rte_flow_update**)
What do we measure?	Rule update rate (kFlows/sec)

Scenario 6 - Results



Scenario 6 - Results



Scenario 2 - Findings

Finding

- NIC rule update operations are **non-atomic** and rely on sequential addition and deletion

Implications

- Too slow for applications that require heavy updates
- Our dedicated update API performs up to 80% faster

More findings in the paper

- Batch or periodic rule updates cause substantial throughput degradation, while the NIC is processing traffic
- Some rule types are installed more quickly than others, depending on the part of the pipeline being used

Conclusion

- Today's NIC classifiers achieve high performance, but not under all circumstances:
  - ➔ We have uncovered several limitations that reduce NIC performance, depending on where/what rules are installed
  - ➔ Packet processing performance drops while updating the classifier
- New direct rule update API for DPDK-based Mellanox NICs

NIC-Bench PAM'21

What you need to know about (Smart) Network Interface Cards
Open source code and results with NVIDIA ConnectX-4, ConnectX-5, ConnectX-6, and Bluefield NICs (here)

Thank you!

More findings in the paper

F1	NIC classification performance drops with an increasing number of tables
F2	Batch or periodic rule updates cause substantial throughput degradation, while the NIC is processing traffic
F3	Some rule types are installed more quickly than others, depending on the part of the pipeline being used
F4	NIC rule update operations are non-atomic; instead, they rely on sequential addition and deletion operations

Conclusion

- Today's NIC classifiers achieve high link speeds, but also exhibit important limitations:
  - Performance across tables is not the same
  - Using an increasing # of tables incurs performance penalty
  - Packet processing performance drops while updating the classifier
- A direct rule update is introduced to perform quick (80% faster) data plane state modifications

NIC-Bench PAM'21

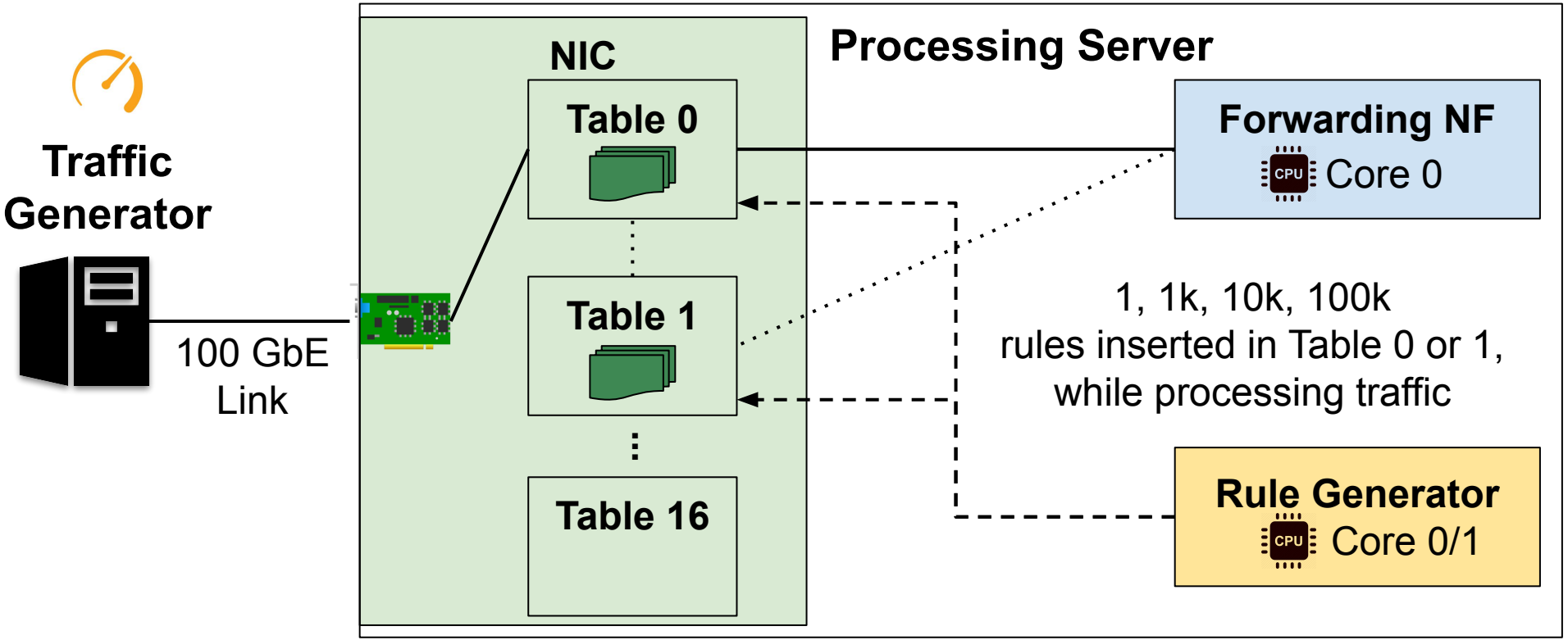
What you need to know about (Smart) Network Interface Cards
Open source code and results with NVIDIA ConnectX-4, ConnectX-5, ConnectX-6, and Bluefield NICs (here)

Thank you!

Back-Up

Scenario 3

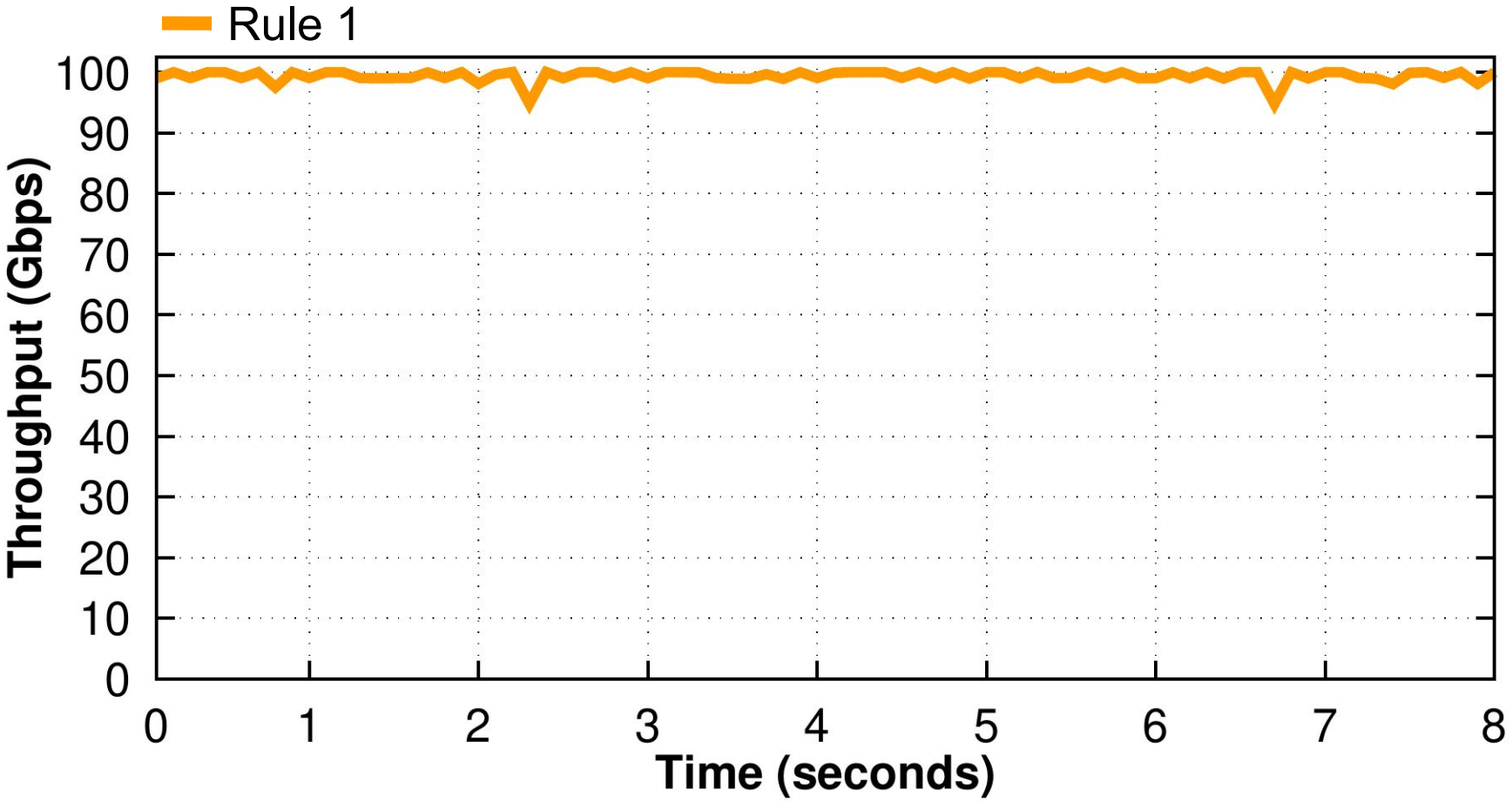
Scenario 3 - Topology



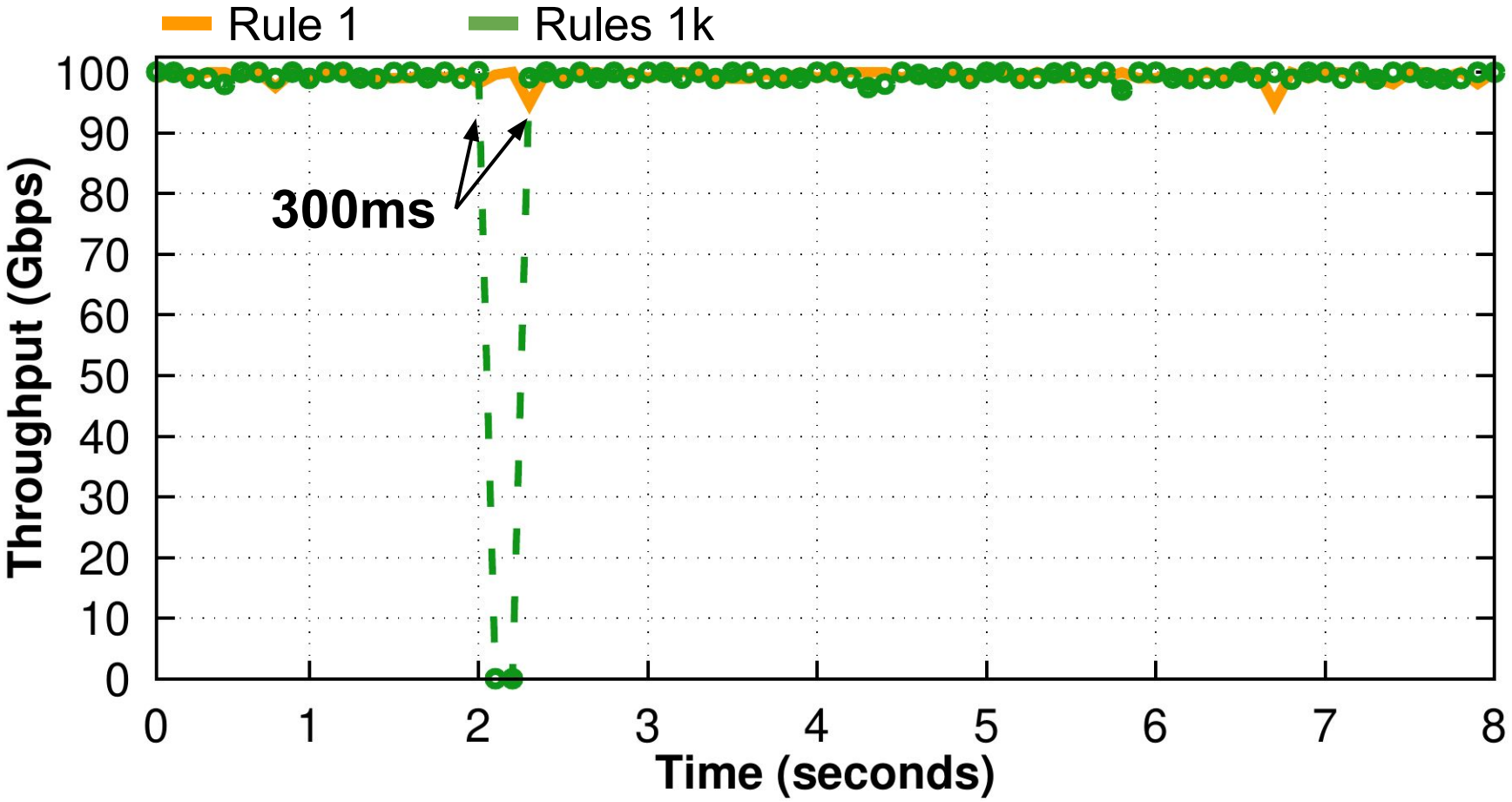
Scenario 3 - Traffic Characteristics

Flows	10k - Proto **UDP**
Rate	**100 Gbps**
Frame length	1500 bytes
Ruleset sizes	1, 1k, 10k, 100k rules
How many rules match traffic?	**1 rule pre-installed**
NIC tables used	**Tables 0 and 1**
Who installs the rules at the DUT?	Core 0 or Core 1
Who processes traffic?	Core 0
What do we measure?	Throughput

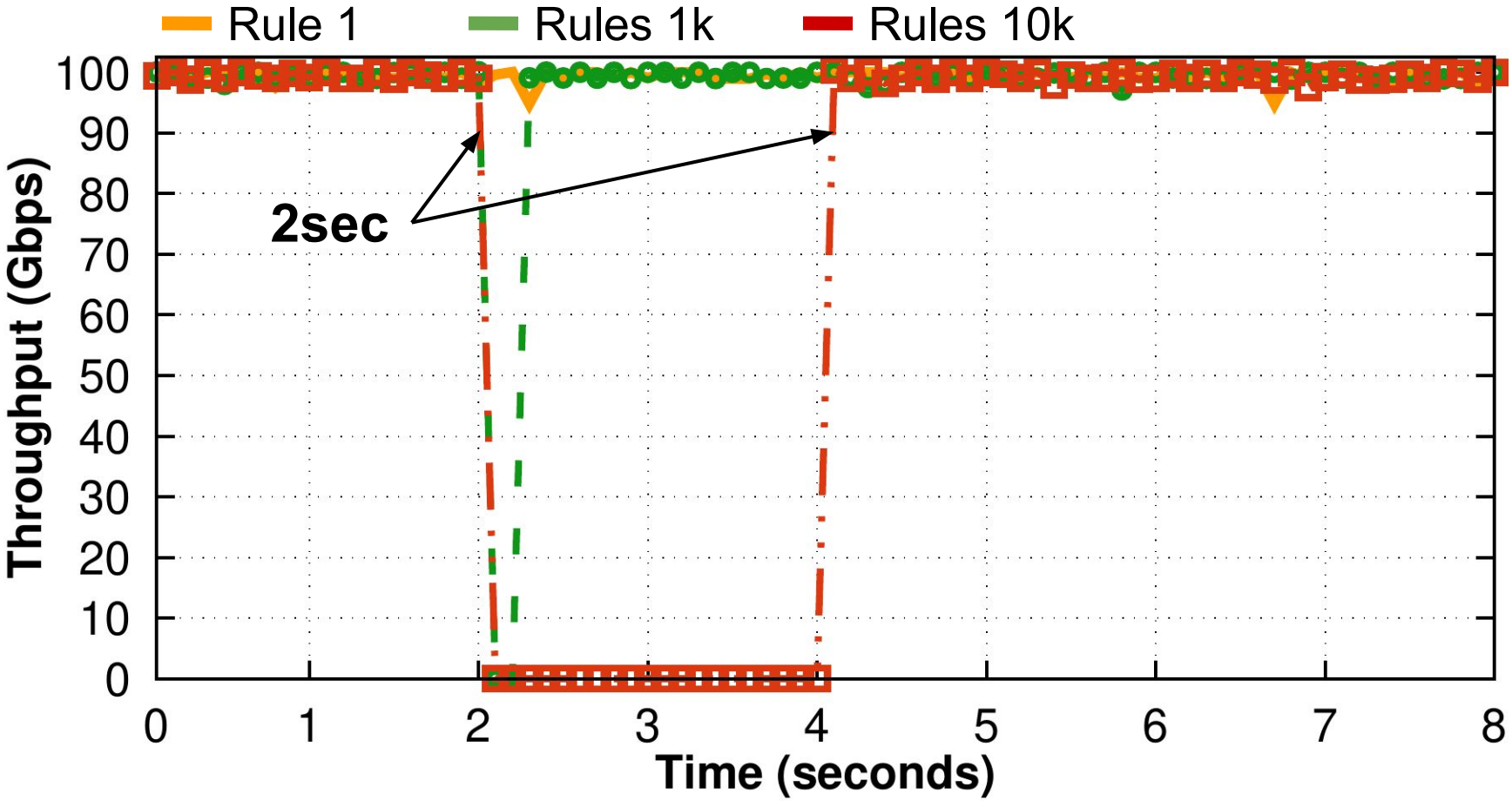
Scenario 3 - Results



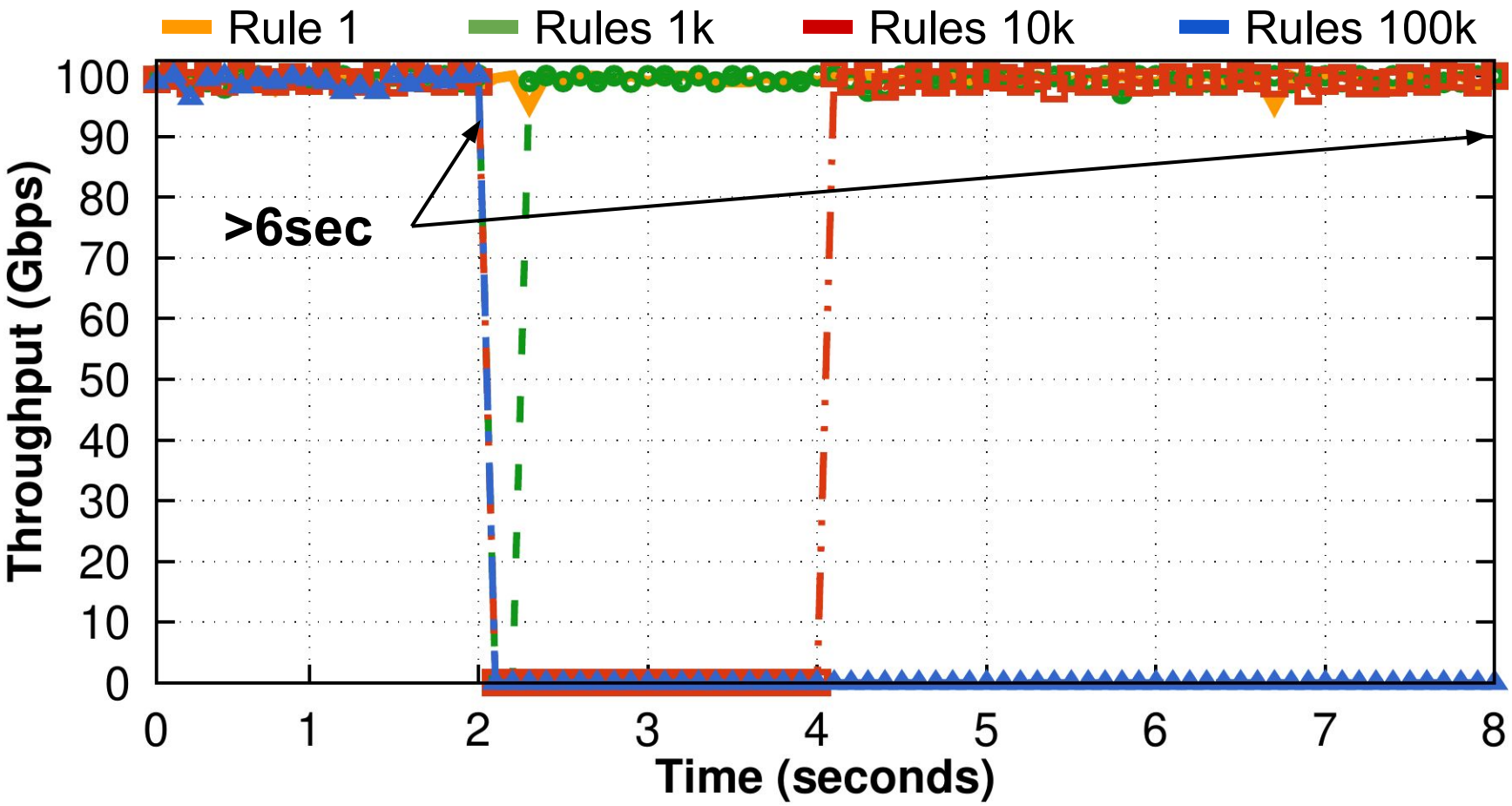
Scenario 3 - Results



Scenario 3 - Results



Scenario 3 - Results



Scenario 3 - Findings

Finding 3

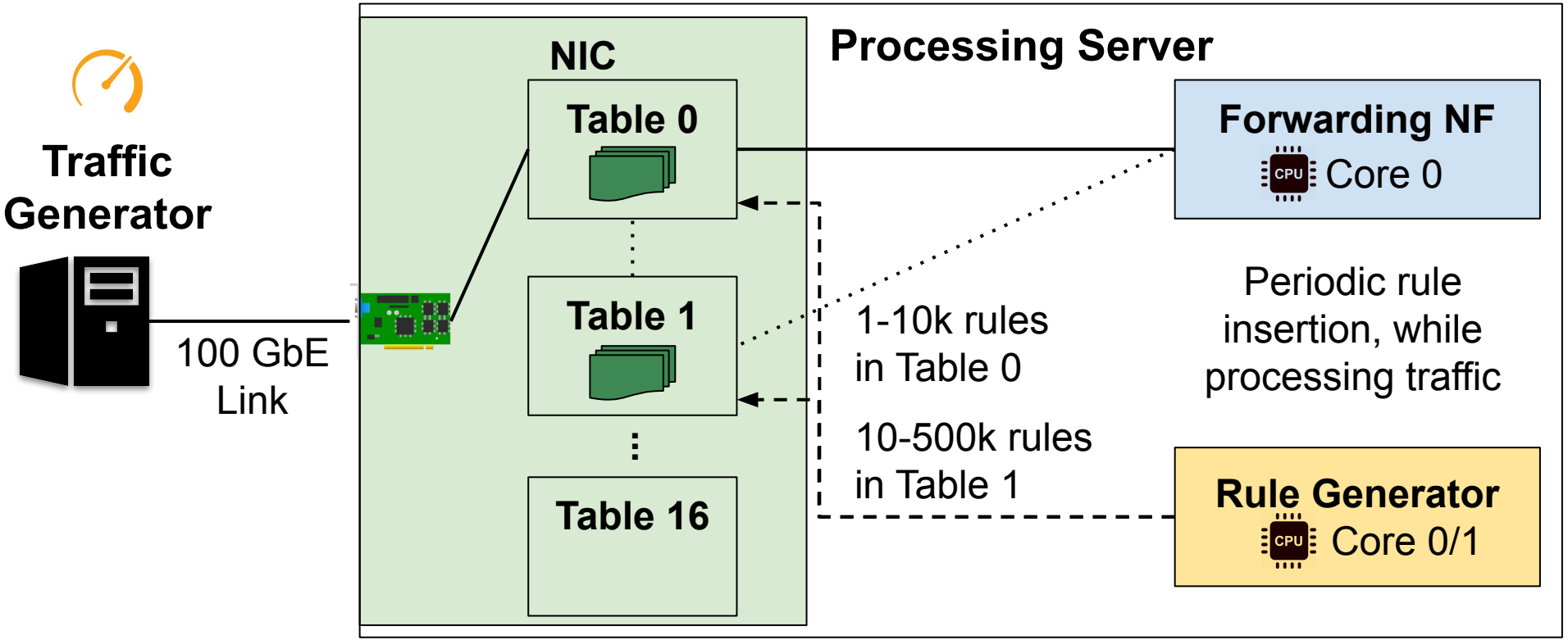
- A batch update of the NIC classifier,while processing traffic, makes the NIC temporarily unavailable

Implications

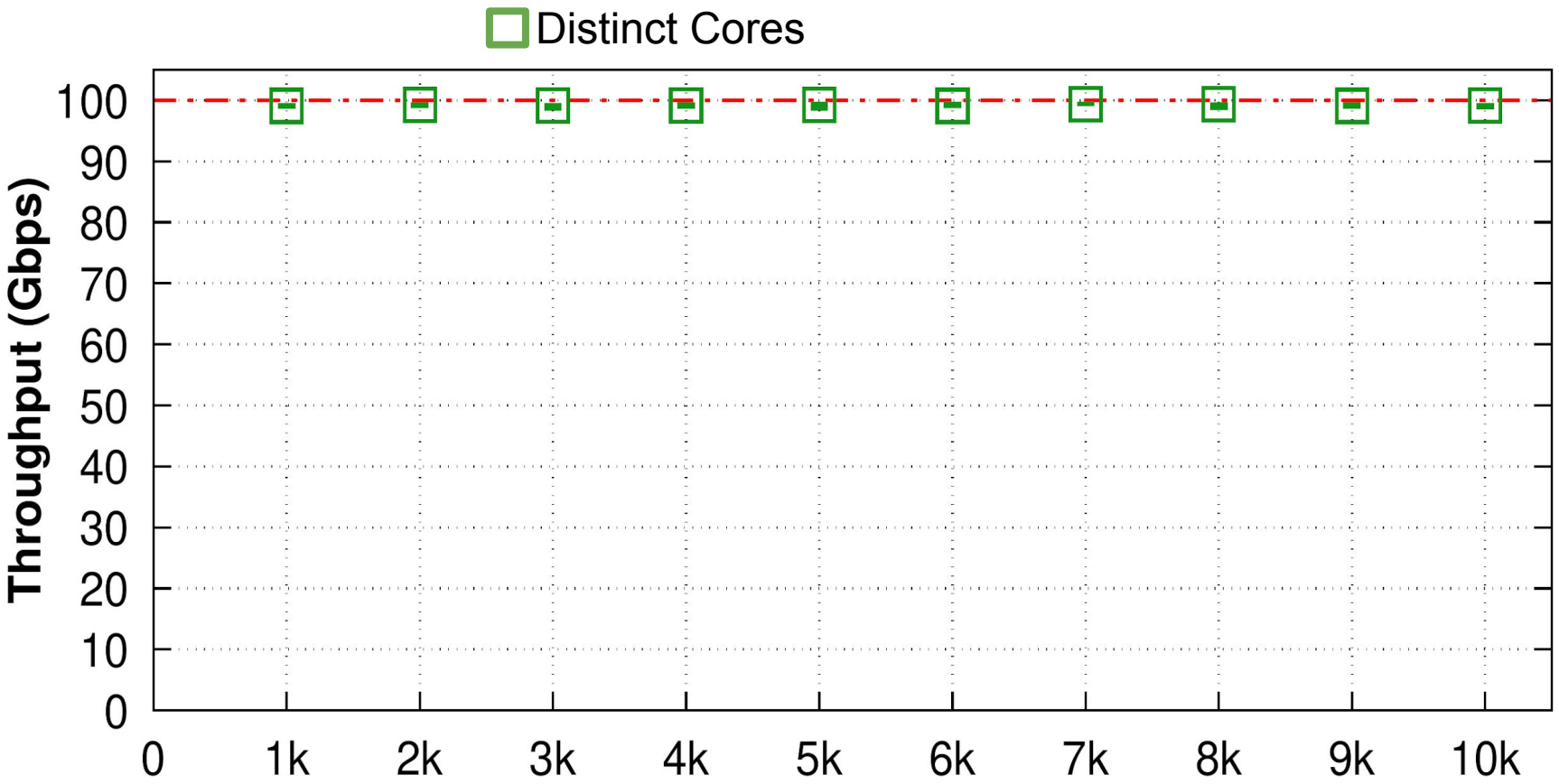
- Table 0: 100% packet loss for up to several seconds
- Table 1: 100% packet loss for a few hundreds of milliseconds

Scenario 4

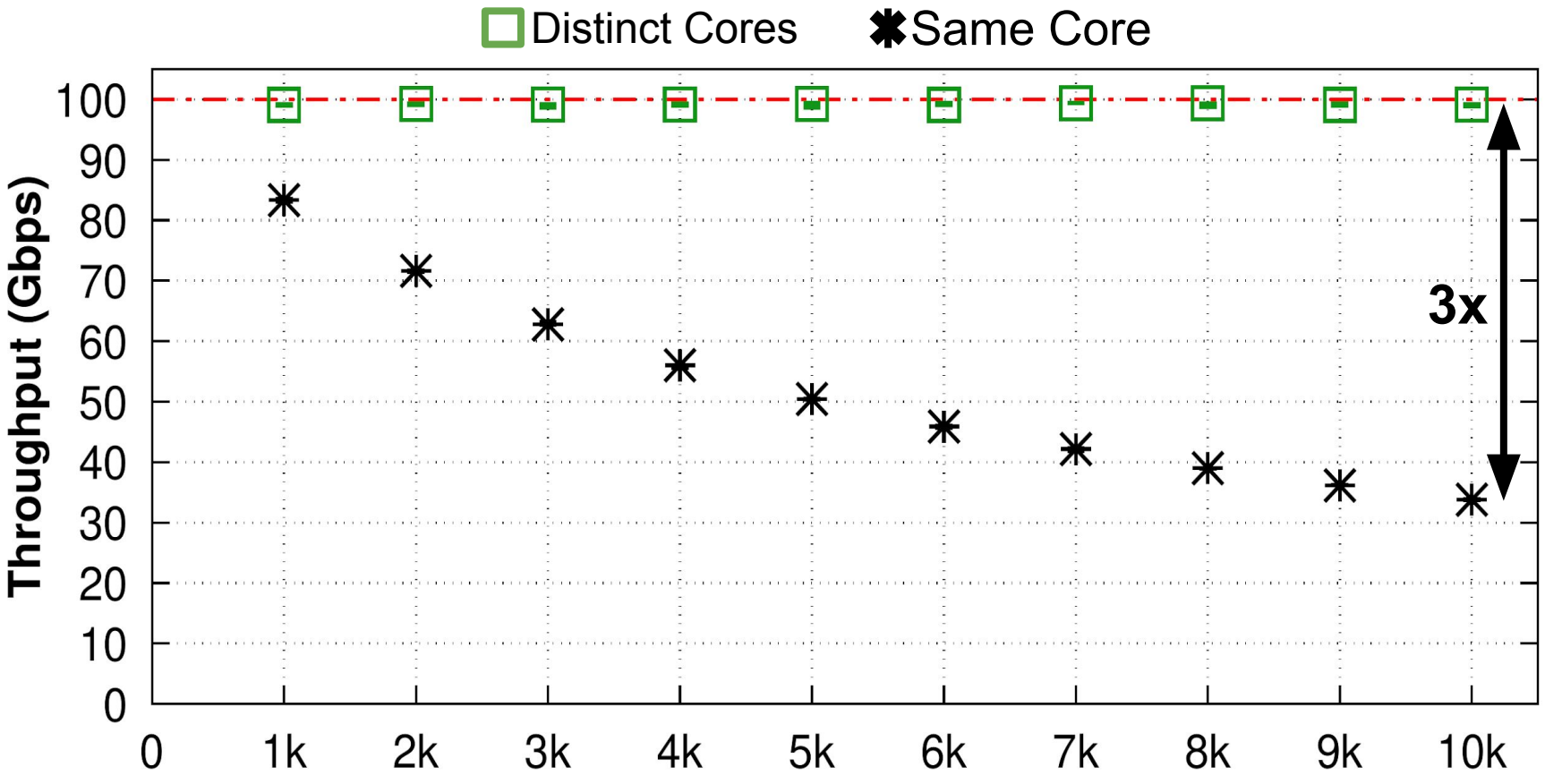
Scenario 4 - Topology



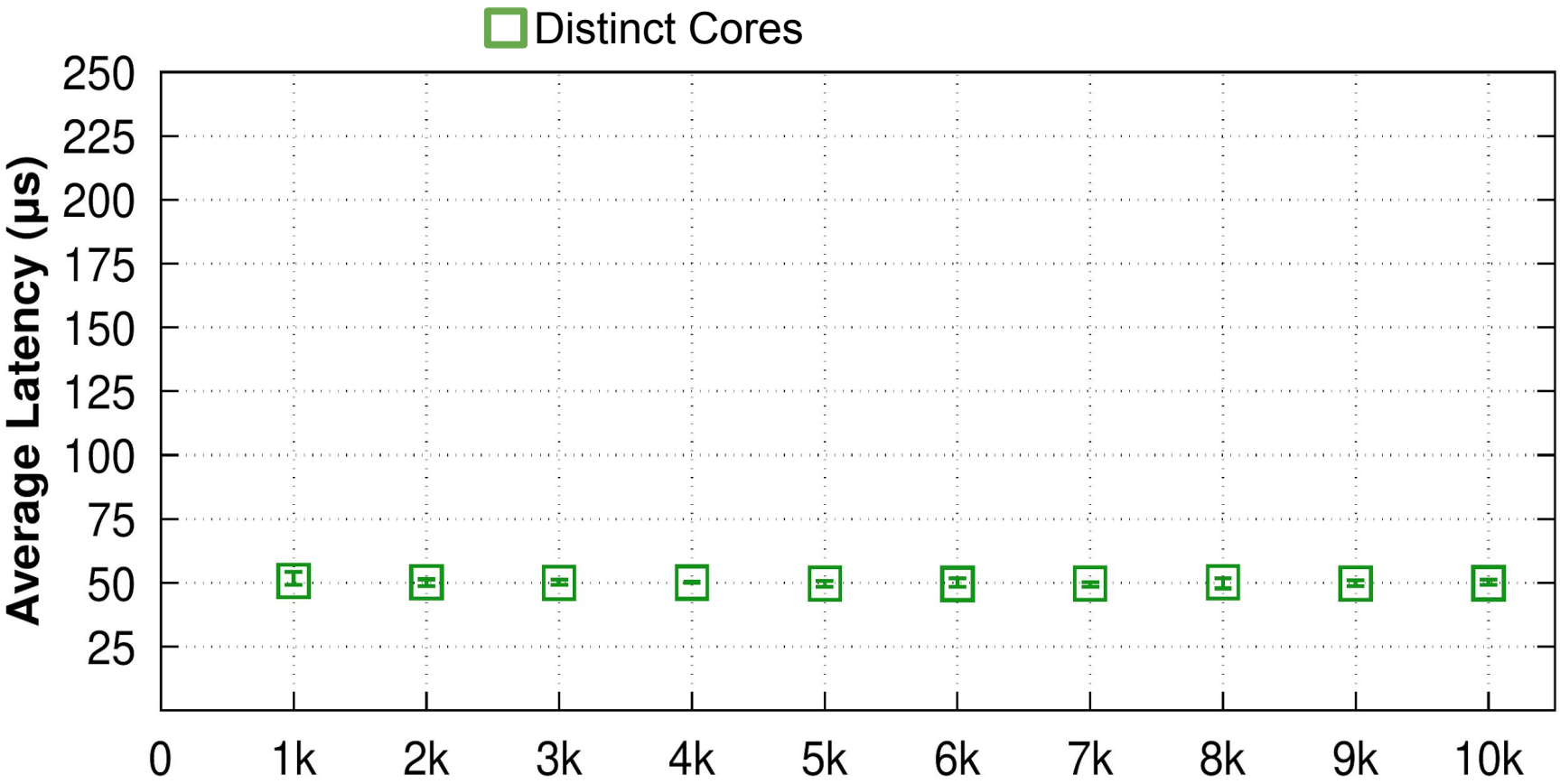
Scenario 4 - Results



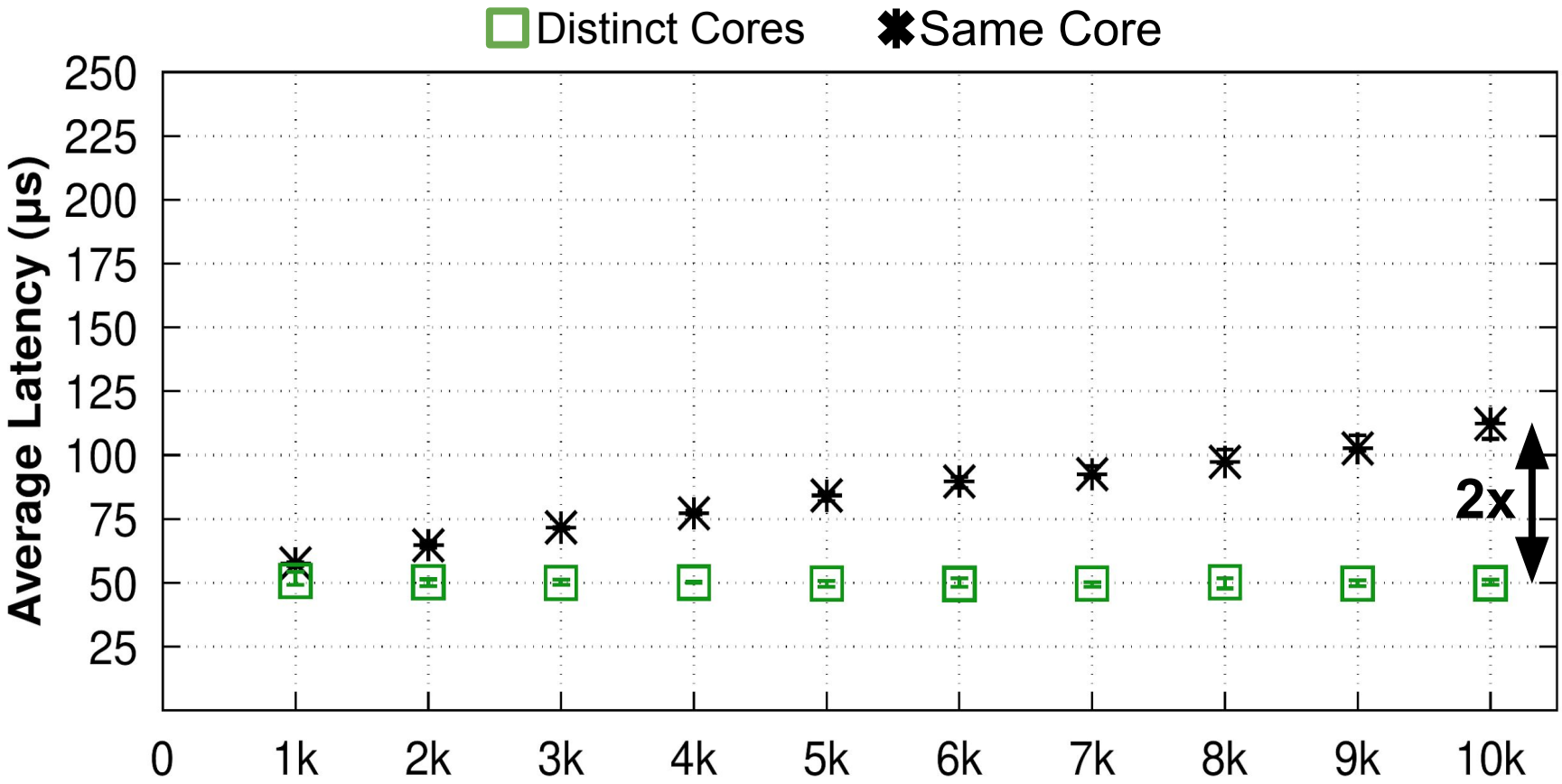
Scenario 4 - Results



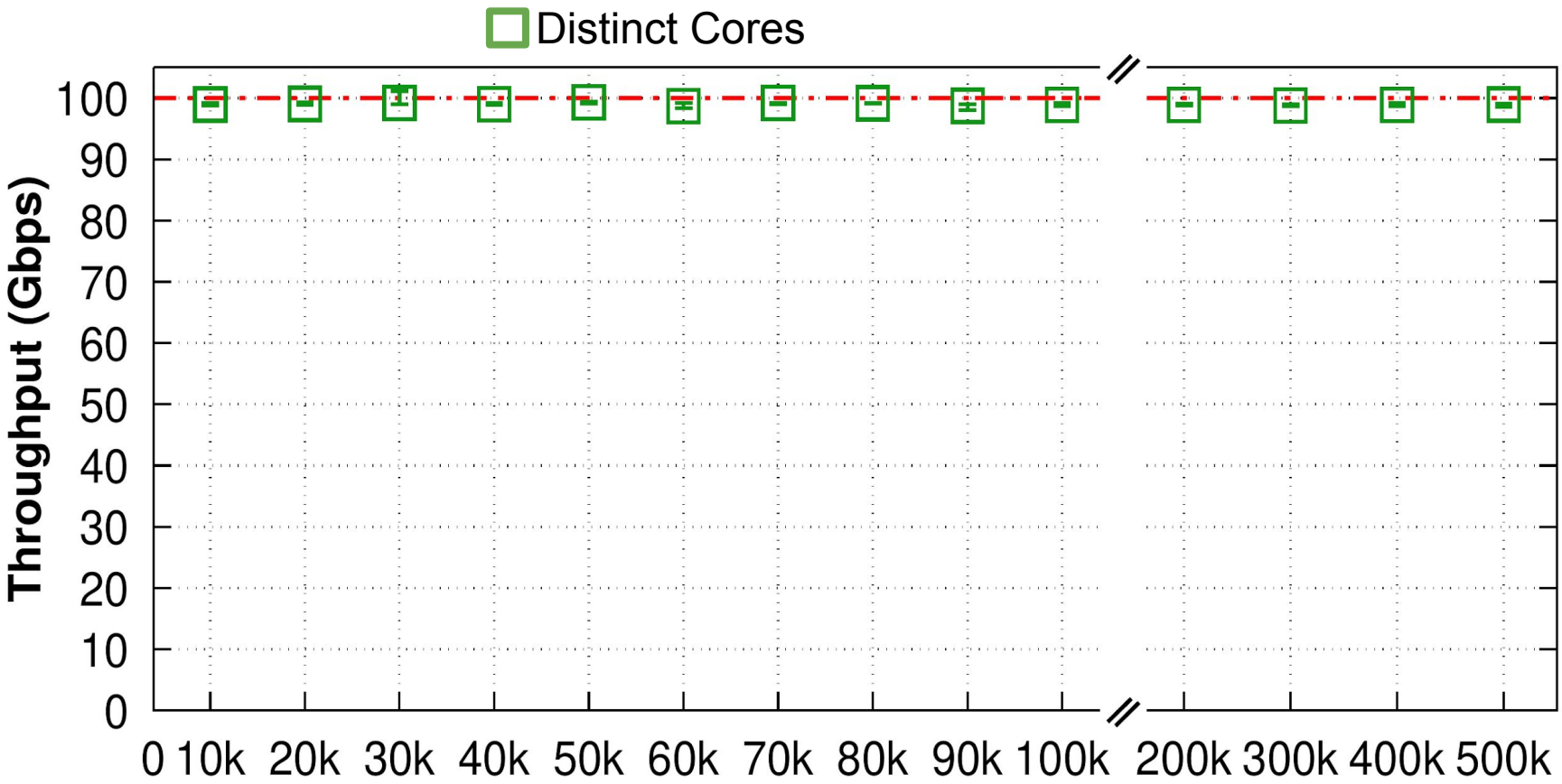
Scenario 4 - Results



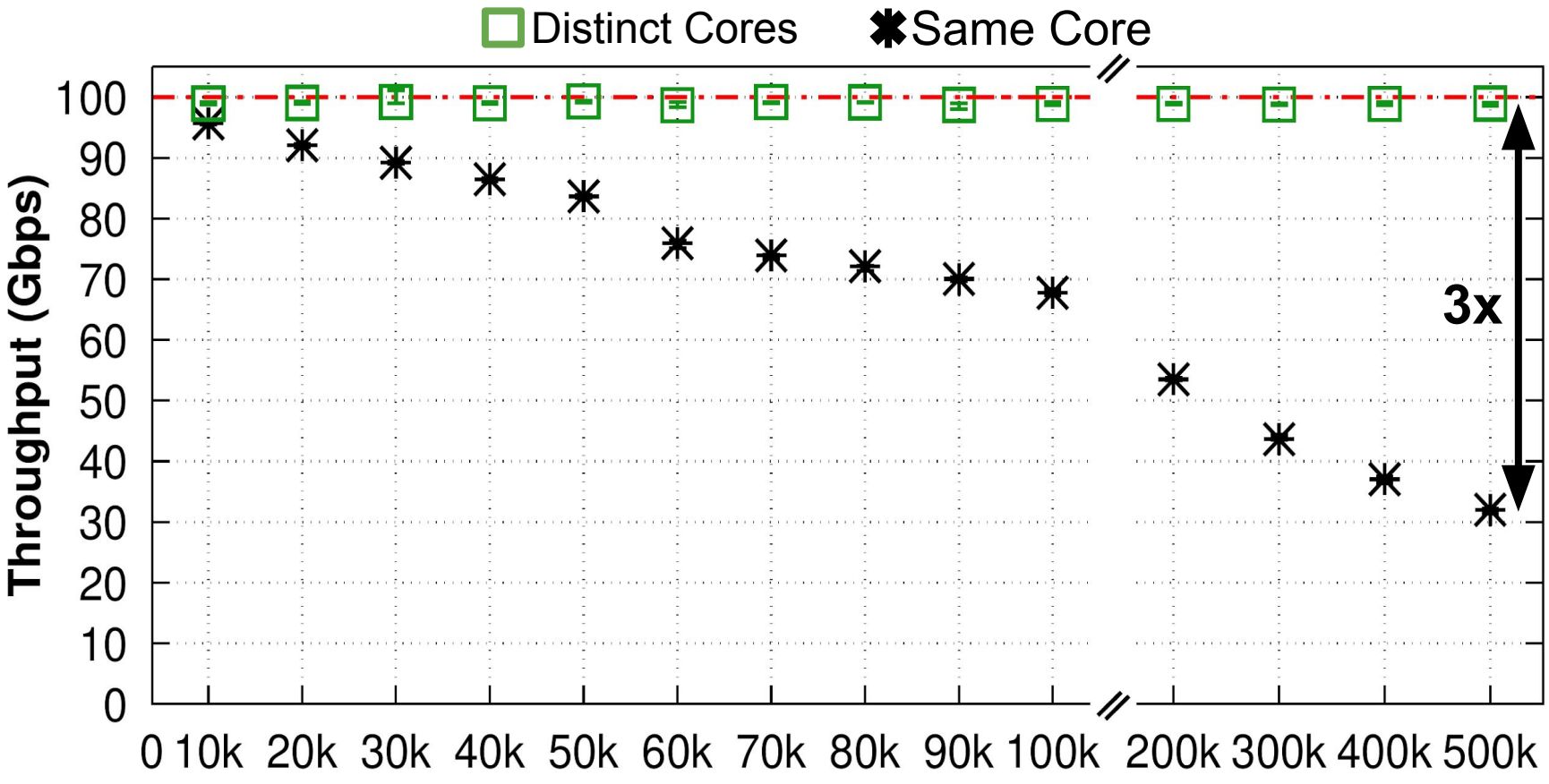
Scenario 4 - Results



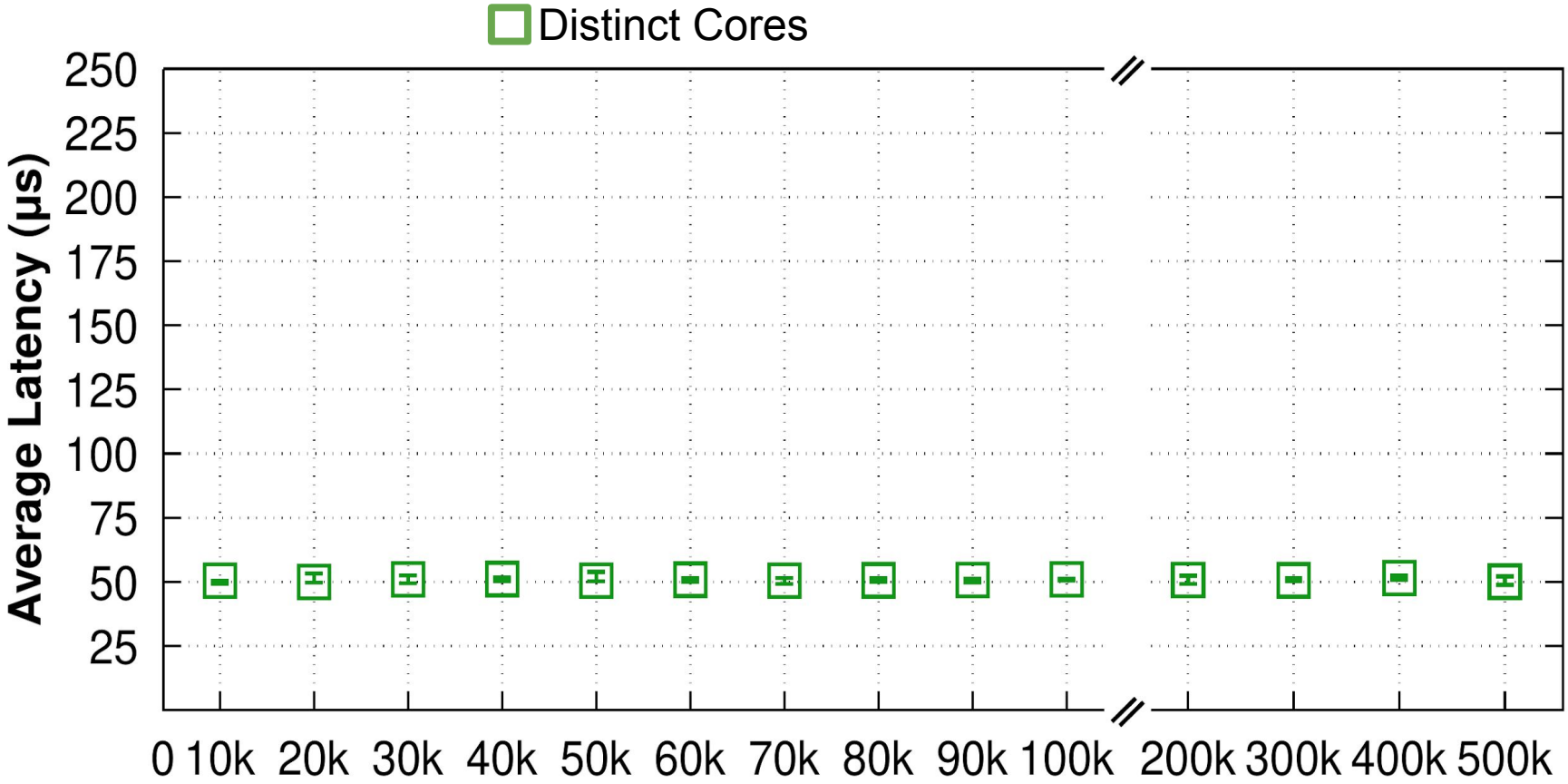
Scenario 4 - Results



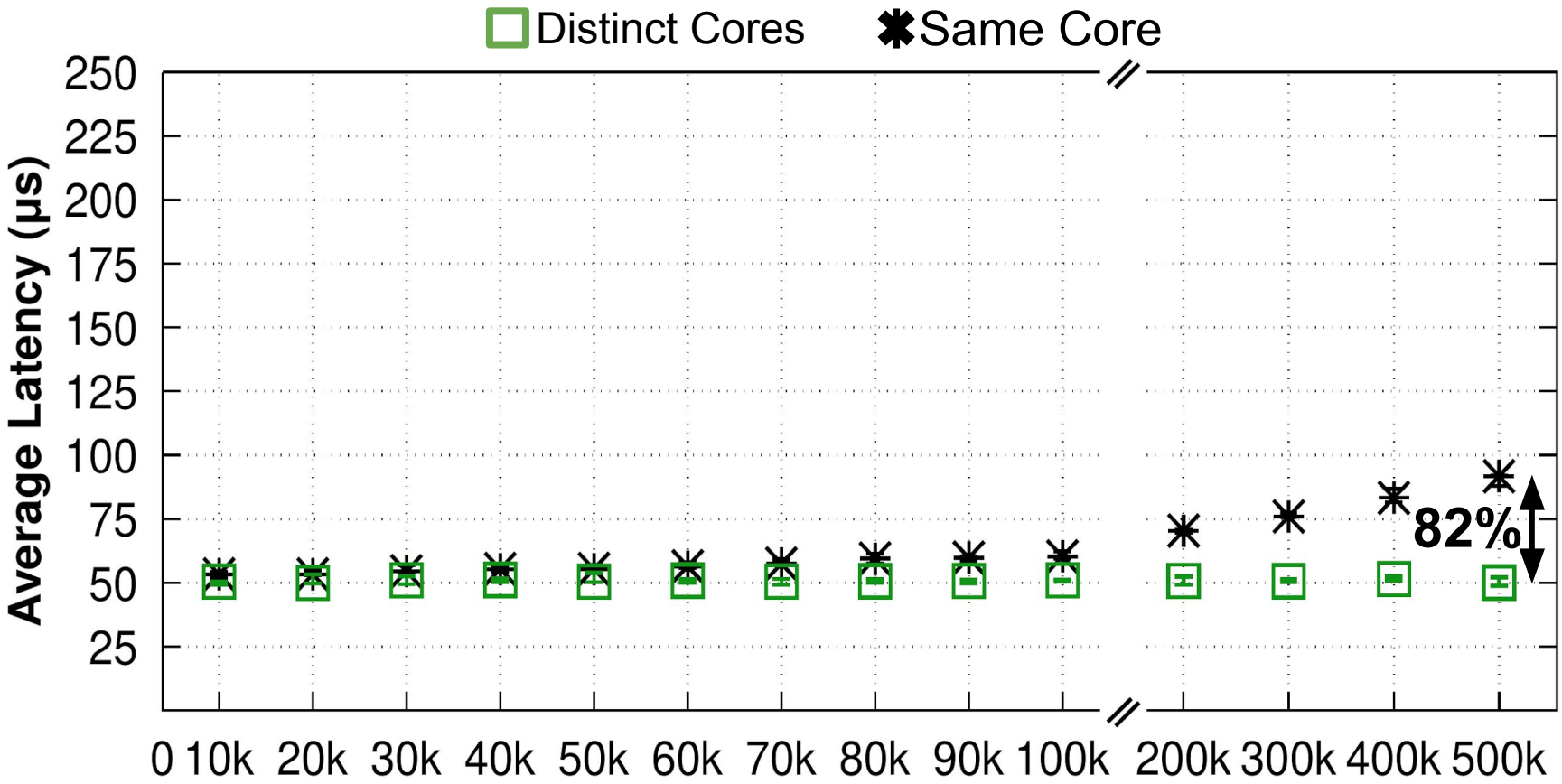
Scenario 4 - Results



Scenario 4 - Results



Scenario 4 - Results



Scenario 4 - Findings

Finding 4

- Frequent updates of the NIC classifier,while processing traffic, causes substantial performance degradation

Finding 5

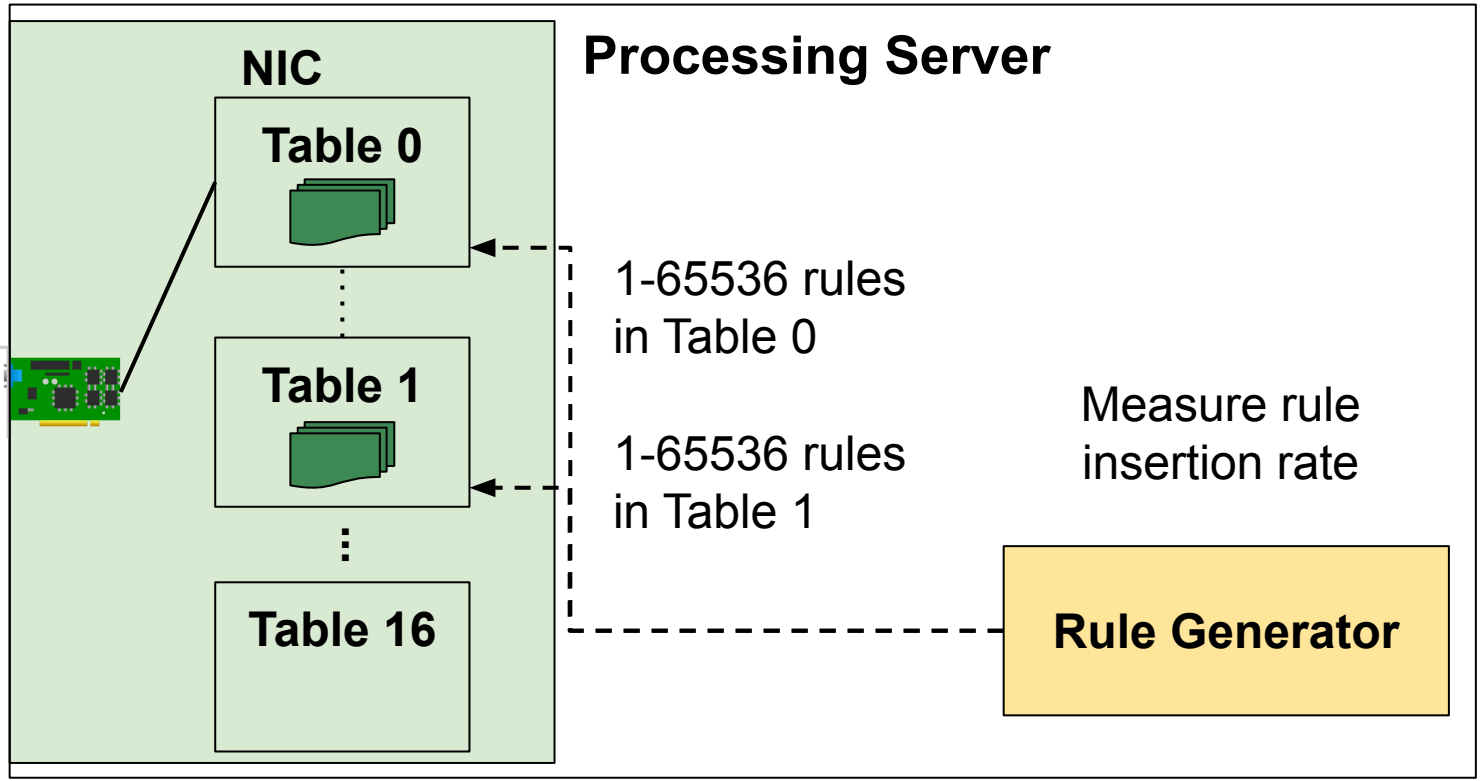
- Updating the NIC classifier from a separate core does not degrade its performance

Implications

- Throughput drops from 100 Gbps to 30 Gbps
- Latency increases by 2x

Scenario 5

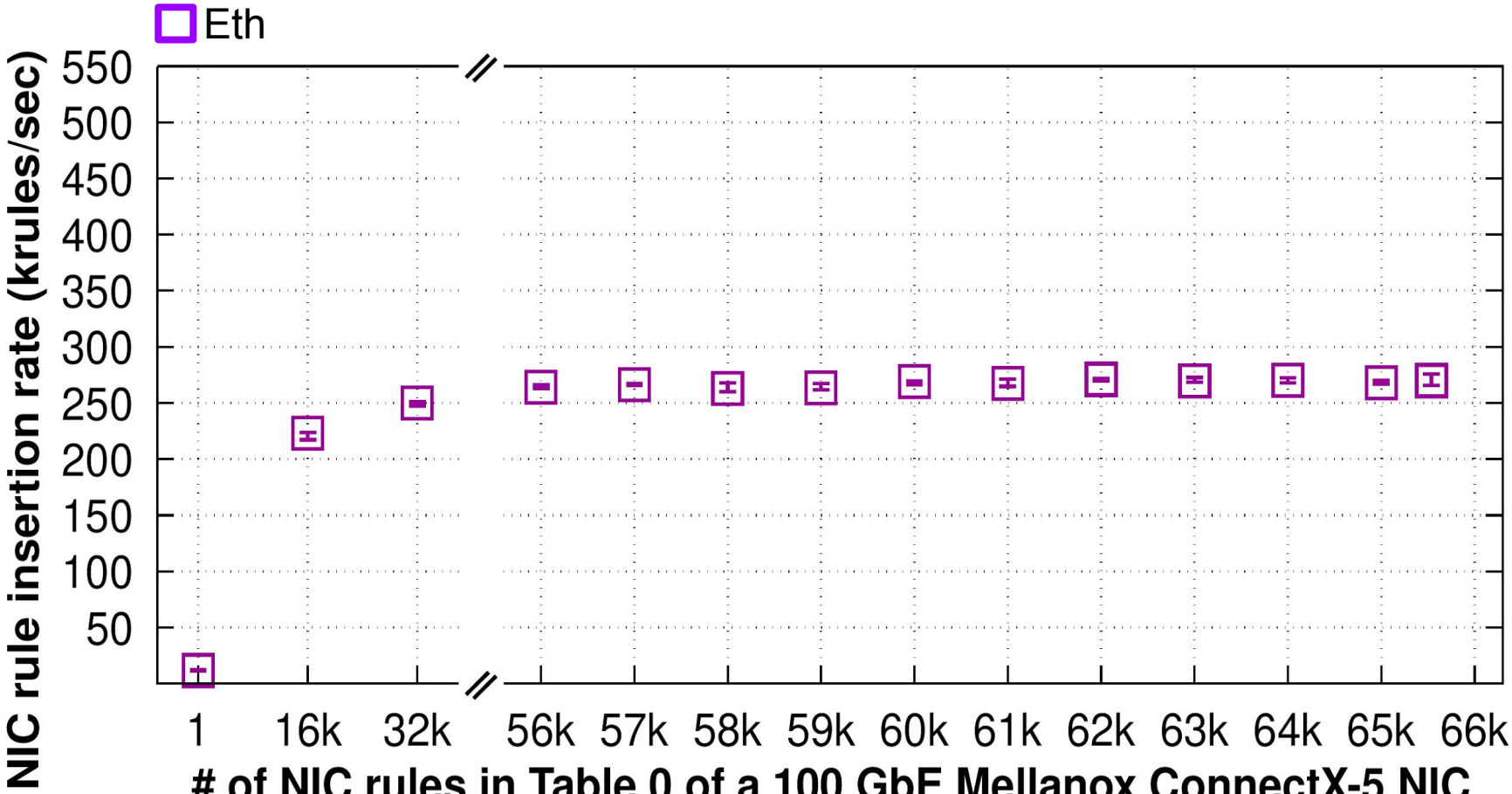
Scenario 5 - Topology



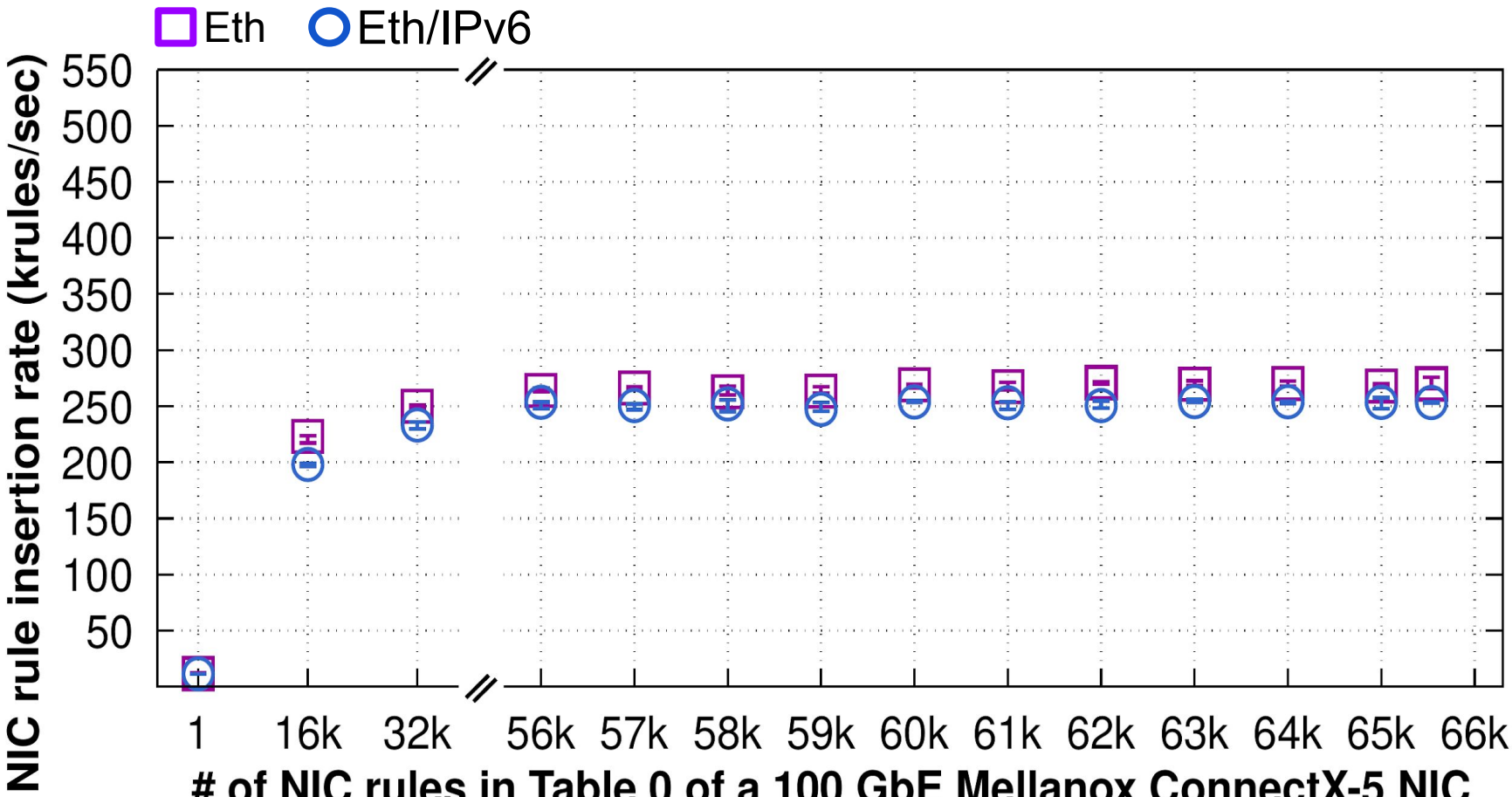
Scenario 5 - Setup

Traffic	No traffic
System	DPDK v20.11 flow-perf tool
Initial table occupancy at the DUT	0 rules
Final table occupancy at the DUT	1-65536 rules
Rules' type	Exact matches
NIC tables used	Table 0 or Table 1
What do we measure?	Rule insertion rate

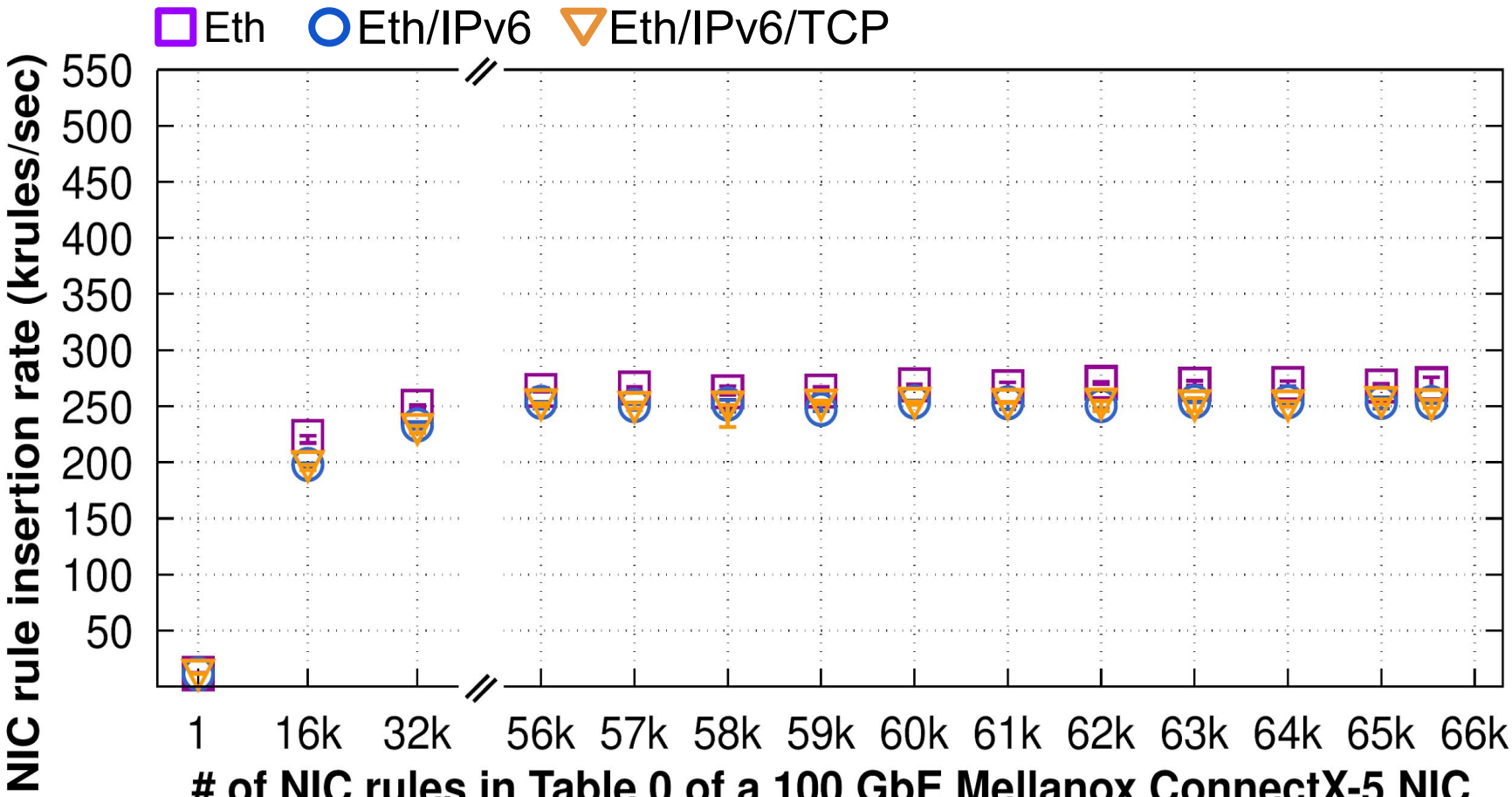
Scenario 5 - Results



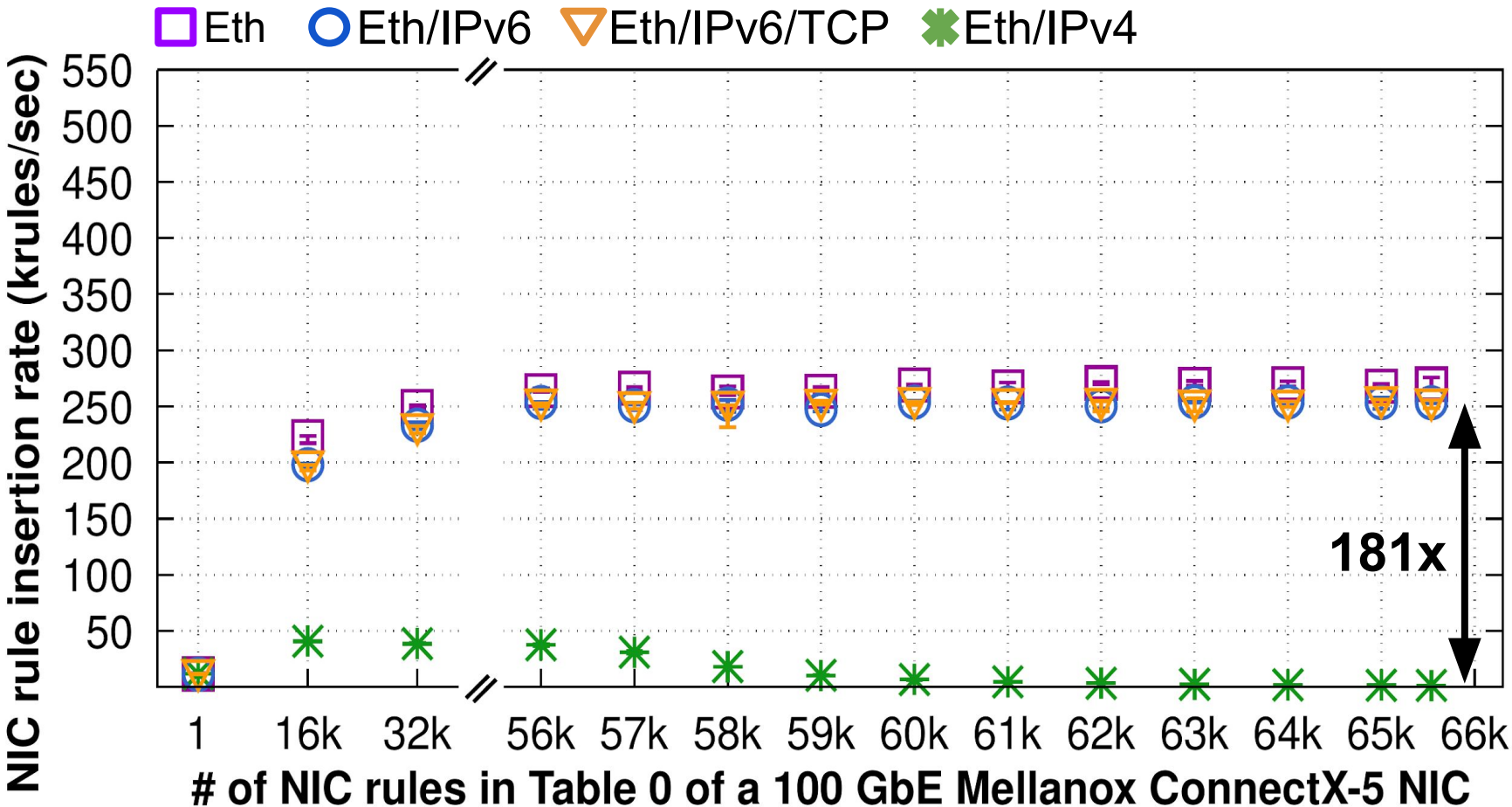
Scenario 5 - Results



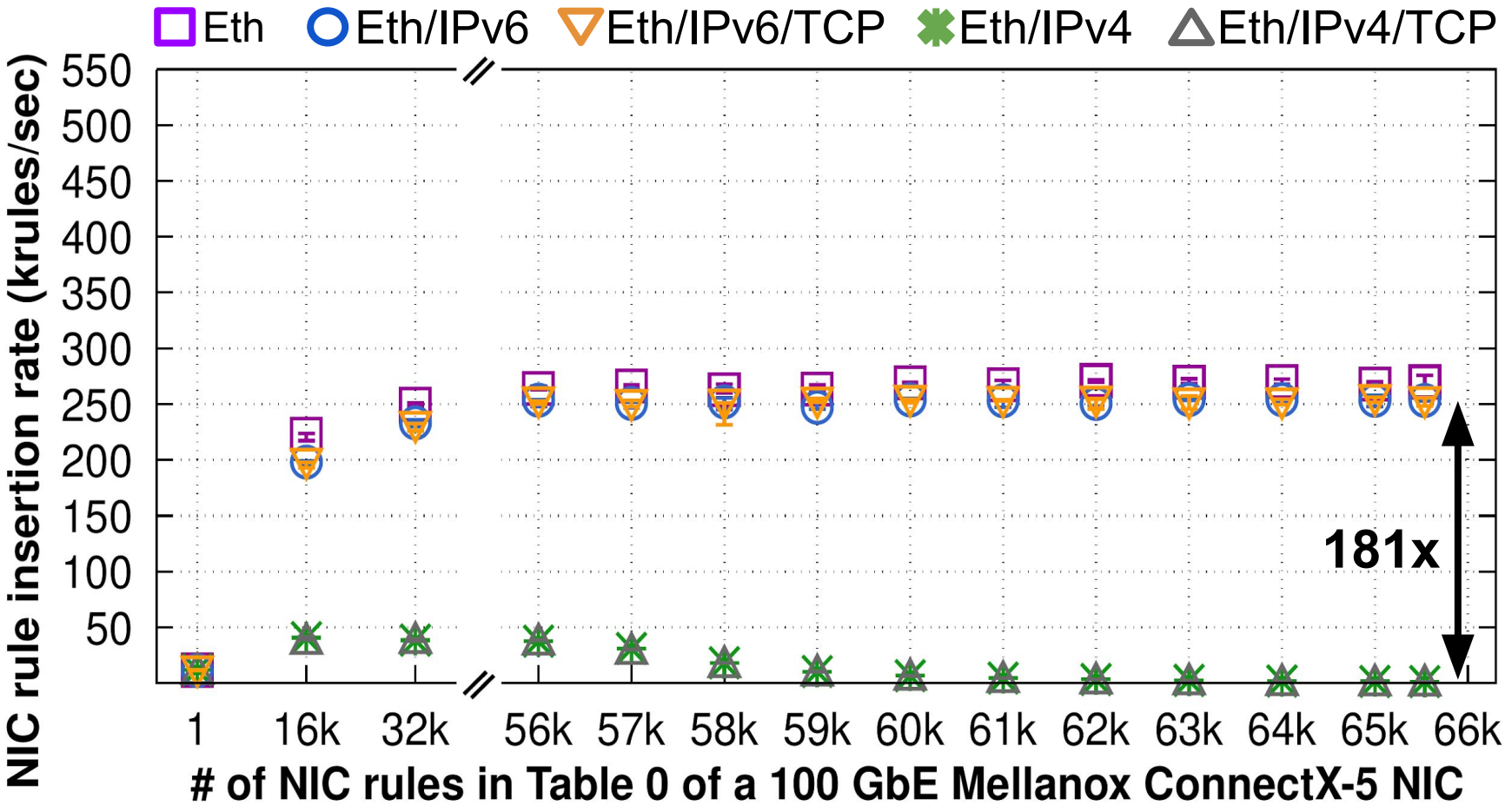
Scenario 5 - Results



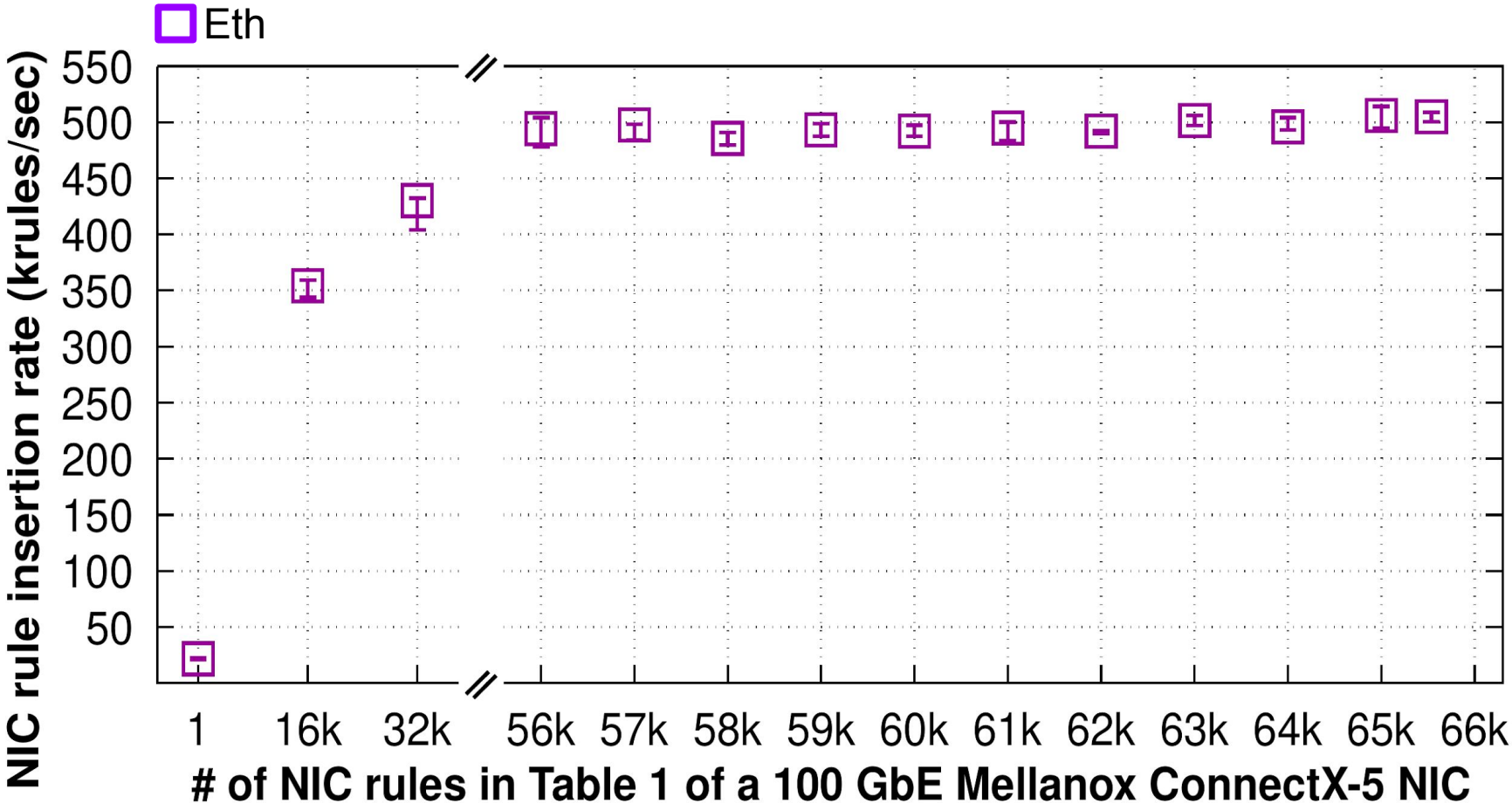
Scenario 5 - Results



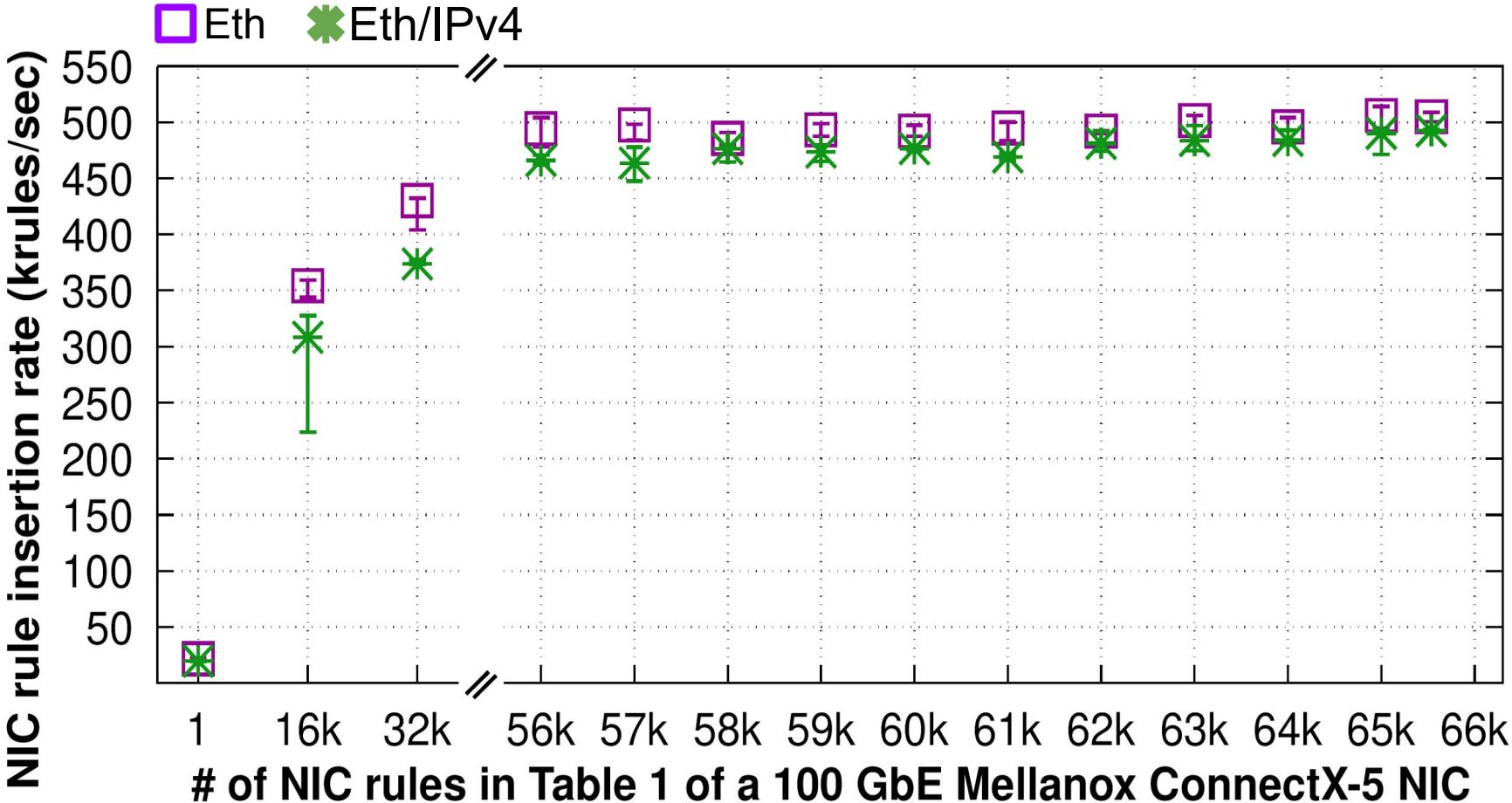
Scenario 5 - Results



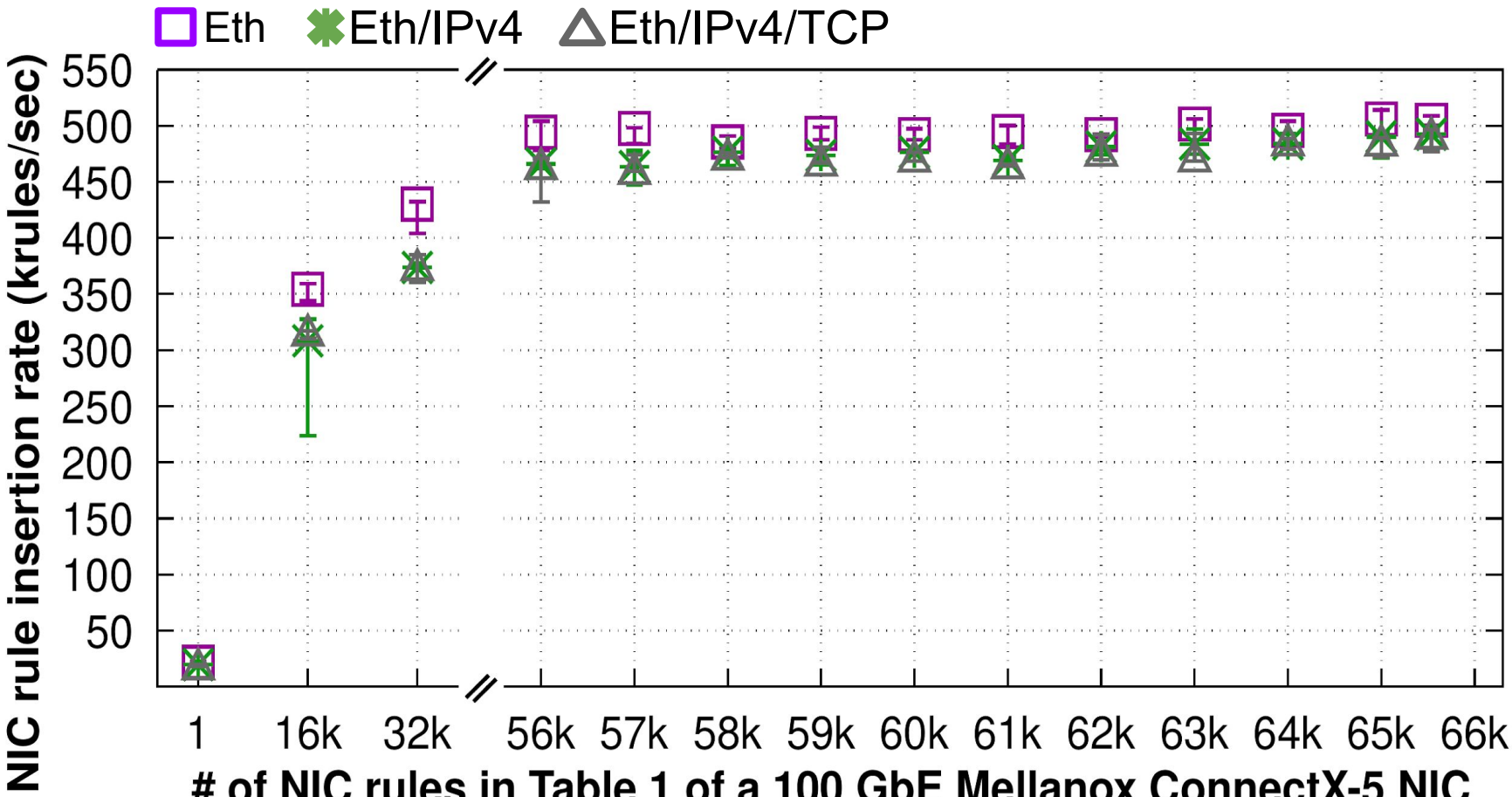
Scenario 5 - Results



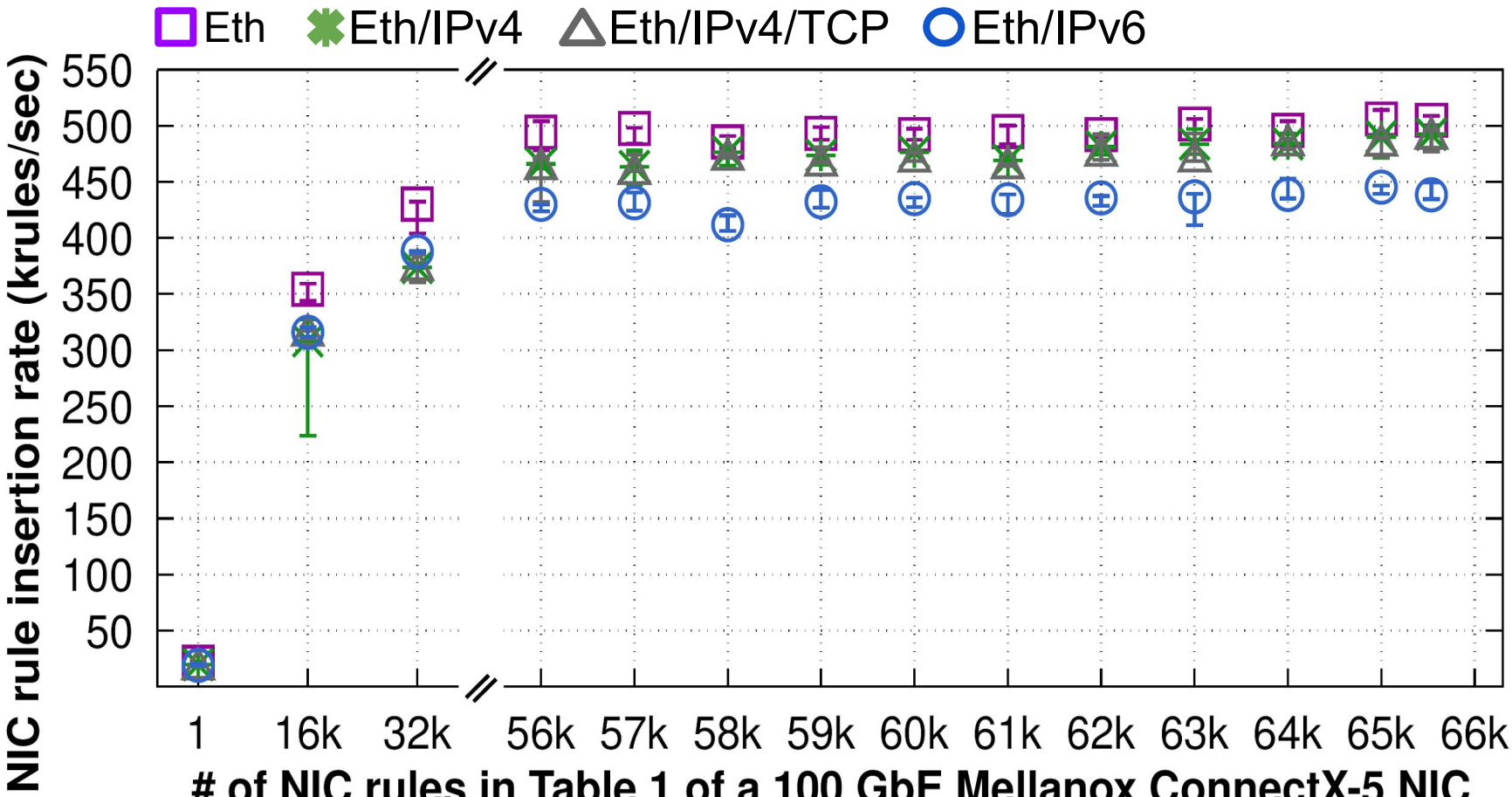
Scenario 5 - Results



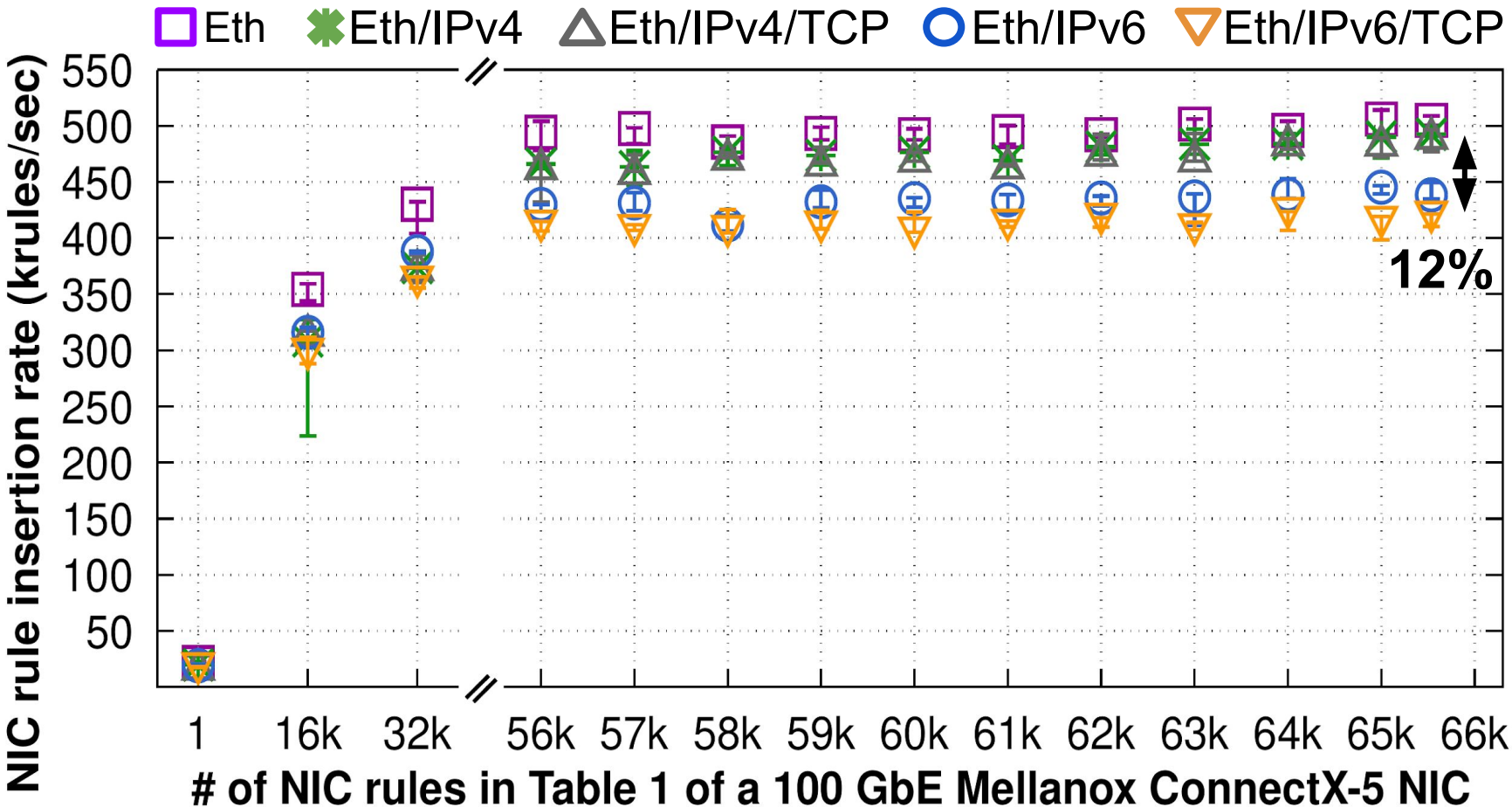
Scenario 5 - Results



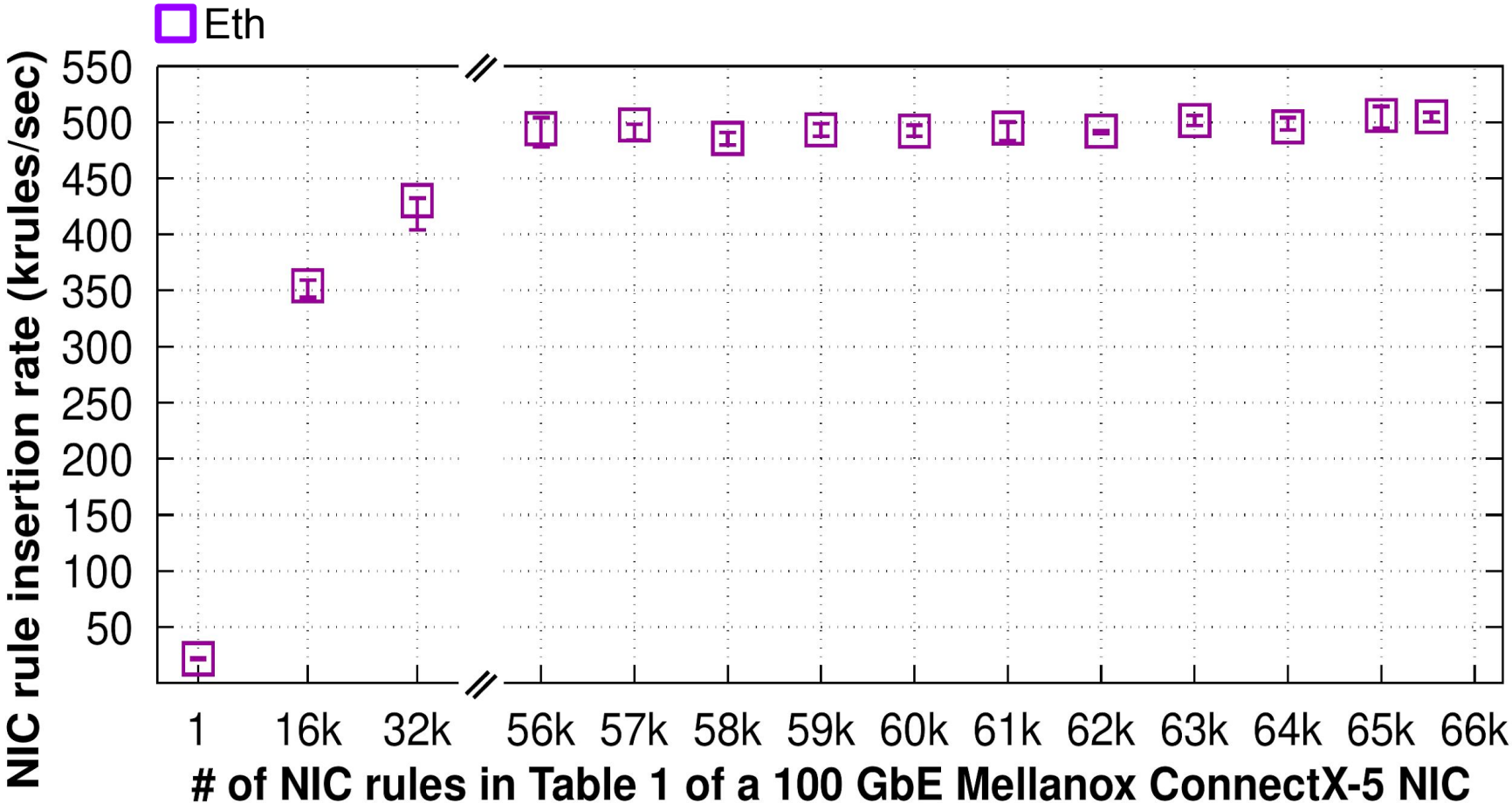
Scenario 5 - Results



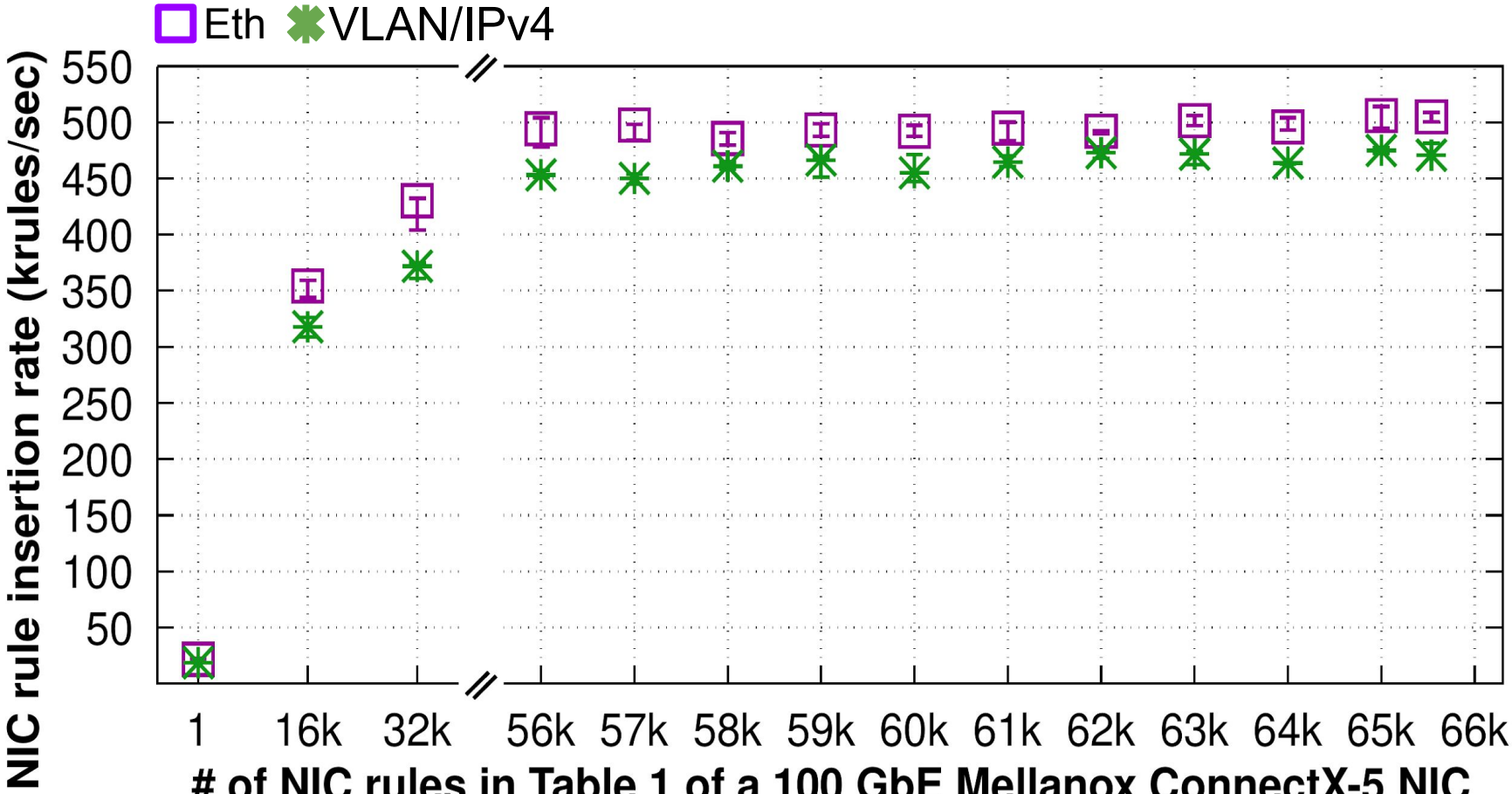
Scenario 5 - Results



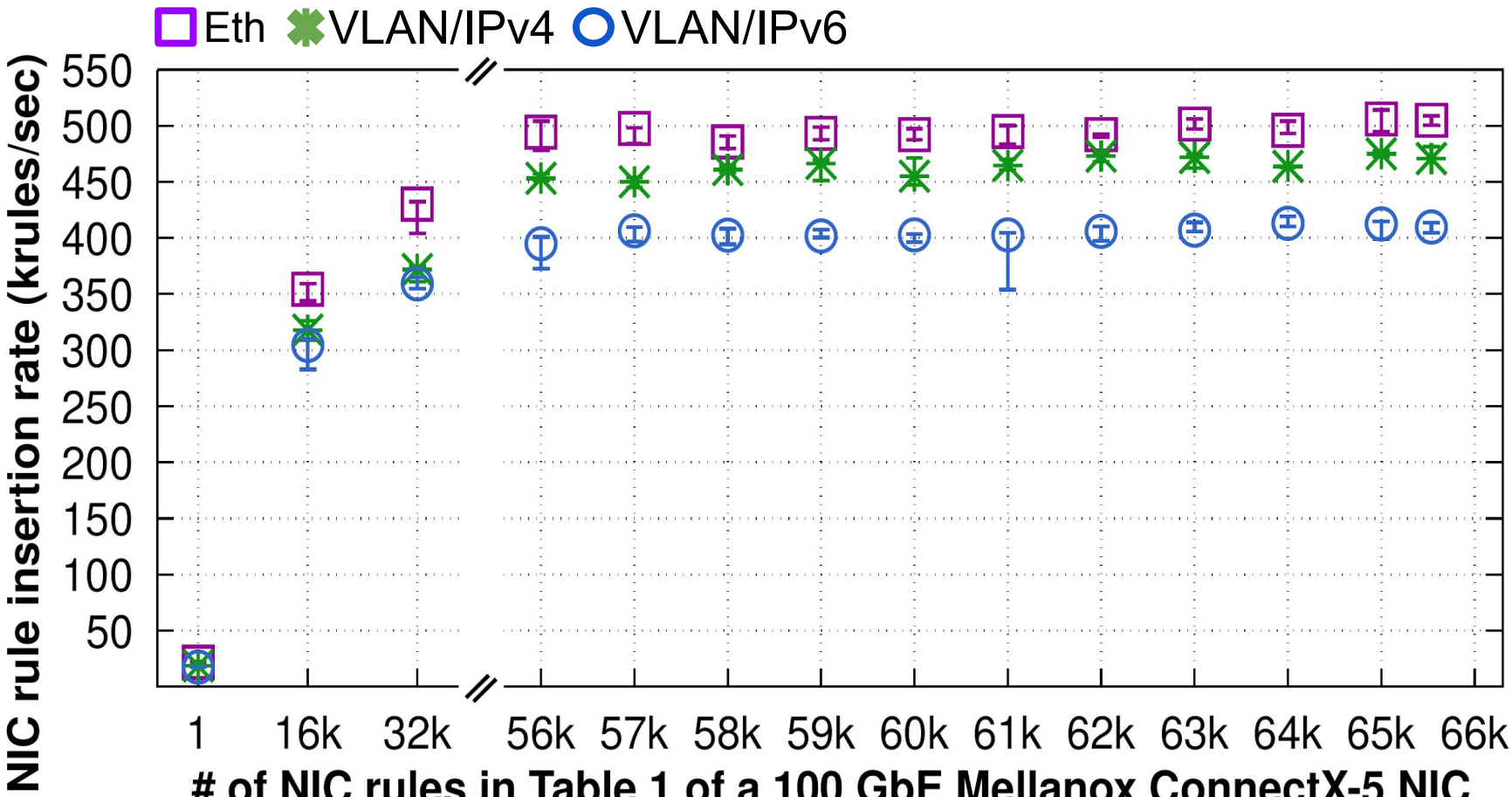
Scenario 5 - Results



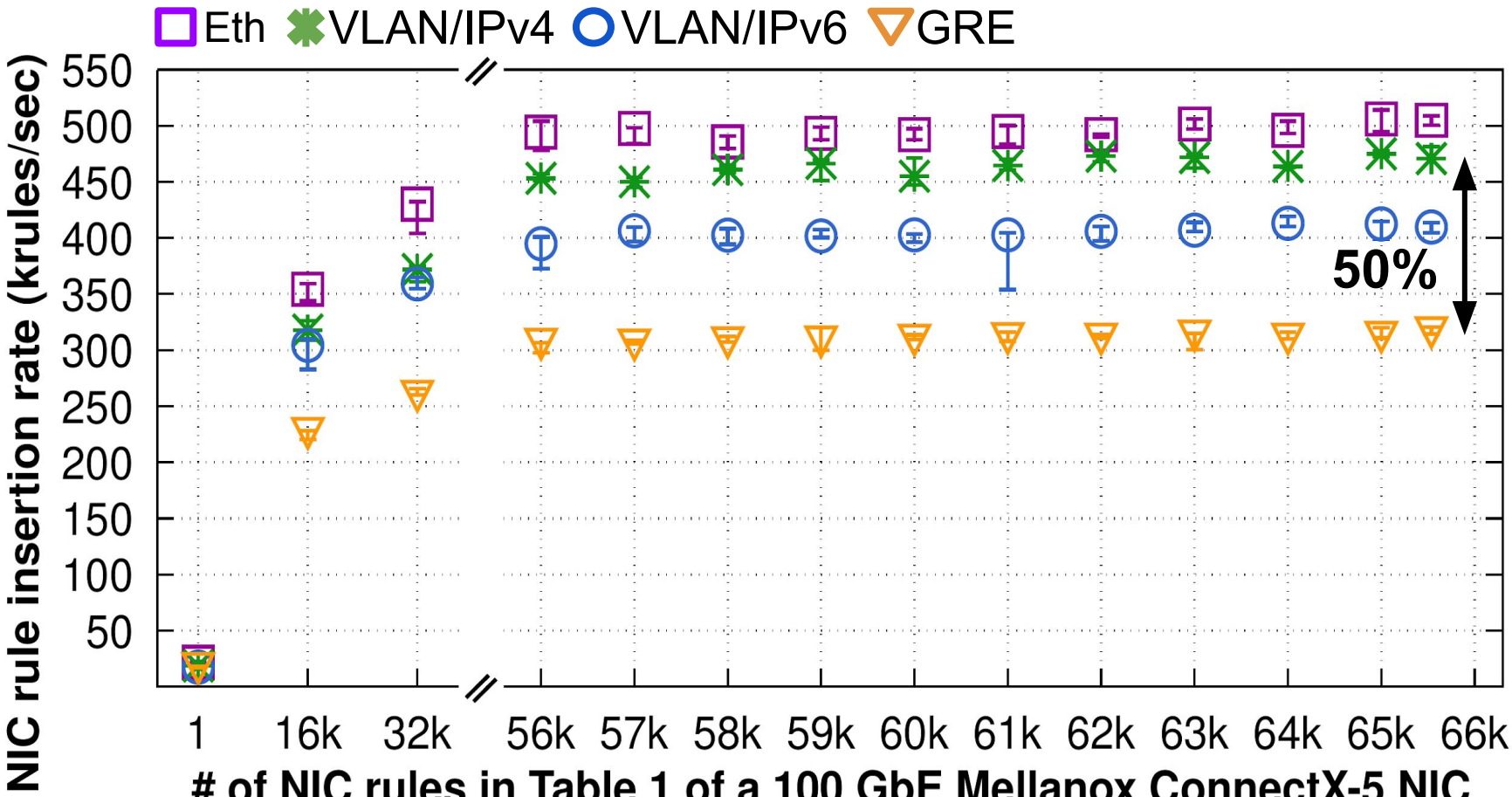
Scenario 5 - Results



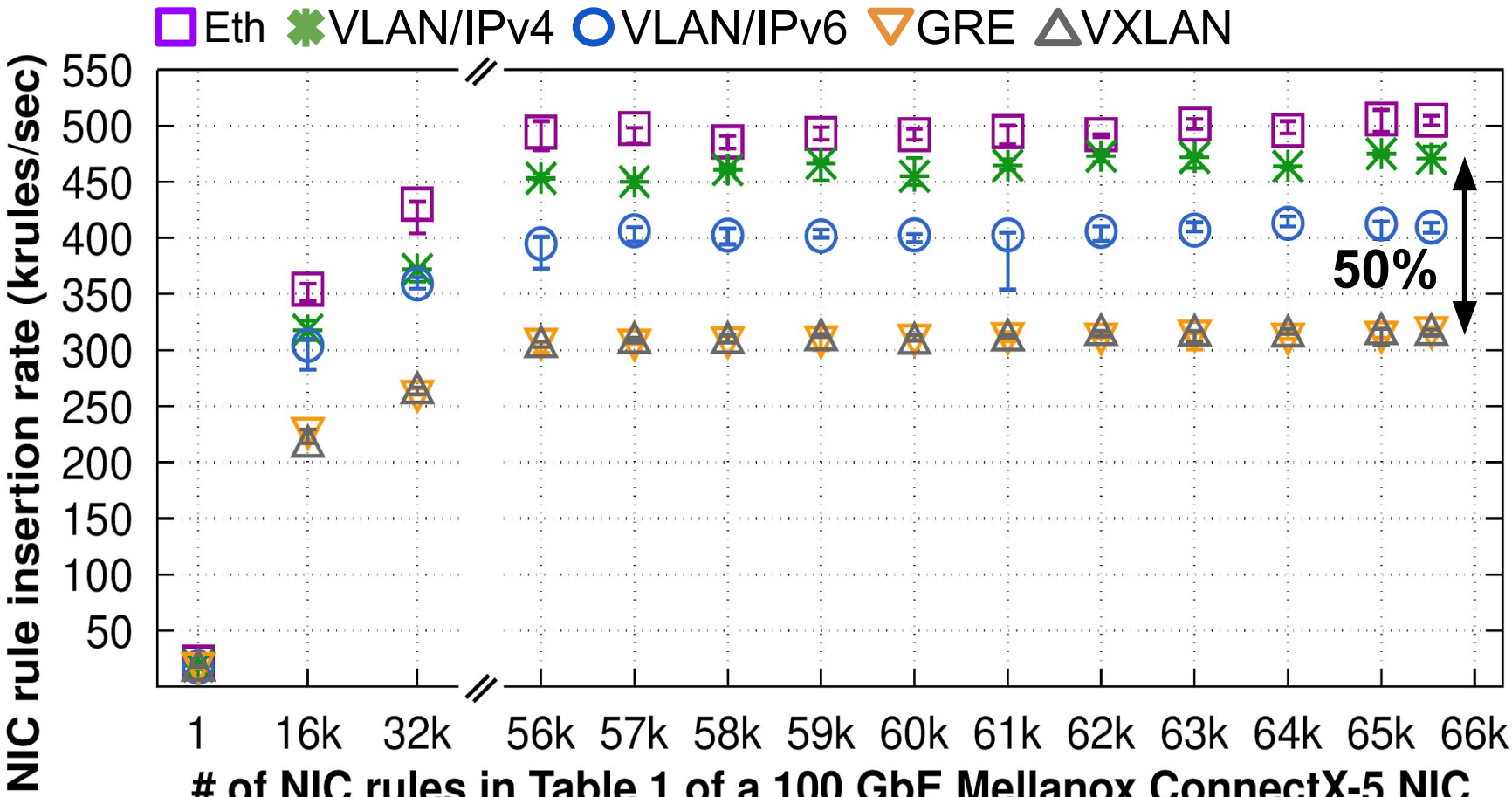
Scenario 5 - Results



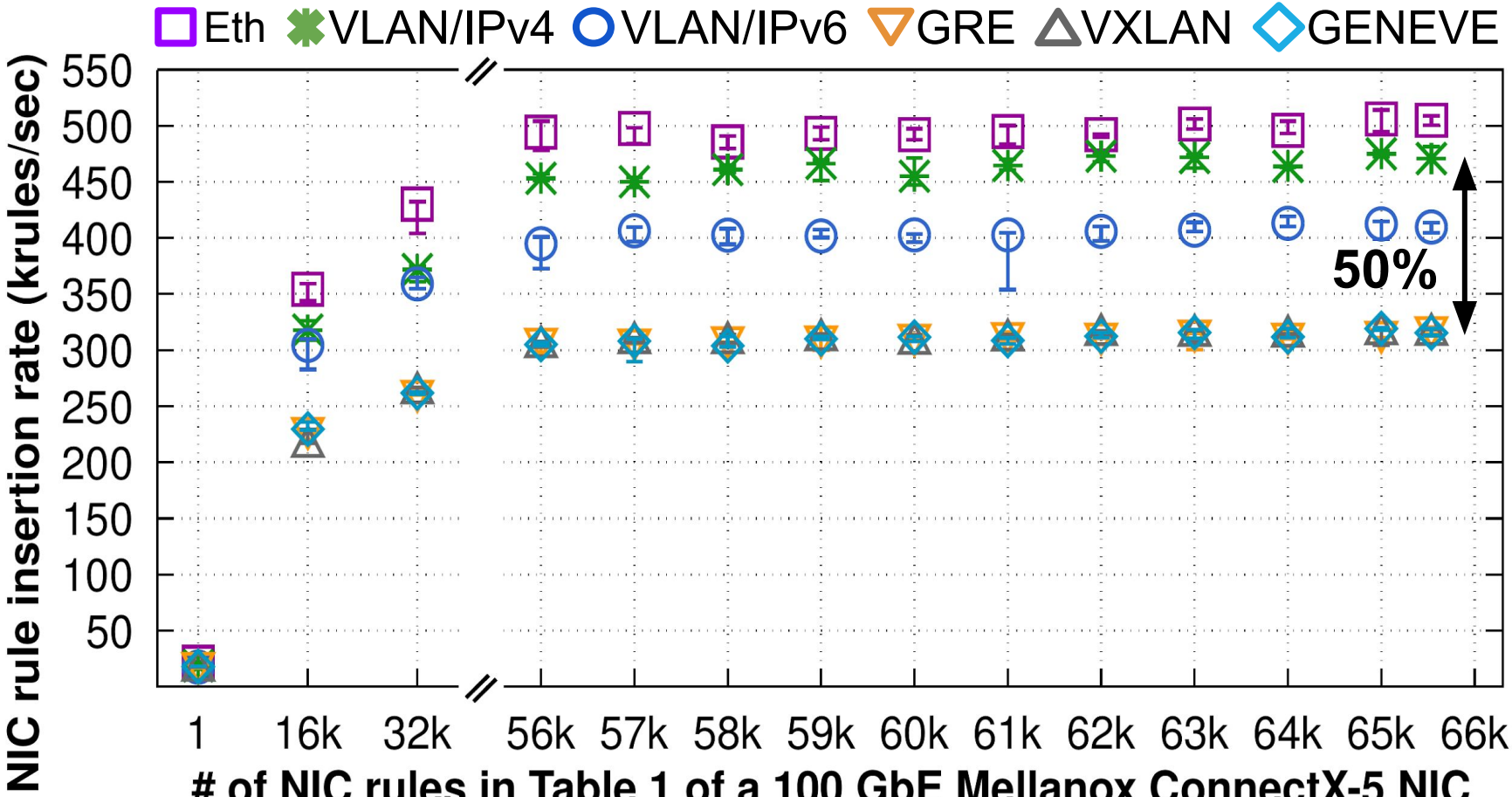
Scenario 5 - Results



Scenario 5 - Results



Scenario 5 - Results



Scenario 5 - Findings

Finding 6

- Internet protocol selection (i.e., IPv4 vs. IPv6) affects the NIC rule installation rate

Implications

- IPv6 vs. IPv4:
  - 5-181x faster in Table 0
  - 12% slower in Table 1

Finding 7

- Network slicing protocol selection affects the NIC rule installation rate

Implications

- Installing VLAN-based rules is up to 50% faster than installing tunnel-based rules

Why NICs are worth studying?

- Fundamental components of data centers (DCs)
  - With edge computing and 5G we are surrounded by DCs
- Modern applications are highly disaggregated
  - Application components (μservices) require fast networking
  - Link speeds keep increasing (200 Gbps and beyond...)
- Computation is increasingly offloaded to the network
  - NICs have also become smarter (Smart NICs)