

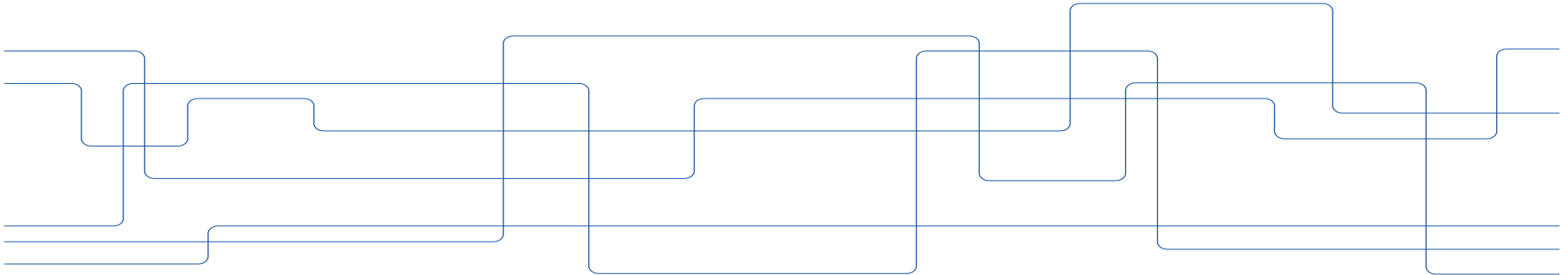


## Packet Order Matters!

# Improving Application Performance by Deliberately Delaying Packets

**Hamid Ghasemirahni\***, Tom Barbette<sup>†</sup>, Georgios P. Katsikas\*, Alireza Farshin\*, Amir Roozbeh\*<sup>‡</sup>,  
Massimo Girondi\*, Marco Chiesa\*, Gerald Q. Maguire Jr.\*<sup>‡</sup>, Dejan Kostic\*

\* KTH Royal Institute of Technology, <sup>†</sup> UCLouvain, <sup>‡</sup> Ericsson Research

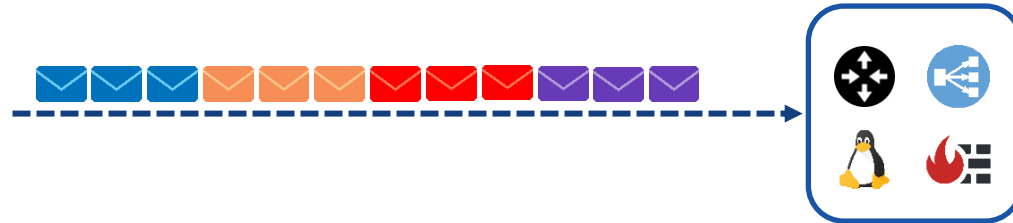


# Introduction

Scenario 1:

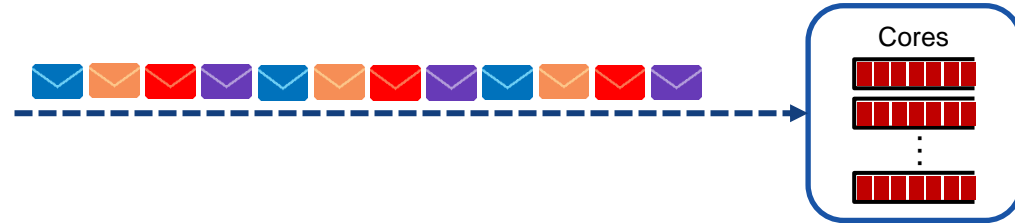


Scenario 2:

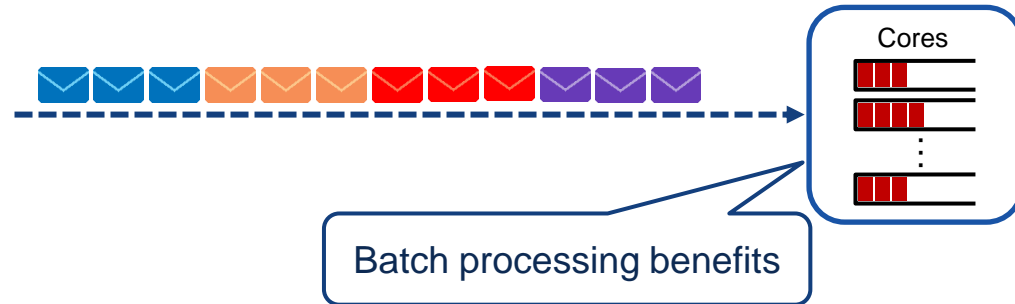


# Introduction

Scenario 1:



Scenario 2:



✓ Pay the price of re-ordering packets

# Main Steps



1- How much does packet order matter? And why?



2- How much is the possibility for reordering packets in a network?



3- Reframer! A software solution to re-order packets!

**84%** higher throughput with a network function chain

**26%** improvement in terms of 99.9<sup>th</sup> percentile latency

**HOW MUCH DOES PACKET ORDER MATTER?**

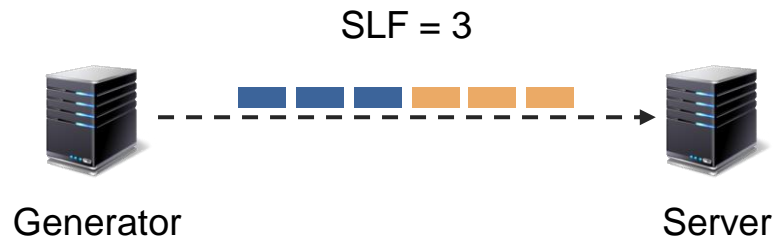
# Experimental Setup

## Spatial Locality Factor (SLF)

Average number of packets, in the **same flow**, that arrive **back-to-back** at a network device

Generate a *synthetic* train of packets with a given SLF

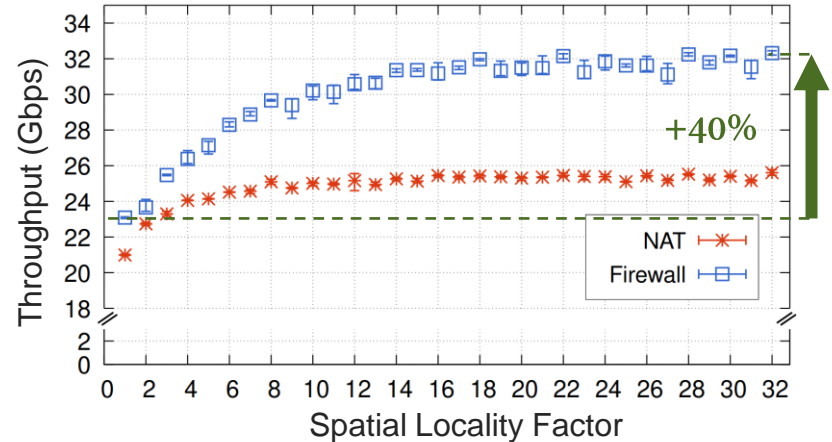
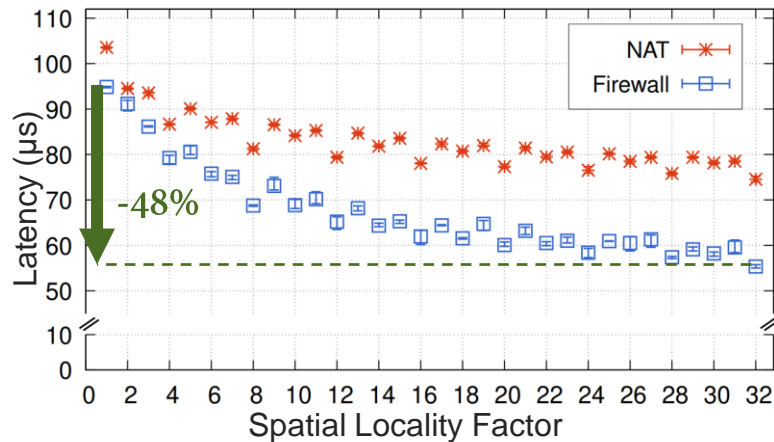
Measure the performance indicators on the server



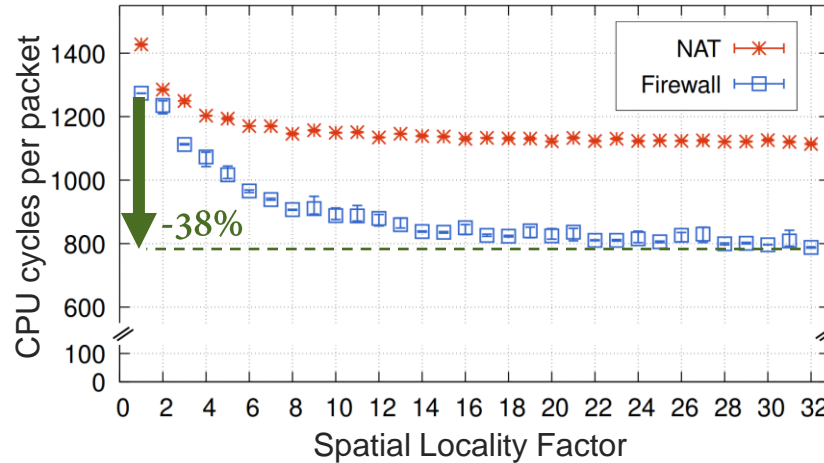
# Impact on Network Functions

NF use cases: NAT and Firewall

Implementation: FastClick with DPDK

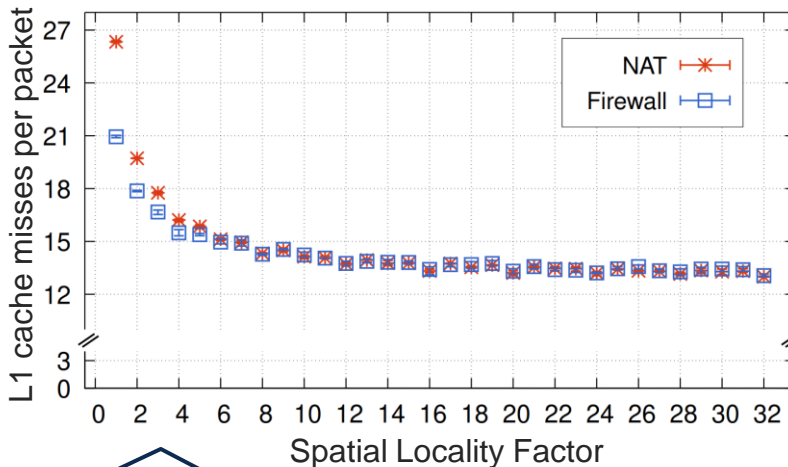
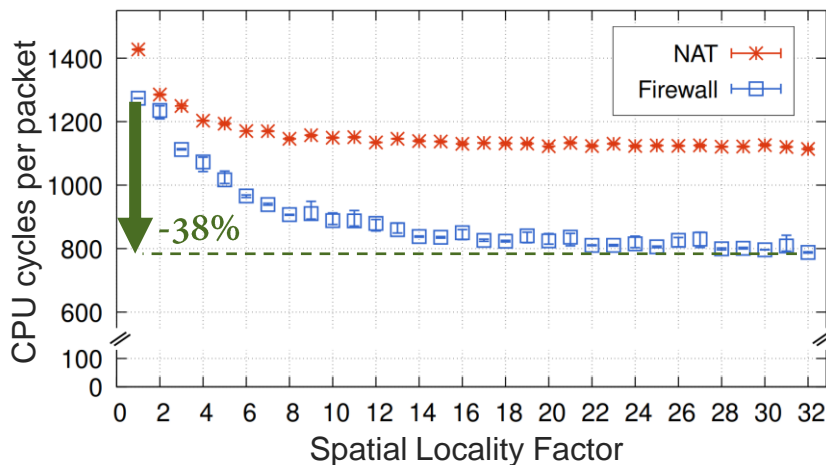


# Impact on Network Functions





# Impact on Network Functions



Cache utilization benefit extends to higher level of caches by deploying complex NFs!

**HOW MUCH IS THE POSSIBILITY FOR REORDERING  
PACKETS IN A IN REAL-WORLD NETWORK?**

# Real-world Trace Analysis

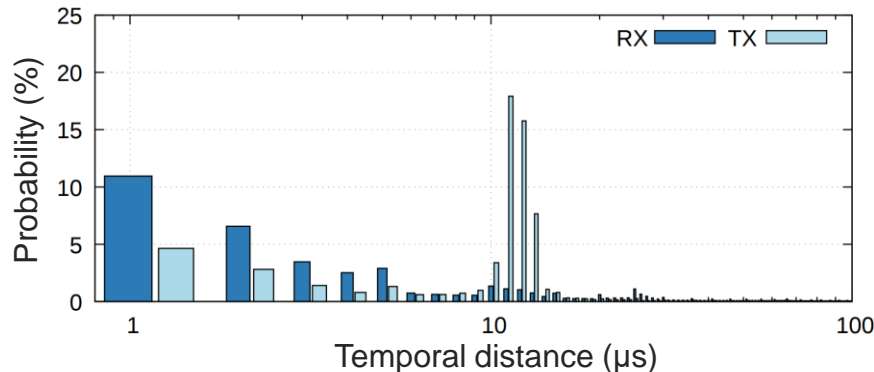
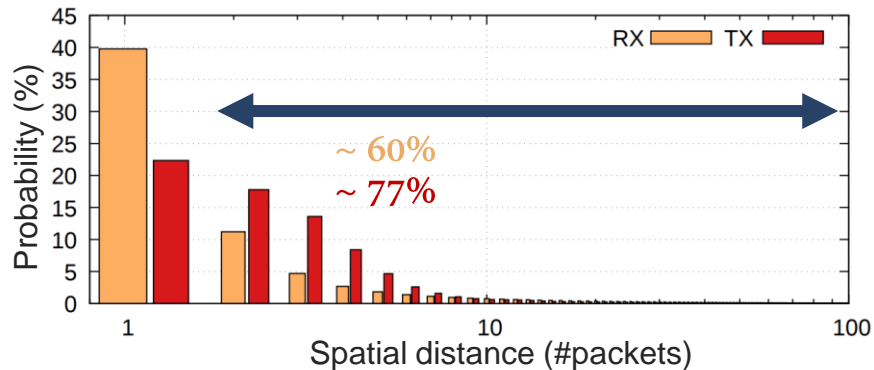
## Spatial distance:

Packets gap between two consecutive packets of the same flow

## Temporal distance:

Time between two consecutive packets of the same flow

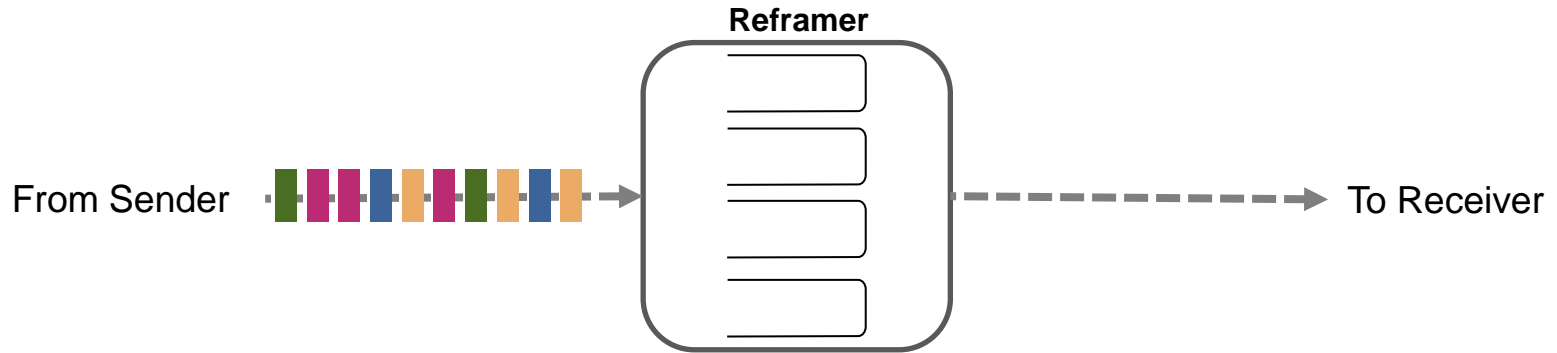
A great opportunity to **reorder** packets by **deliberately delaying** them for a **short period of time**



**REFRAMER**

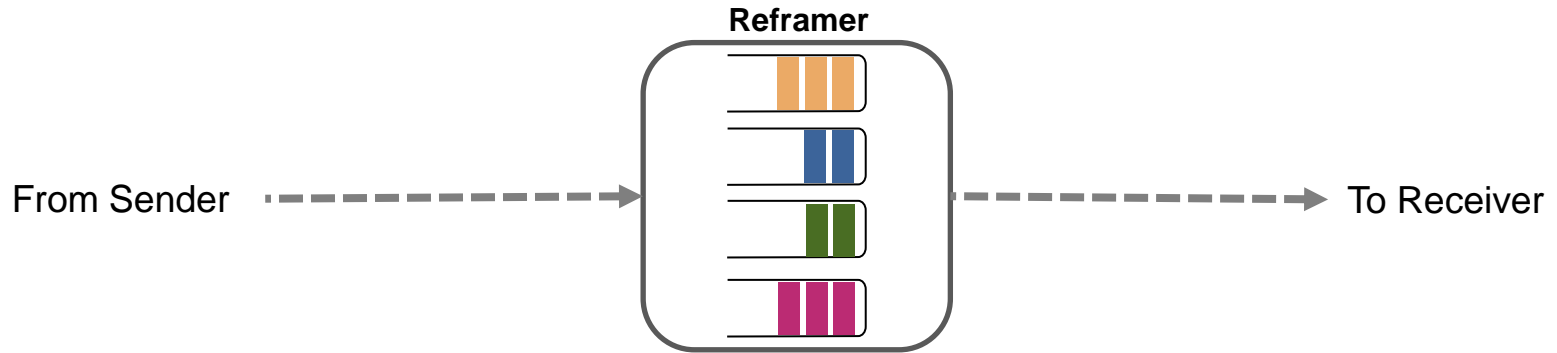
# Reframer

We leverage the idea of **classifying**, briefly **delaying**, and **reordering** the incoming packets to **increase spatial locality**



# Reframer

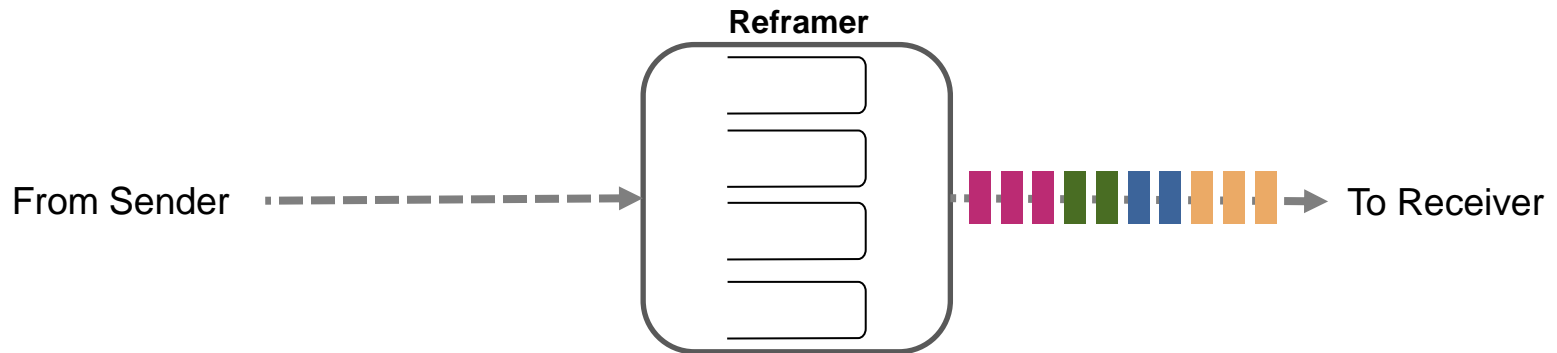
We leverage the idea of **classifying**, briefly **delaying**, and **reordering** the incoming packets to **increase spatial locality**





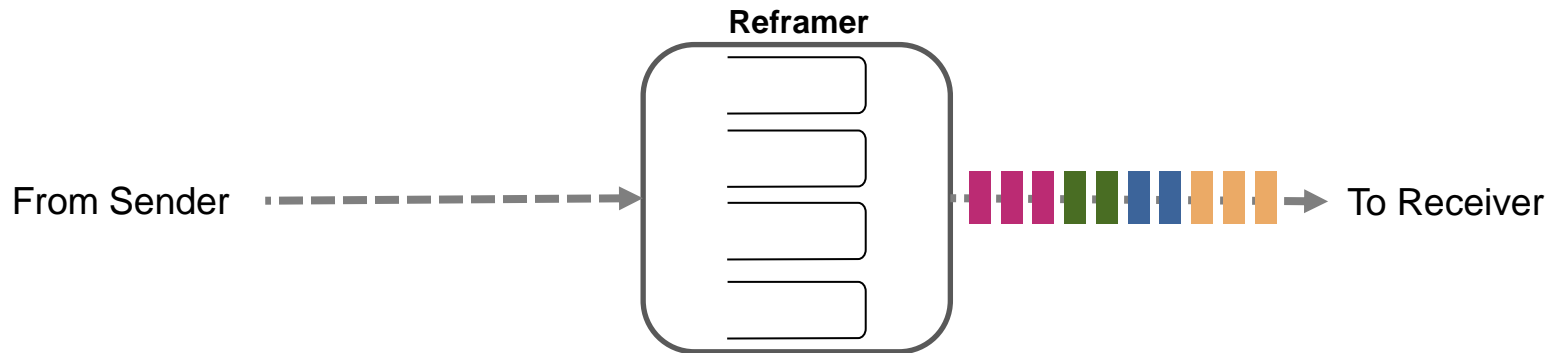
# Reframer

We leverage the idea of **classifying**, briefly **delaying**, and **reordering** the incoming packets to **increase spatial locality**



# Reframer

We leverage the idea of **classifying**, briefly **delaying**, and **reordering** the incoming packets to **increase spatial locality**



Buffering time  $\nearrow$  : Increases packets latency

Buffering time  $\searrow$  : Reduces output spatial locality





# Reframer (Challenges)

## Efficiency

Having a small number of operations for buffering and flushing out

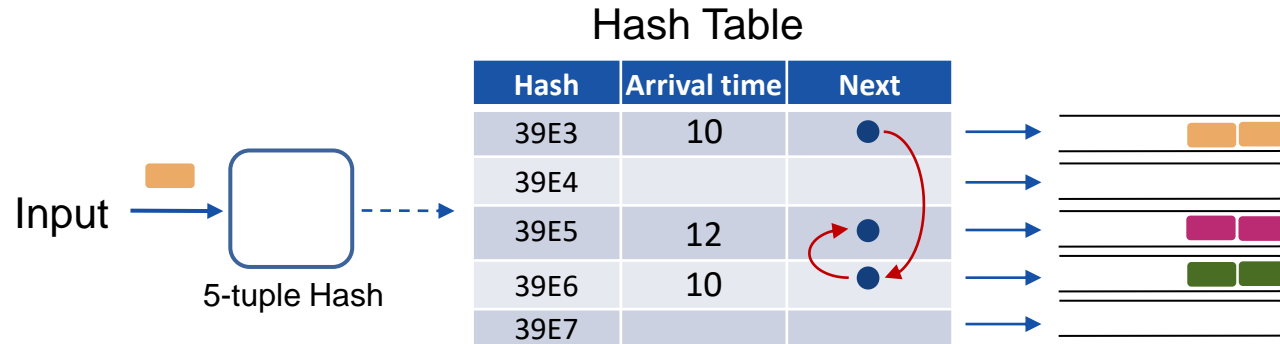
## Flexibility

Realizing as many scheduling policies as possible

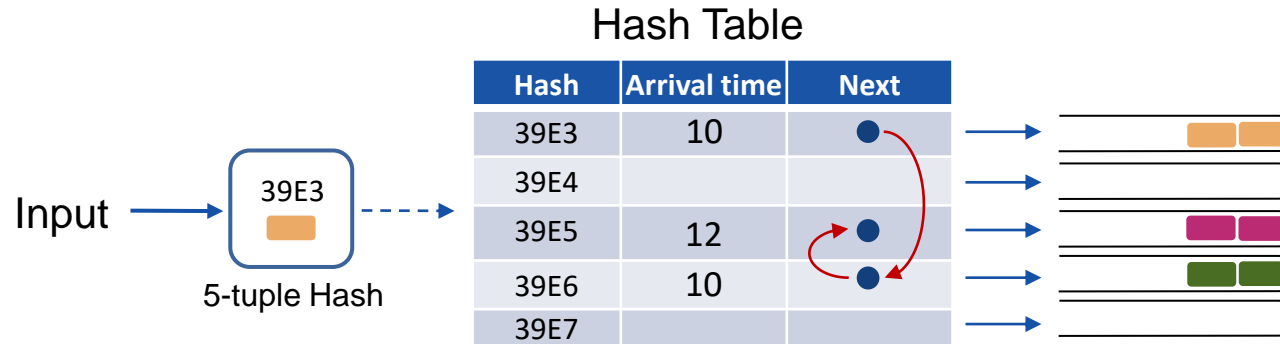
## Scalability

Scaling up linearly by increasing available cores or instances

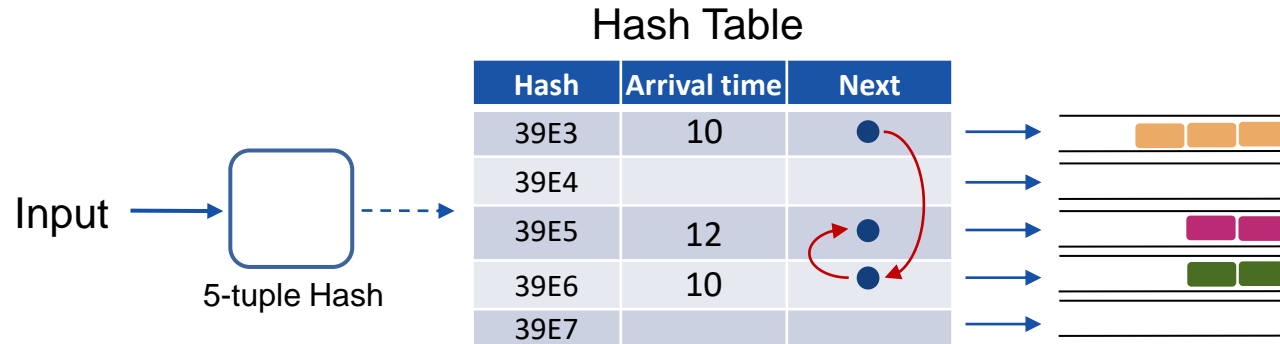
# Reframer (Design)



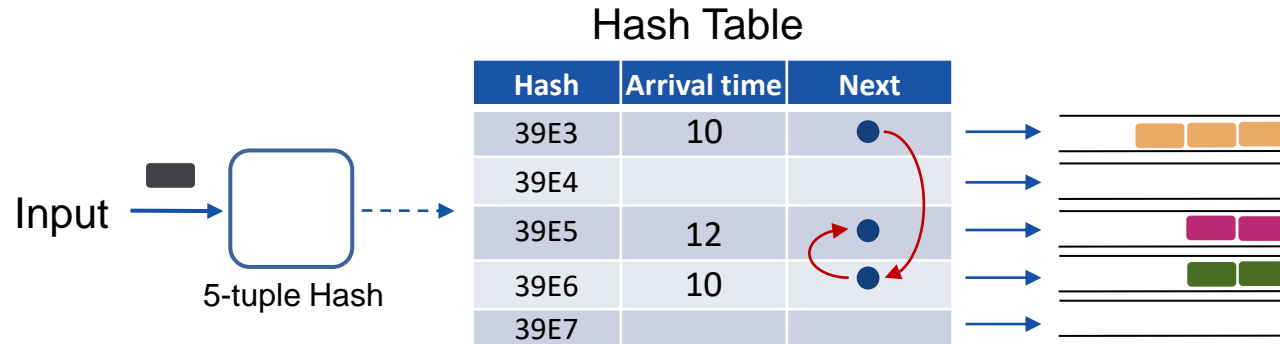
# Reframer (Design)



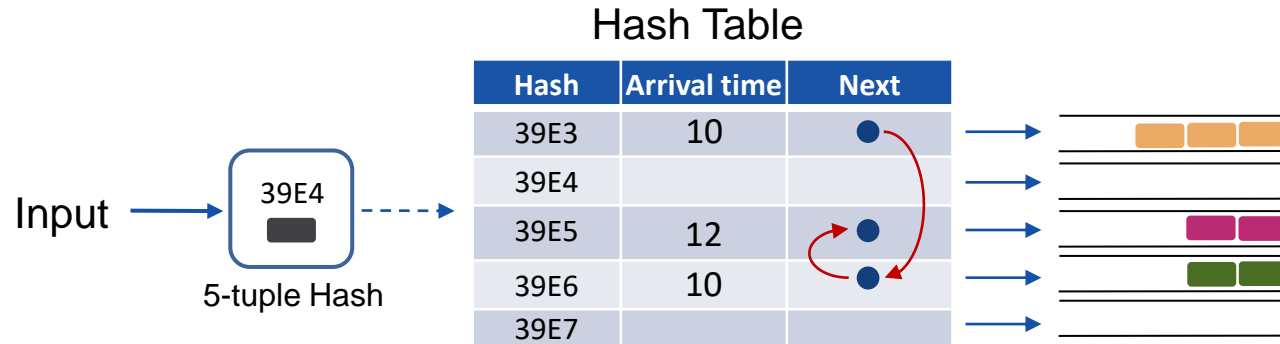
# Reframer (Design)



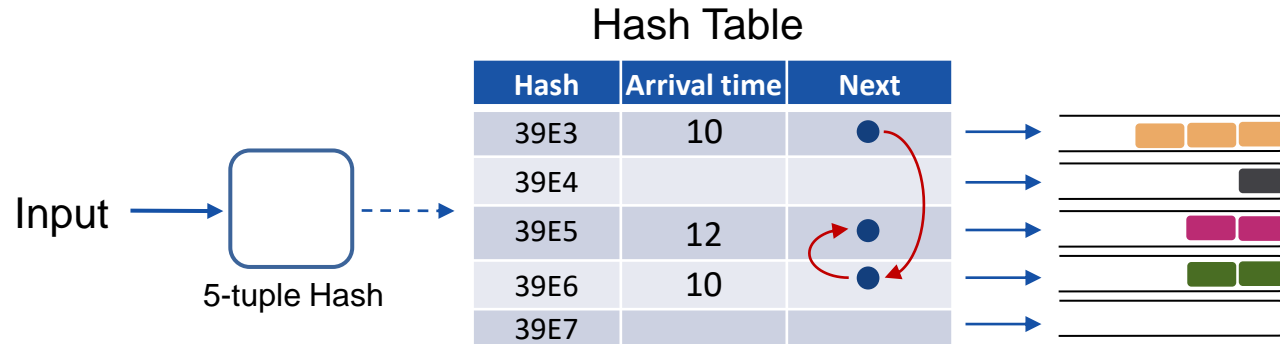
# Reframer (Design)



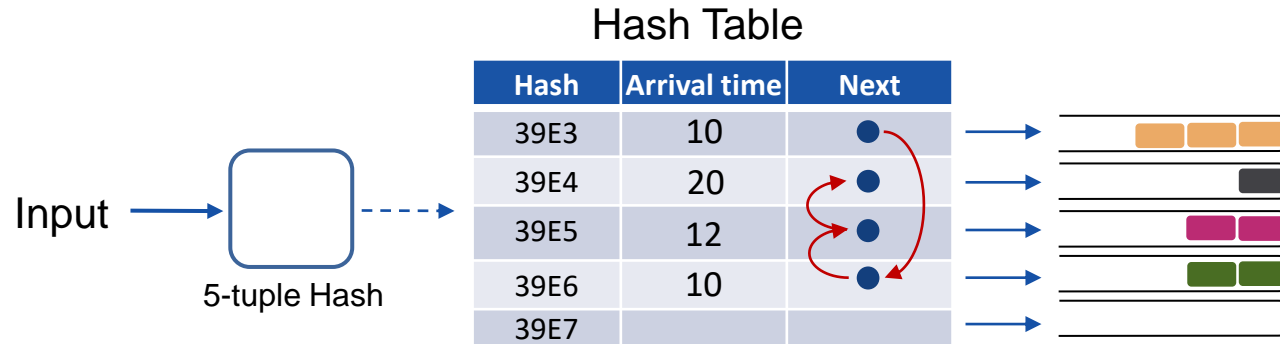
# Reframer (Design)



# Reframer (Design)

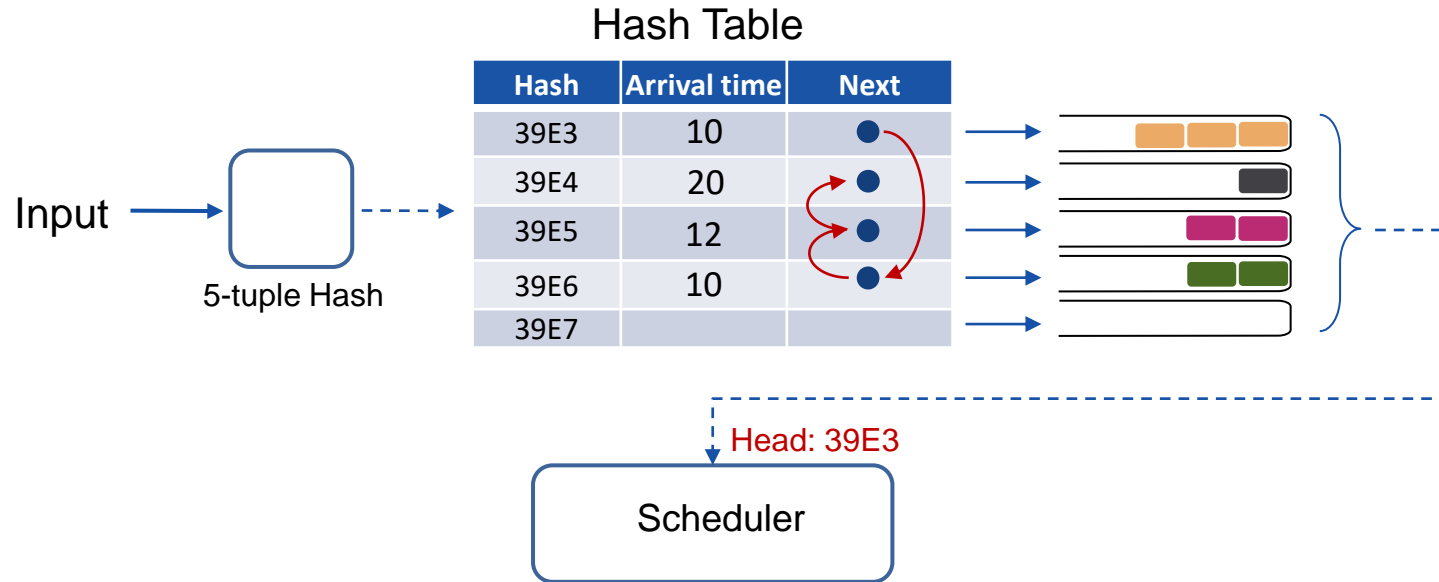


# Reframer (Design)

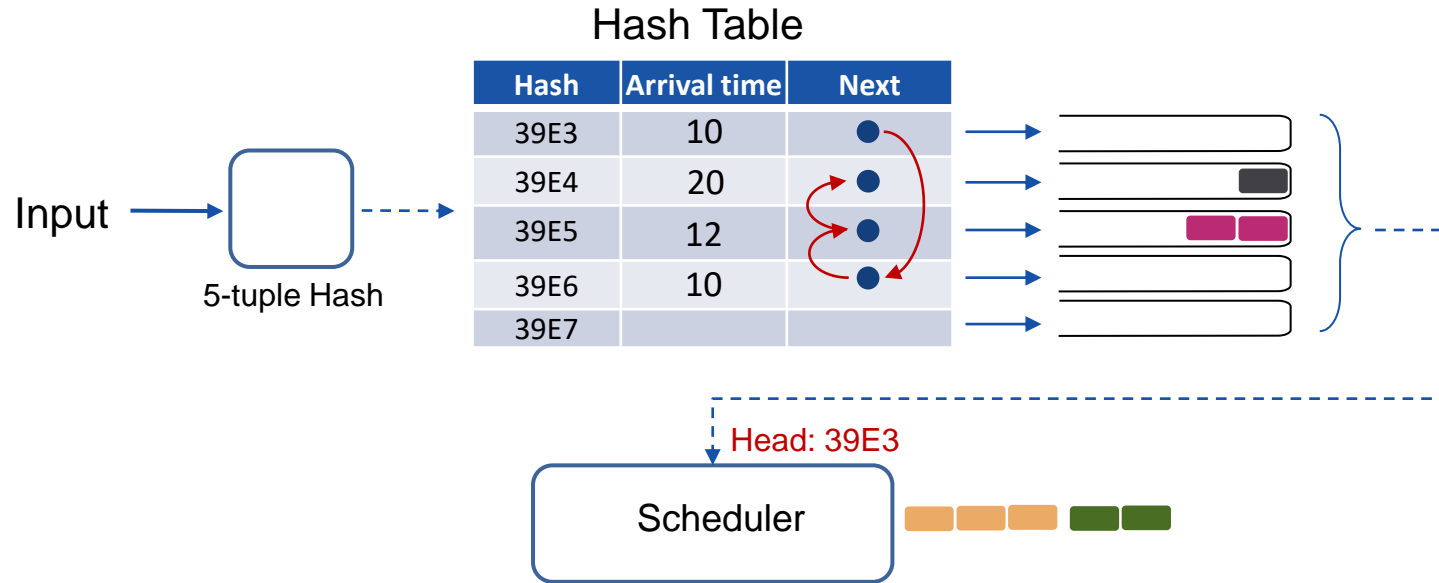




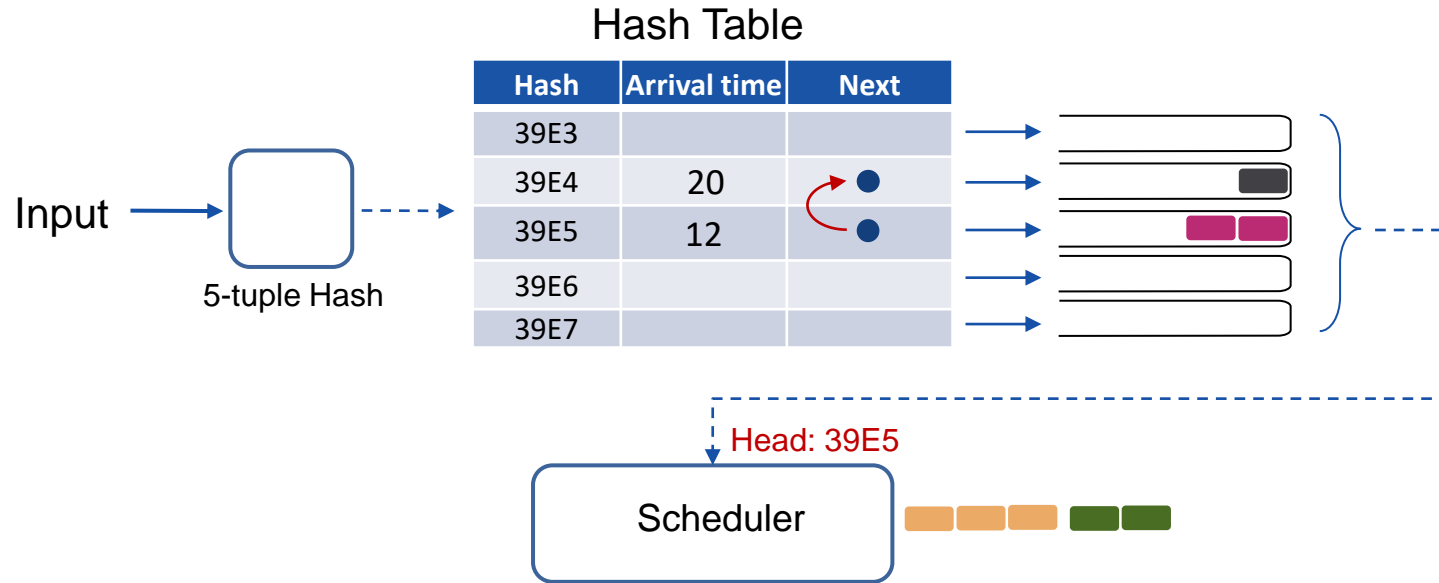
# Reframer (Design)



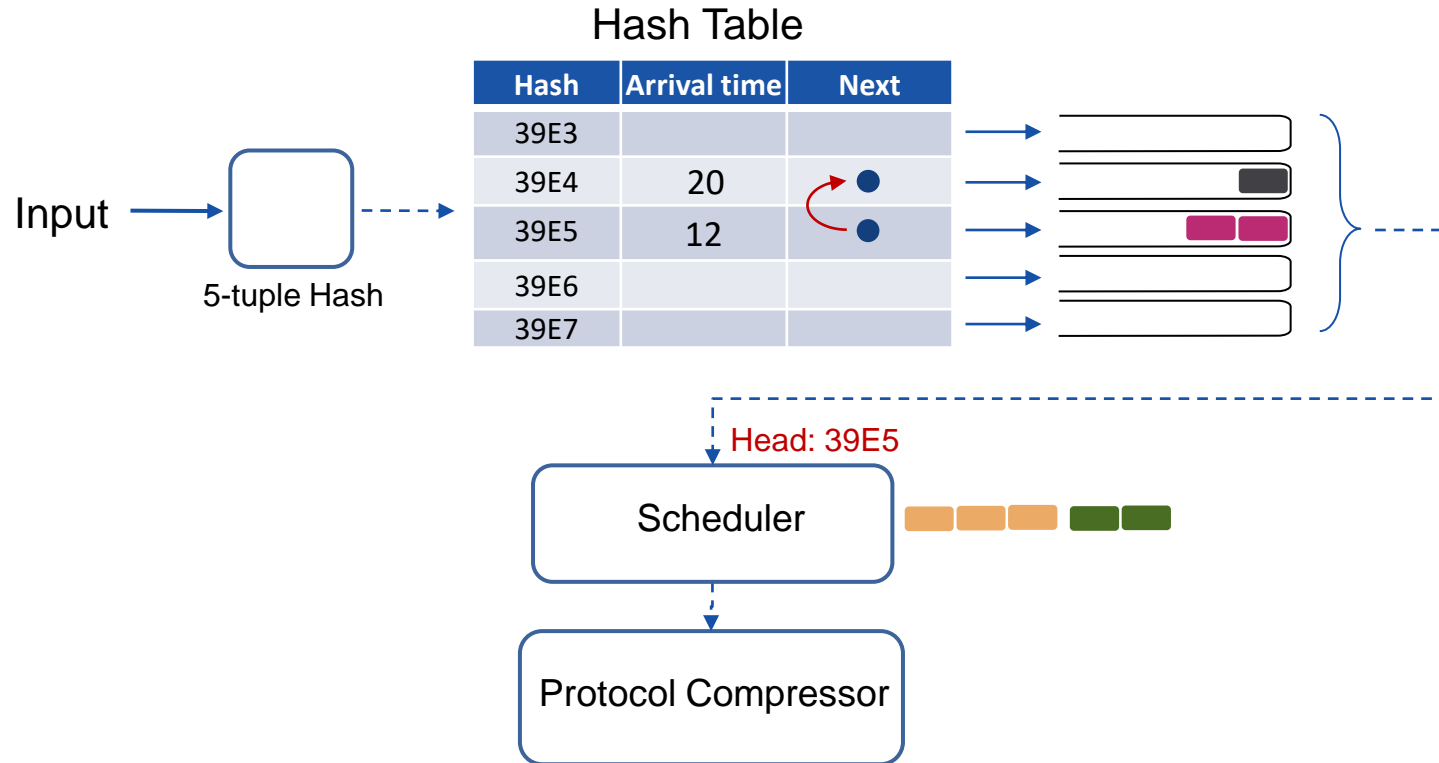
# Reframer (Design)



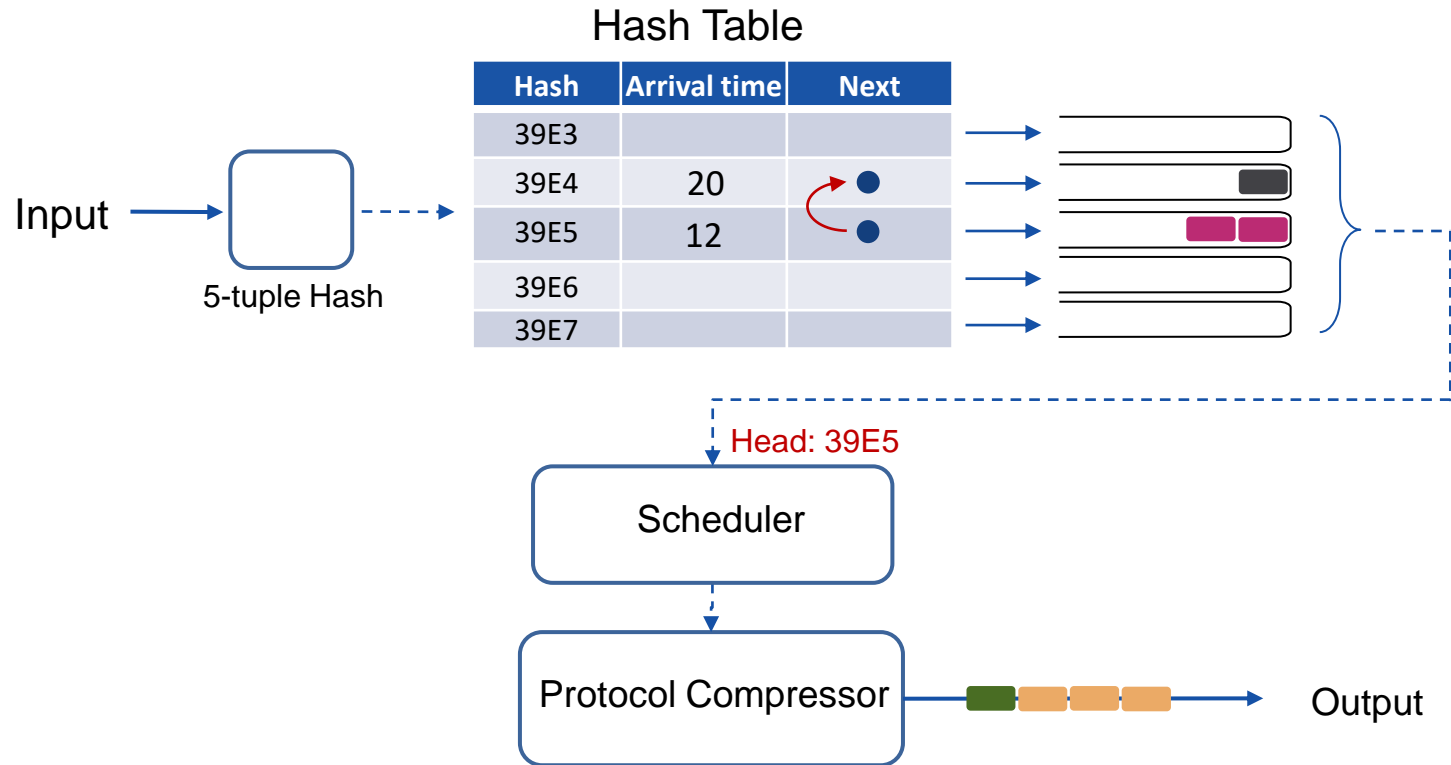
# Reframer (Design)



# Reframer (Design)



# Reframer (Design)



# EVALUATION

# Evaluation

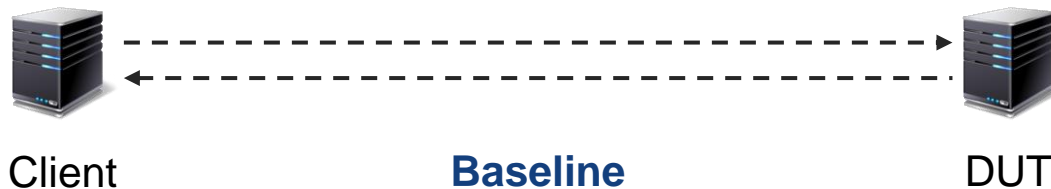
## Questions:

Can Reframer increase throughput of an NF chain?

Does the benefit of Reframer persist with the same amount of resources?

How scalable is Reframer?

## Testbed:



# Evaluation

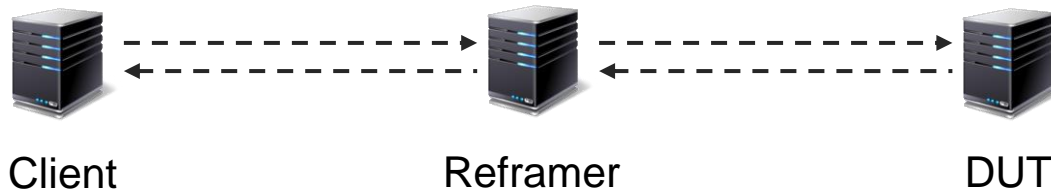
## Questions:

Can Reframer increase throughput of an NF chain?

Does the benefit of Reframer persist with the same amount of resources?

How scalable is Reframer?

## Testbed:

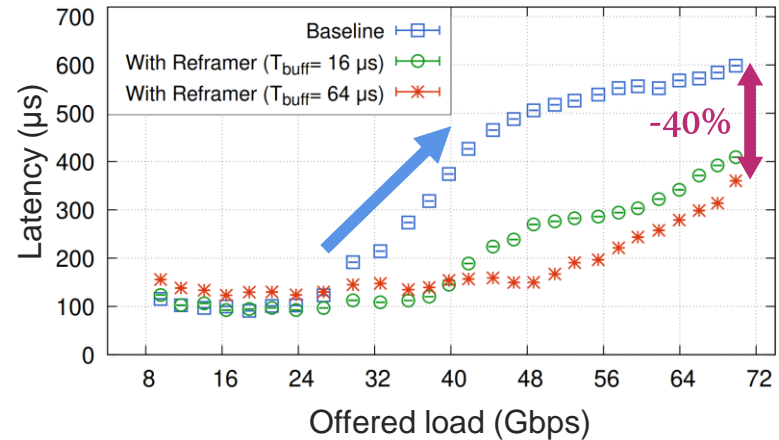
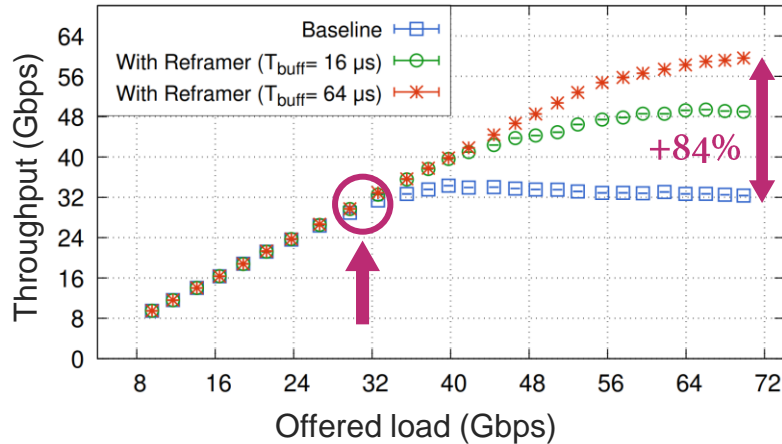




# Packet Level Analysis

Reframer is deployed in front of a NF chain on a **dedicated** server

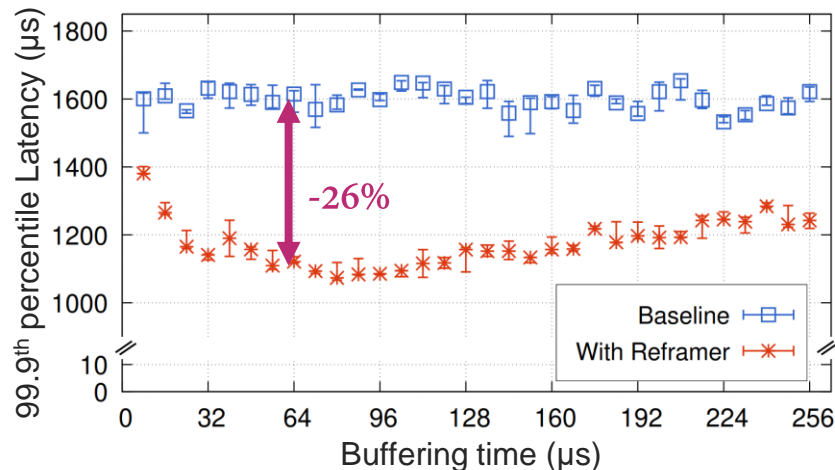
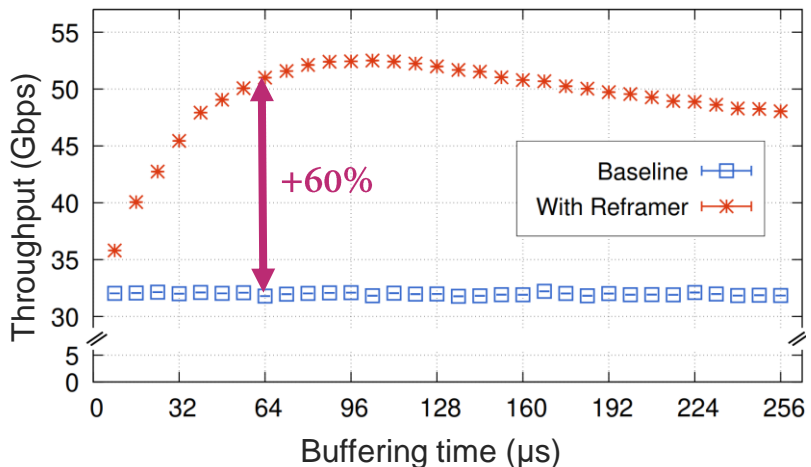
DUT applications:



# Packet Level Analysis

Reframer is deployed on the **same server** with the NF chain

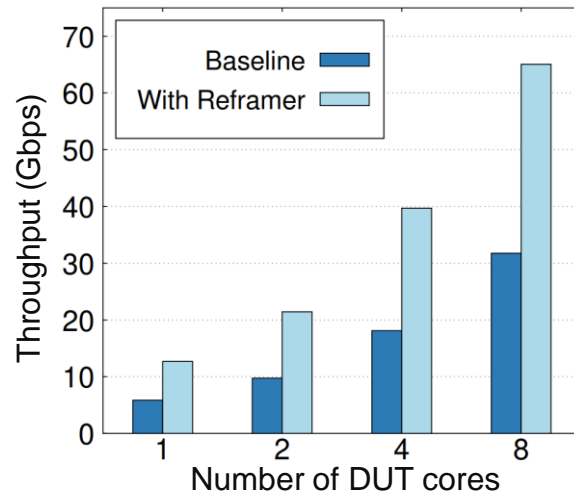
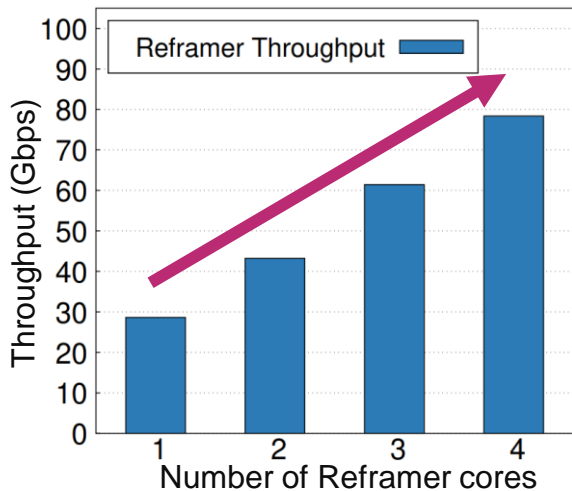
No additional hardware resource is used to run Reframer!



# Reframer Scalability

Reframer scales almost **linearly** with increasing CPU cores

Reframer benefit **does not** depend on number of DUT cores



**CONCLUSION**



# Conclusion

## Three major steps:

**Step1.** Packets order impacts network functions and Linux kernel performance

**Step2.** Our Real-world trace analysis confirms the possibility of ordering packets

**Step3.** We design, implement, and evaluate **Reframer**

## Evaluation:

Up to **84%** benefit in terms of throughput

Up to **26%** improvement in terms of 99.9<sup>th</sup> percentile latency



hamidgh09/Reframer



*Knut och Alice  
Wallenbergs  
Stiftelse*