Carousel: Scalable Traffic Shaping at End Hosts

Ahmed Saeed, Nandita Dukkipati, Vytautas Valancius,
Vinh The Lam, Carlo Contavalli, and Amin Vahdat

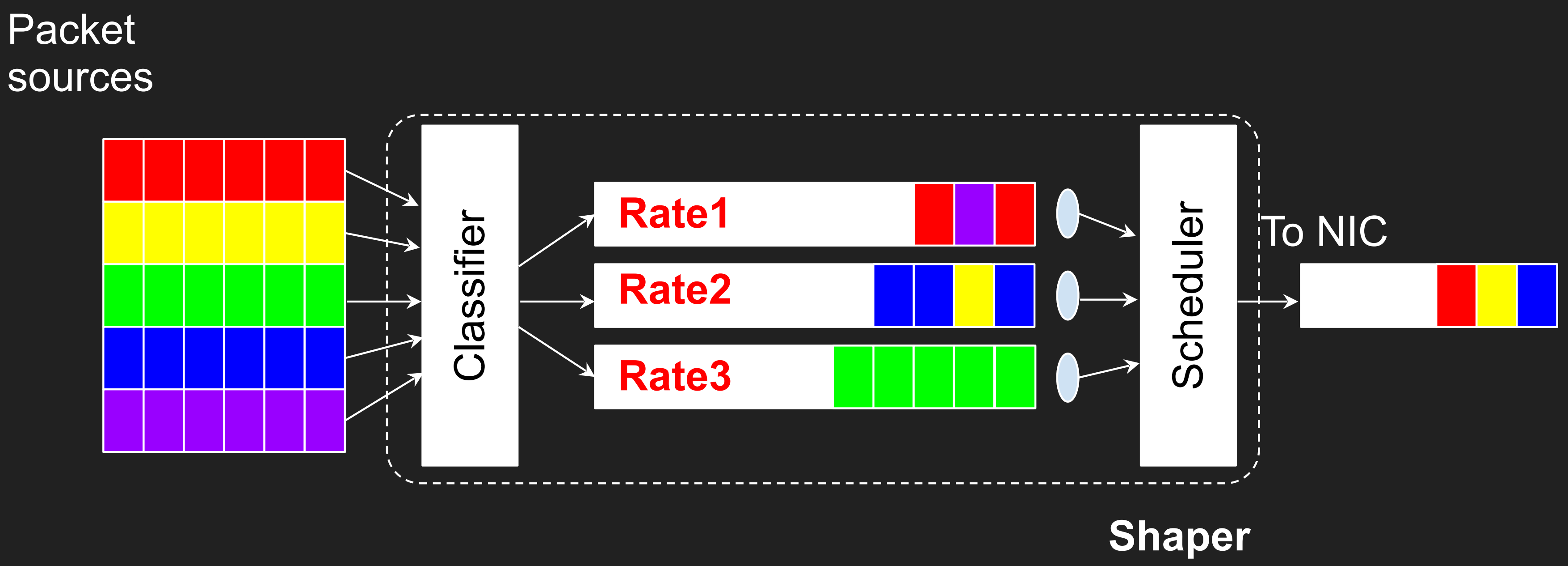
Google
google.com/datacenters

Rate limiting and isolation between thousands of flows per machine [BwE - SIGCOMM ’15]

Rate limiting and isolation between thousands of flows per machine [BwE - SIGCOMM ’15]

New protocols that require per-flow pacing [TCP BBR and TIMELY - SIGCOMM ’15]

Traffic Shaping



Packet sources

Classifier

Rate1

Rate2

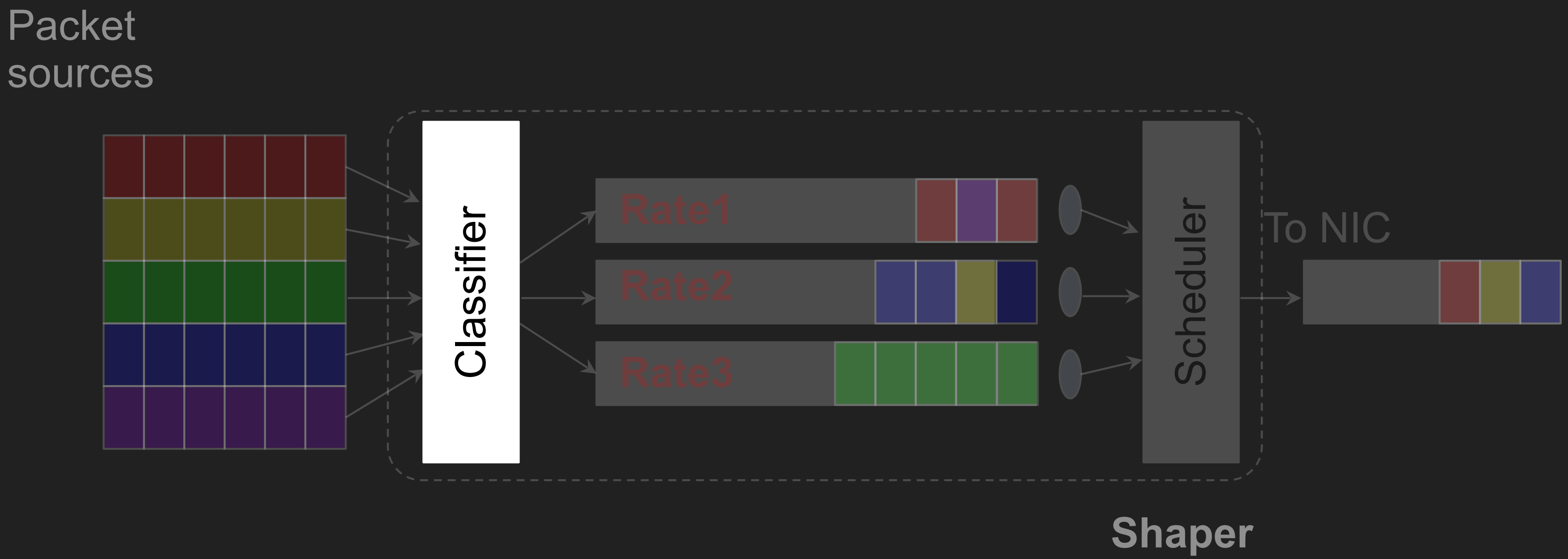
Rate3

Scheduler

To NIC

Shaper

Traffic Shaping



Packet sources

Classifier

Rate1

Rate2

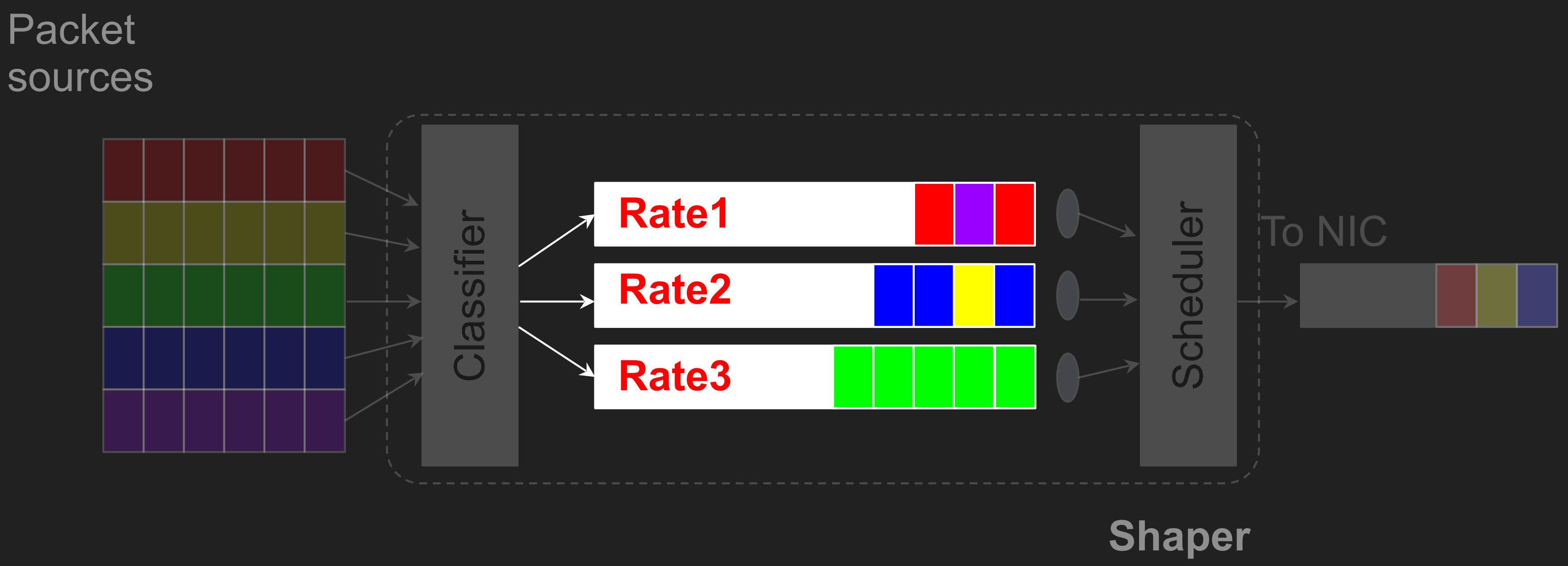
Rate3

Scheduler

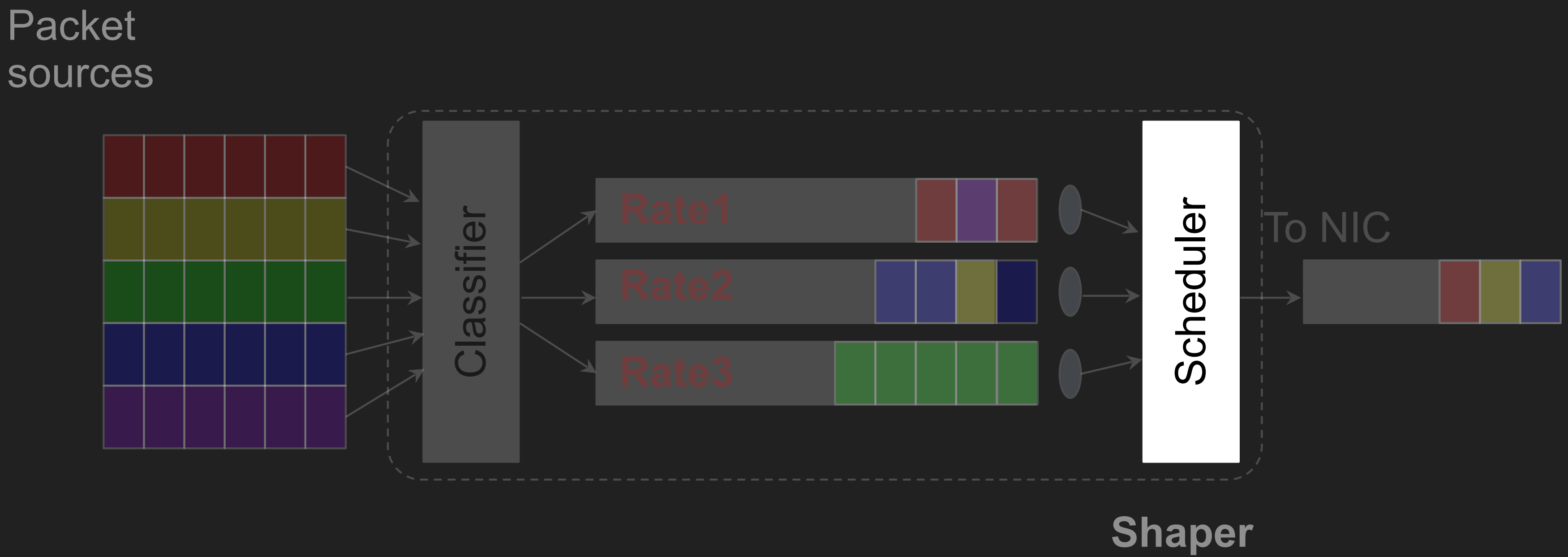
To NIC

Shaper

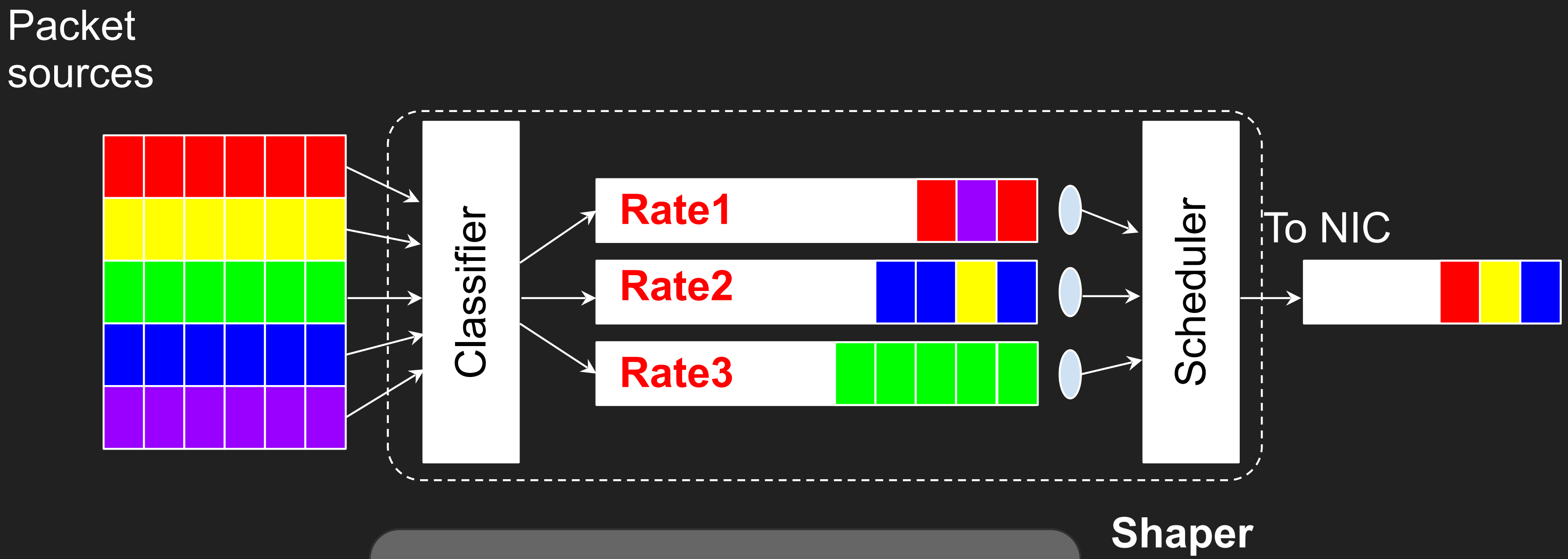
Traffic Shaping



Traffic Shaping

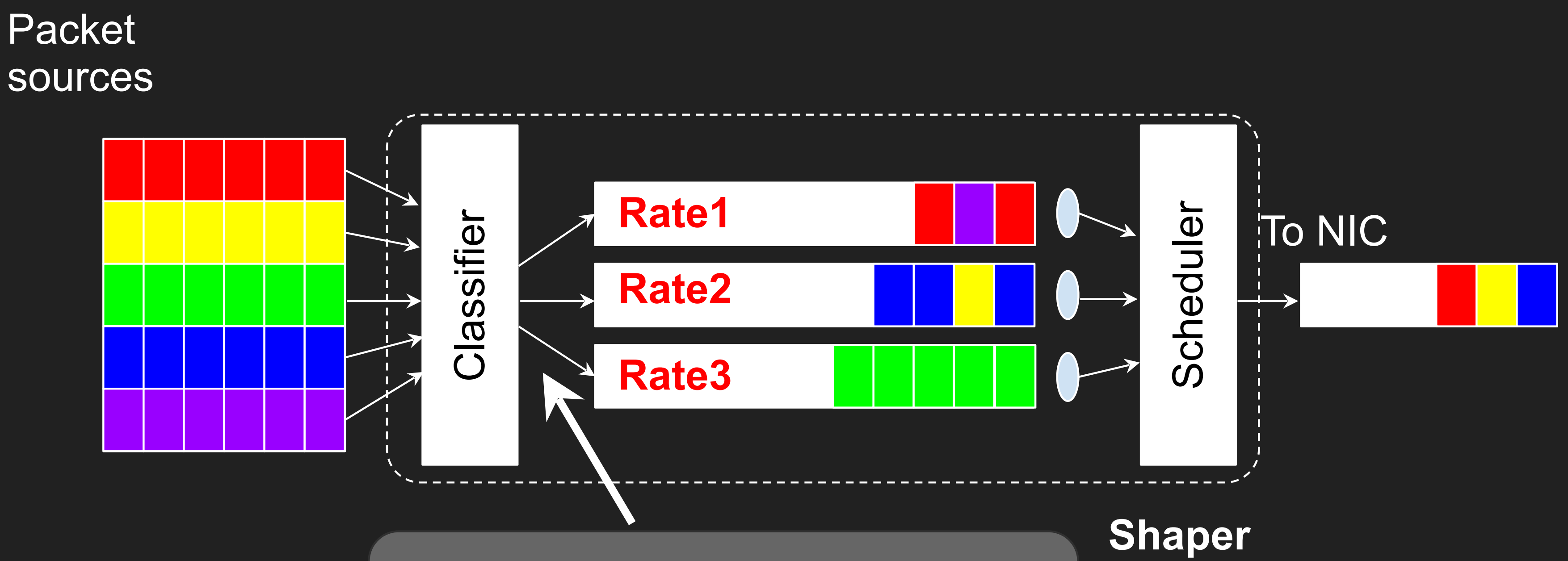


Traffic Shaping



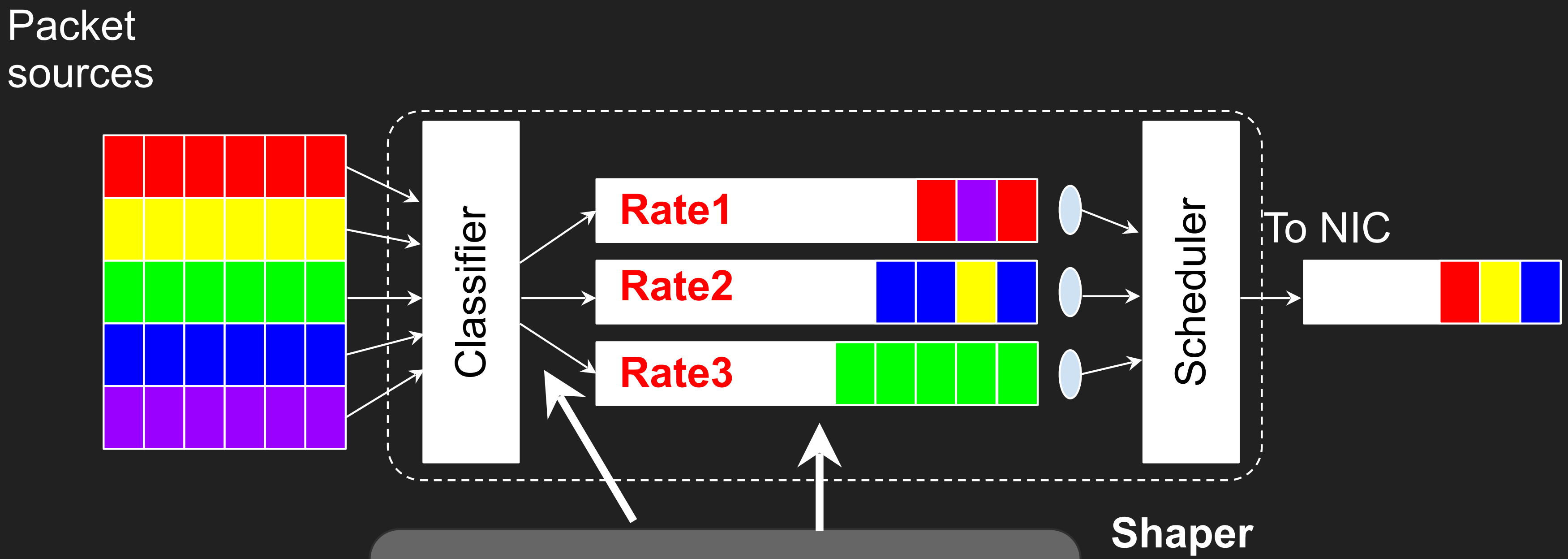
Overhead of managing a queue per configured rate

Traffic Shaping



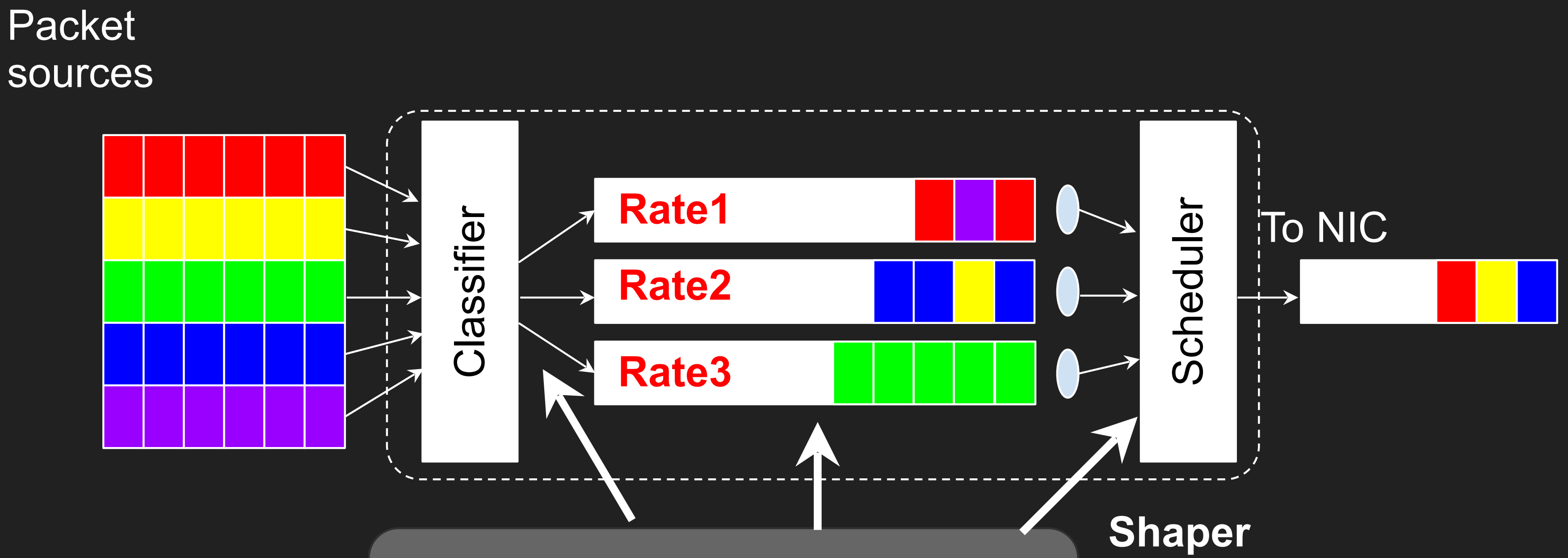
Overhead of managing a queue per configured rate

Traffic Shaping



Overhead of managing a queue per configured rate

Traffic Shaping

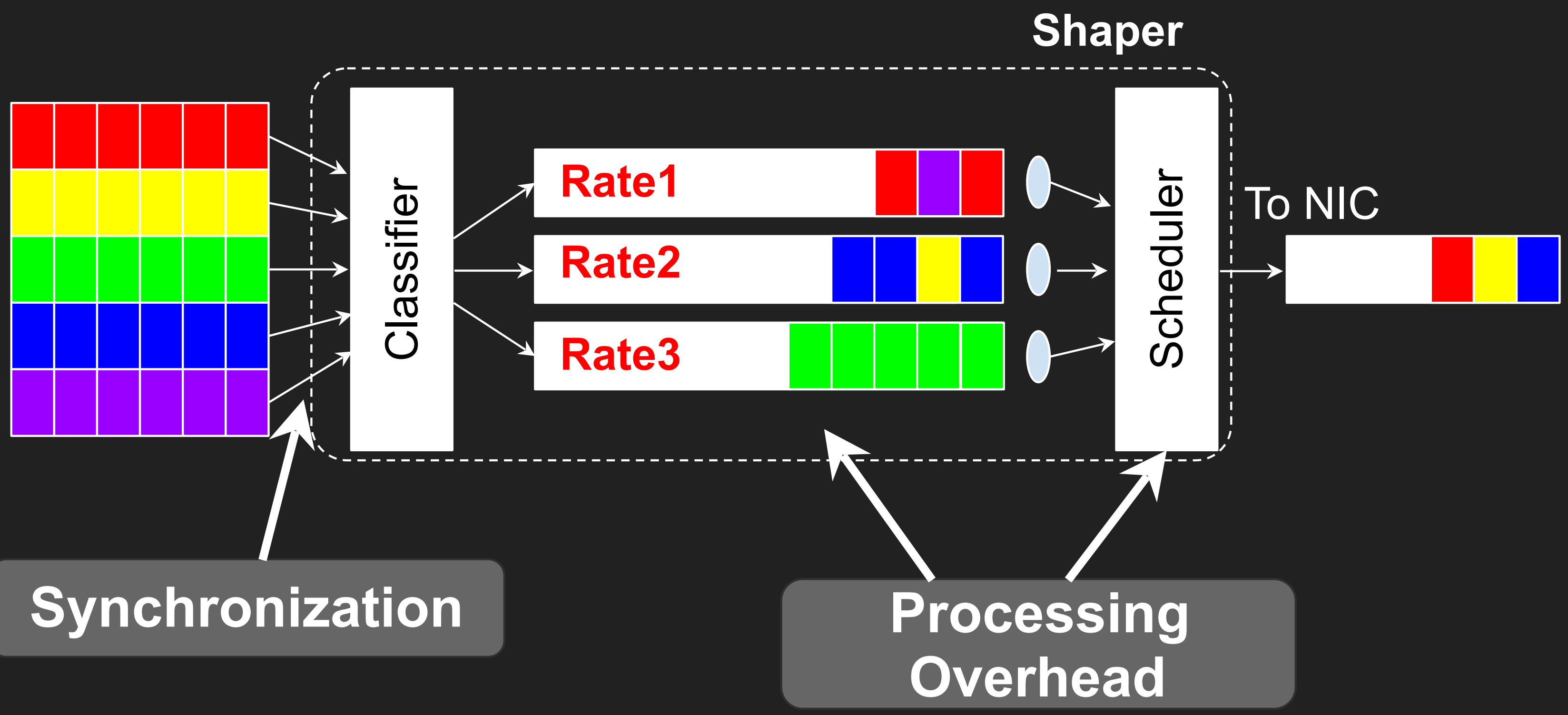


Overhead of managing a queue per configured rate

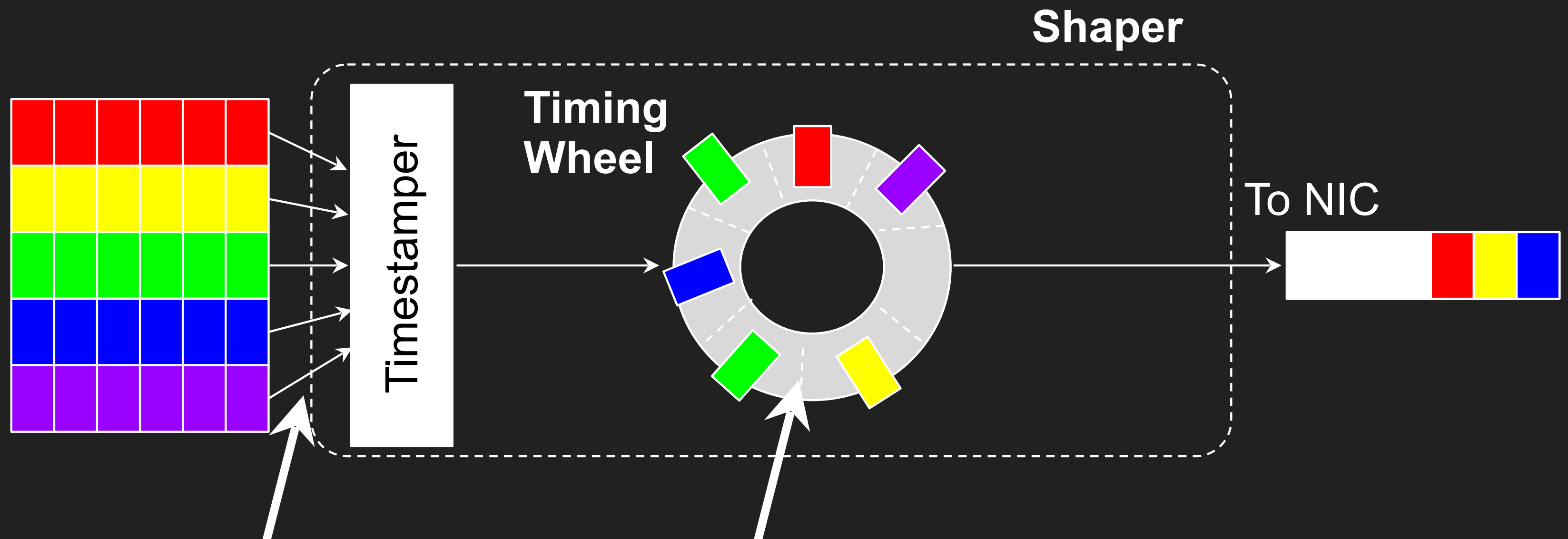
We need new traffic shapers that can handle tens of thousands of flows and rates

Main Idea💡
Replace the **many queues**
with **a single low-overhead queue**

Contributions



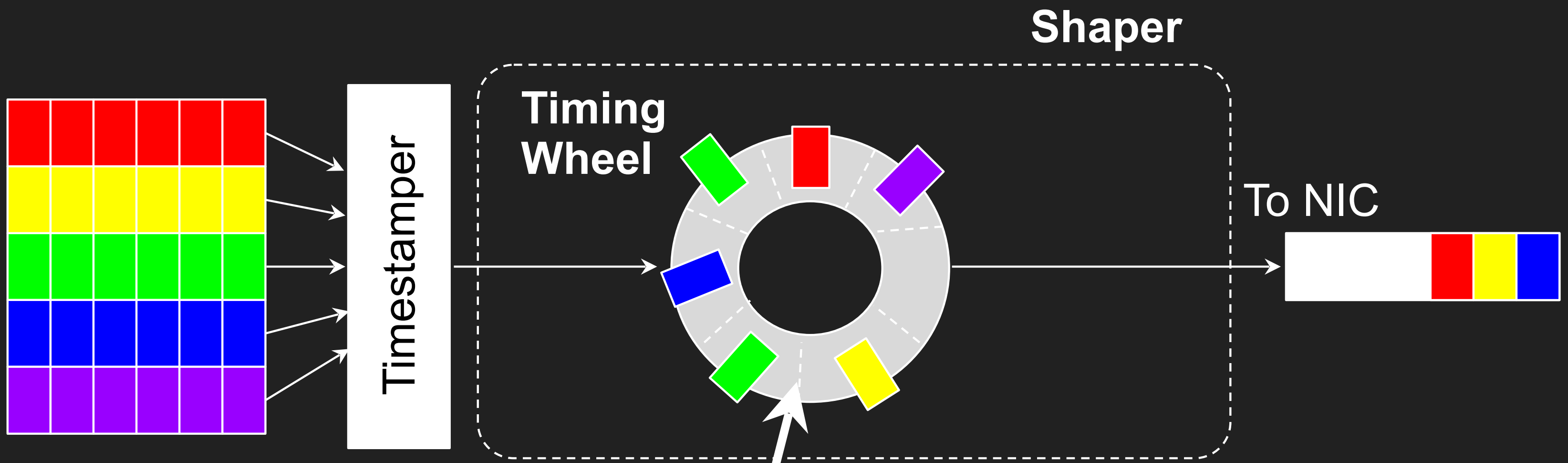
Contributions



Synchronization

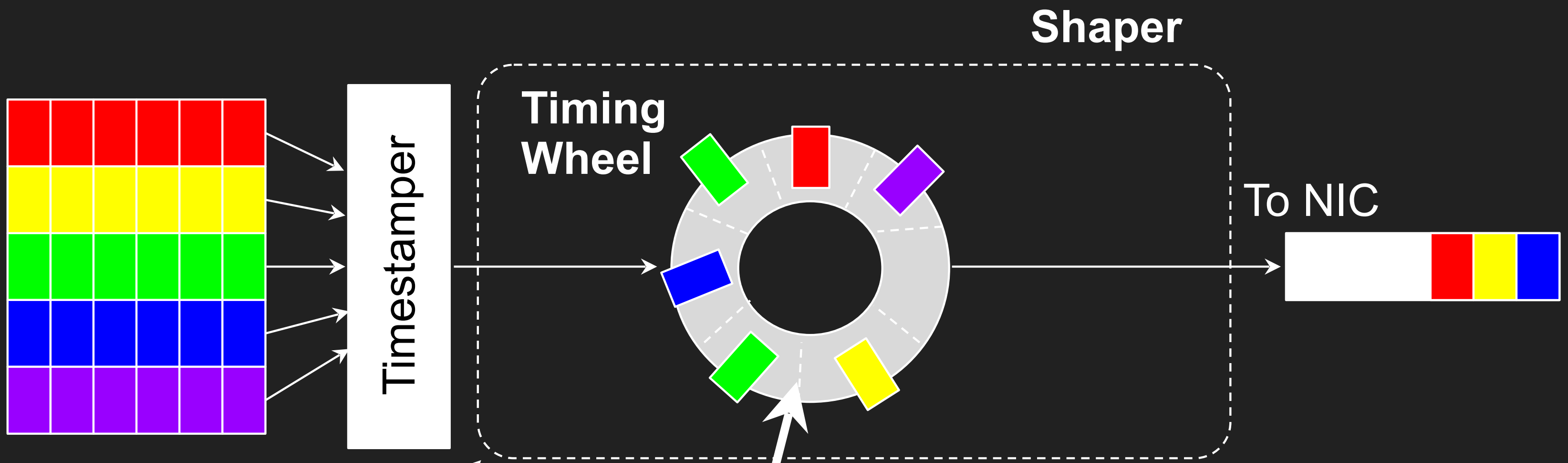
Single, O(1), Queue

Contributions



Single, O(1), Queue

Contributions

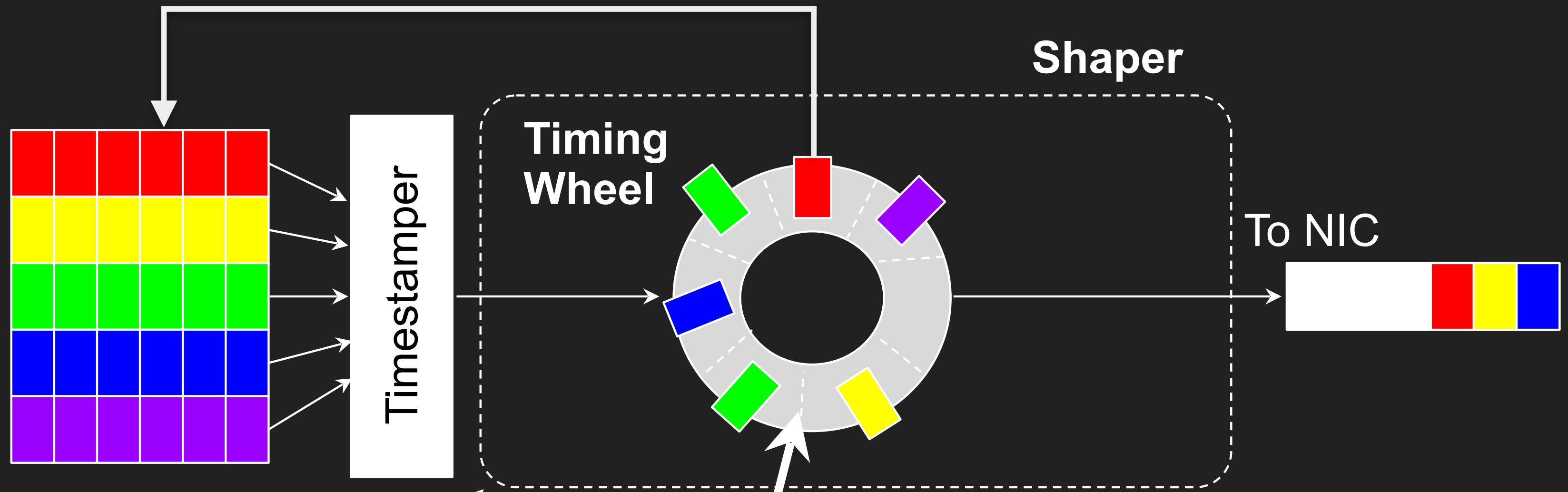


One Shaper per Core

Single, O(1), Queue

Contributions

Apply backpressure



One Shaper per Core

Single, O(1), Queue

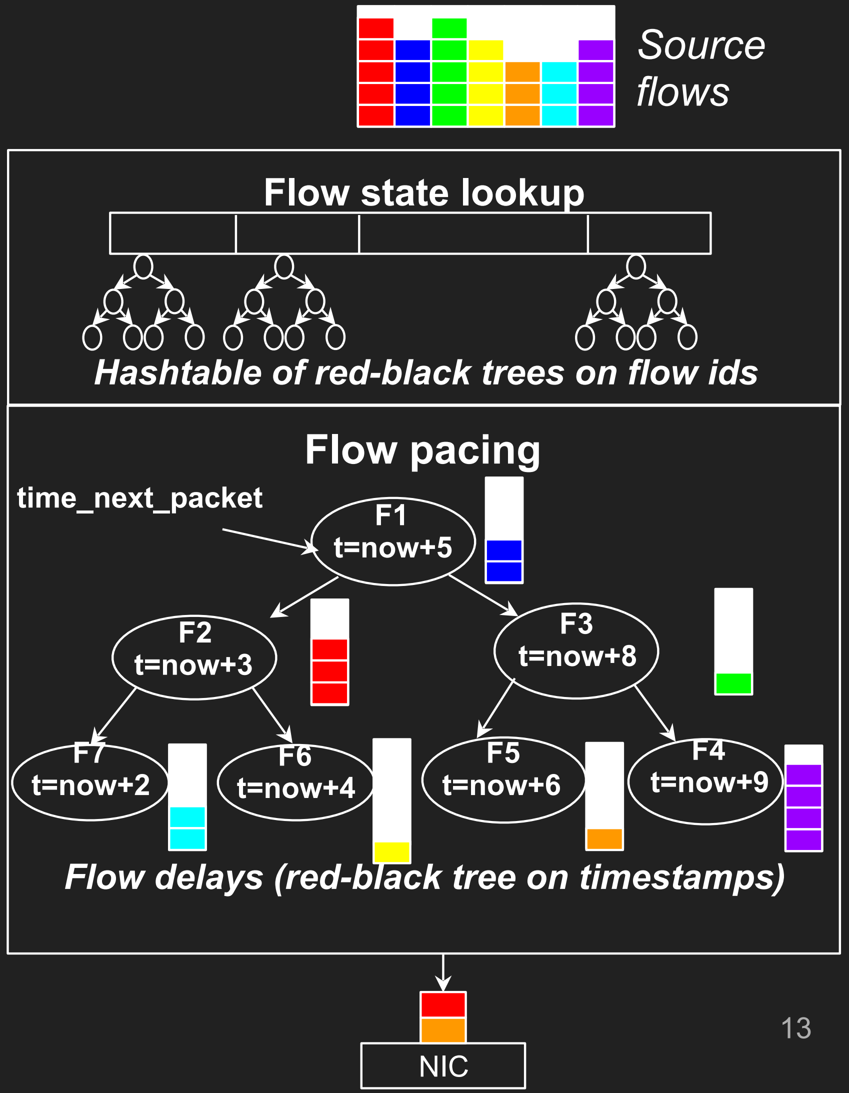
Outline

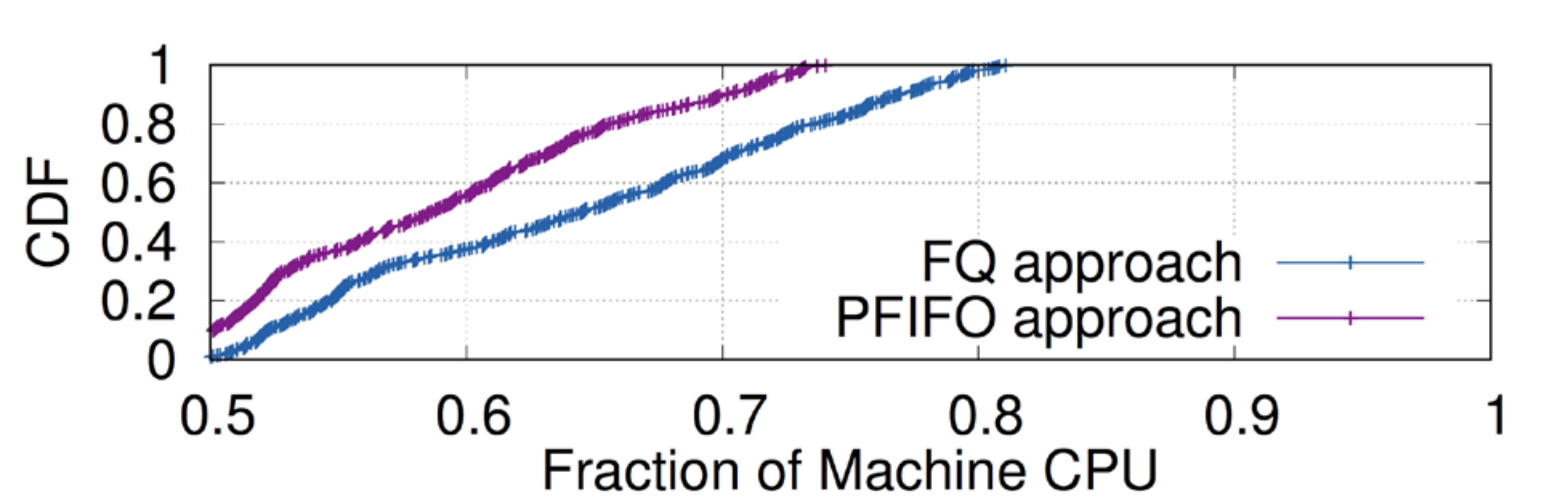
- Problems with Current Shapers
- Carousel Overview
- Single Queue Shaping
- Backpressure
- Evaluation

Problems with Current Shapers

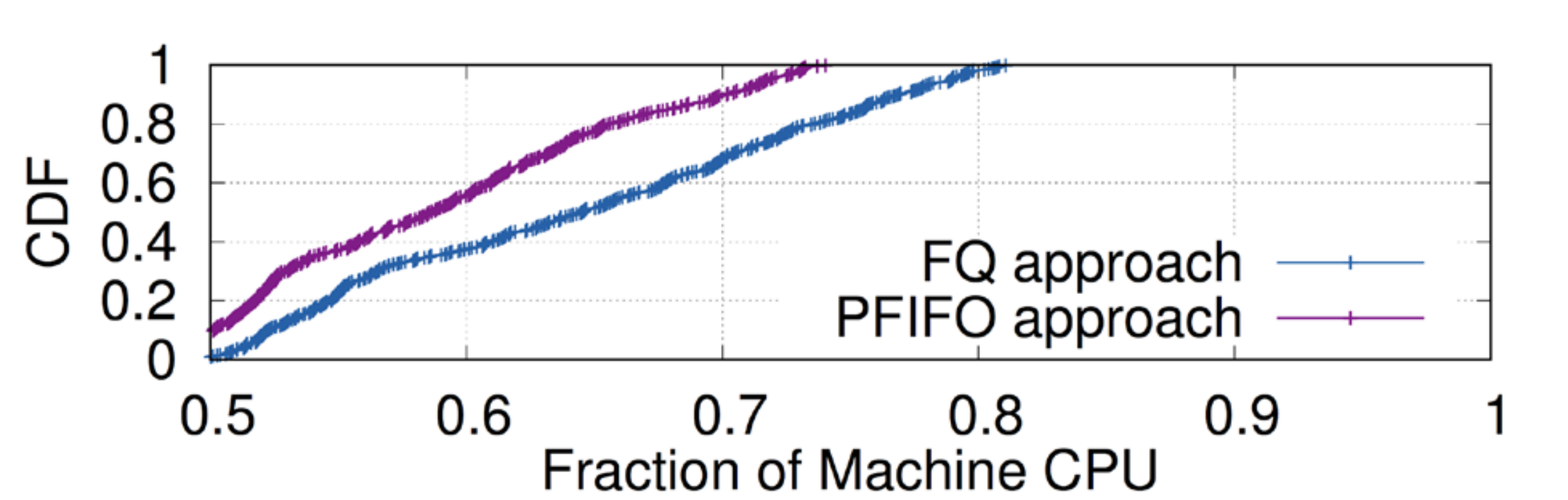
FQ/Pacing

- Implements per TCP flow pacing
- Requires a queue per flow
    - Flows are kept in order of their scheduled transmission time
    - Flows are dequeued in order
- O(log n) operations per packet to operate on a sorted list of flows





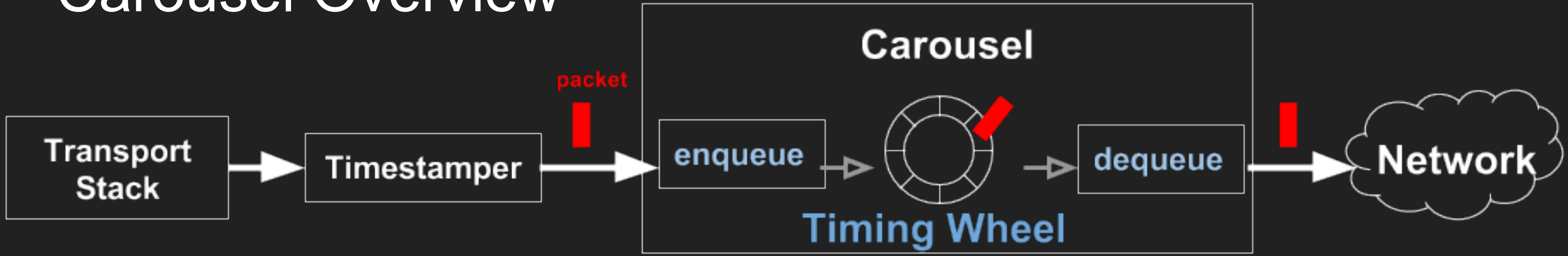
CPU utilization for FQ/pacing and a NOOP Qdisc for the same load



FQ/Pacing introduces 10% more CPU overhead

Carousel Overview

Carousel Overview



- Relies on a single queue for all packets from all flows
- Requires a high frequency timer or busy polling
- Pinned to a single core

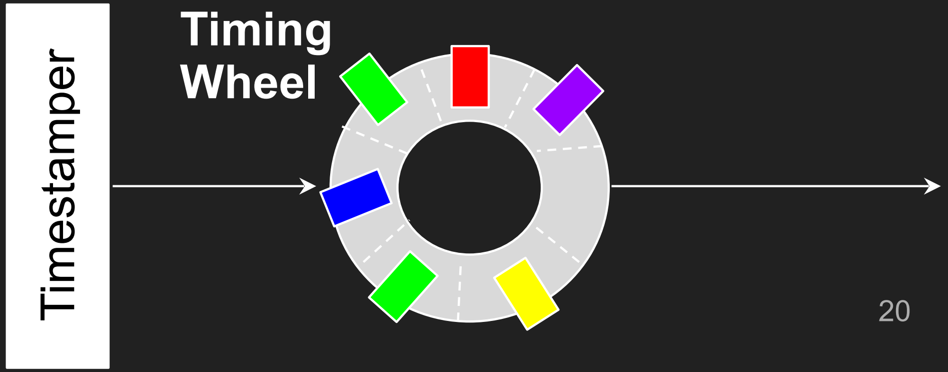
Single Queue Shaping

Single Queue Shaping

- All packet are sorted by their transmission time in one data structure
- A single queue for all traffic will need to handle tens of thousands of packets
- **Challenge:**
  Enqueue and dequeue in a data structure of sorted elements at line rate

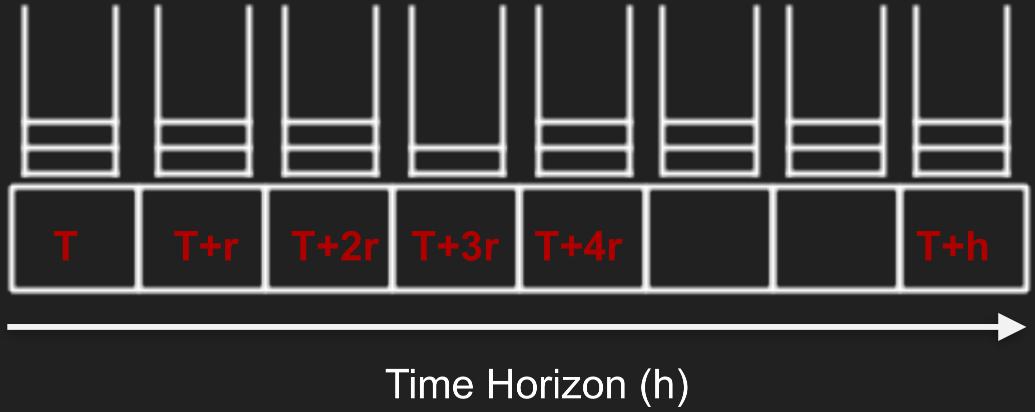
Single Queue Shaping

- All packet are sorted by their transmission time in one data structure
- A single queue for all traffic will need to handle tens of thousands of packets
- **Challenge:**
  Enqueue and dequeue in a data structure of sorted elements at line rate



Timing Wheel [Varghese et al. SOSP ’87]

- Bucket sort approach to Calendar Queue covering a time horizon
  - Relies on having a minimum rates
- Implemented as an array of buckets each a linked list of packets
  - Each bucket represents a certain time range

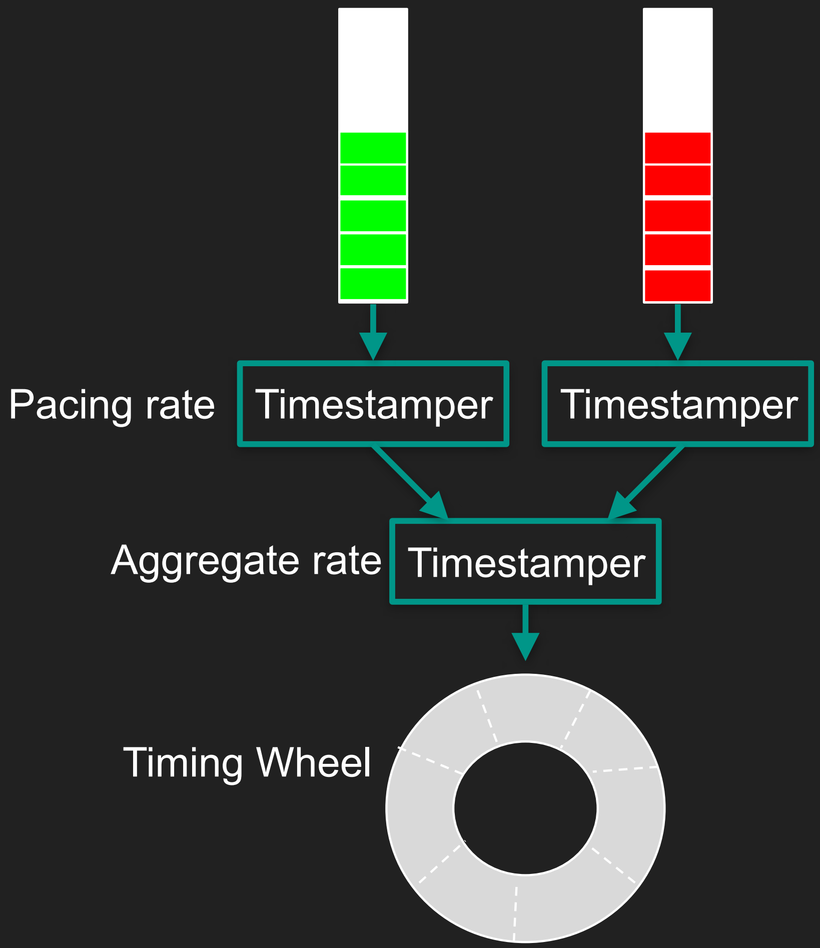


Timing Wheel Benchmark

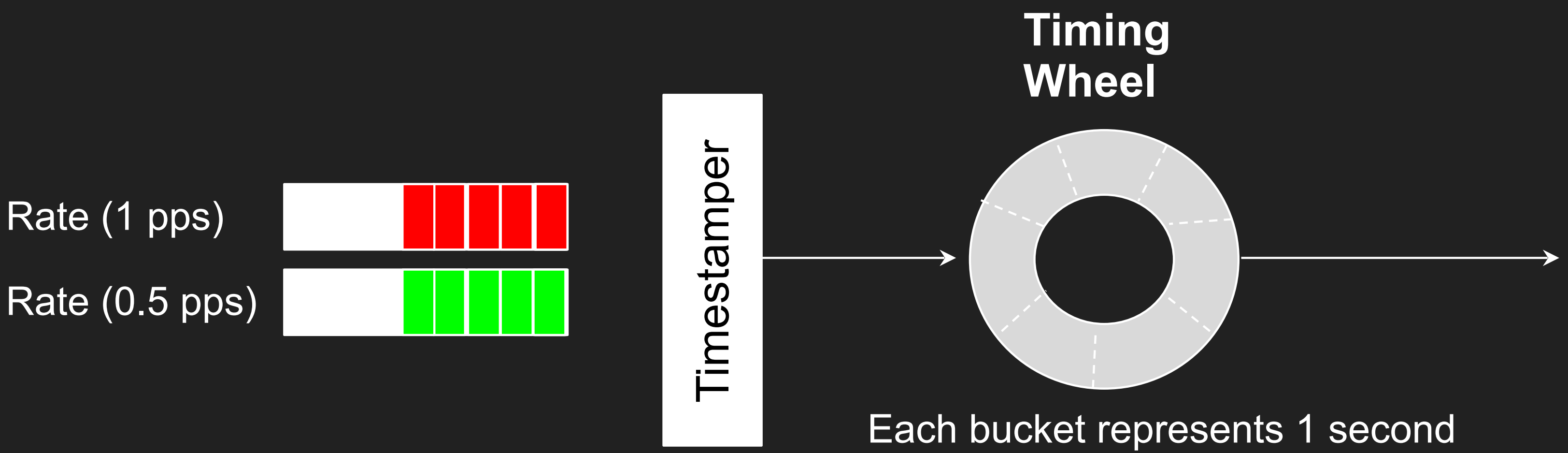
- Measured overhead per enqueue/dequeue pairs
- Overhead per element is between 21-22 nanoseconds
  - Fixed for 2000 to 2 million sorted elements
  - 21 nanoseconds per packet = 500 Gbps (for 1500 byte packets)

Timestampers

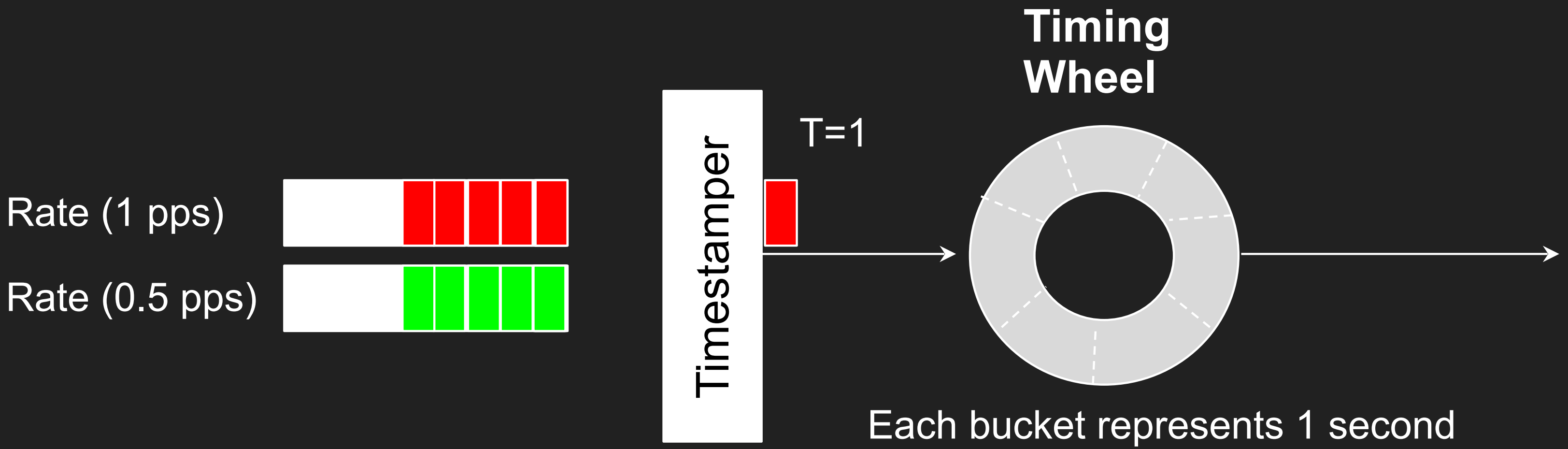
- Packets are timestamped by policy enforcers in their transmission path
  - TCP timestamps a packet based on its pacing rate
  - Bandwidth enforcer timestamps a packet based on its policy-based aggregate rate
- Carousel picks the largest timestamp
- NextTimestamp = LastTimestamp + $\frac{\text{SizeOfPacket}}{\text{ConfiguredRate}}$



Example of Shaping using Carousel

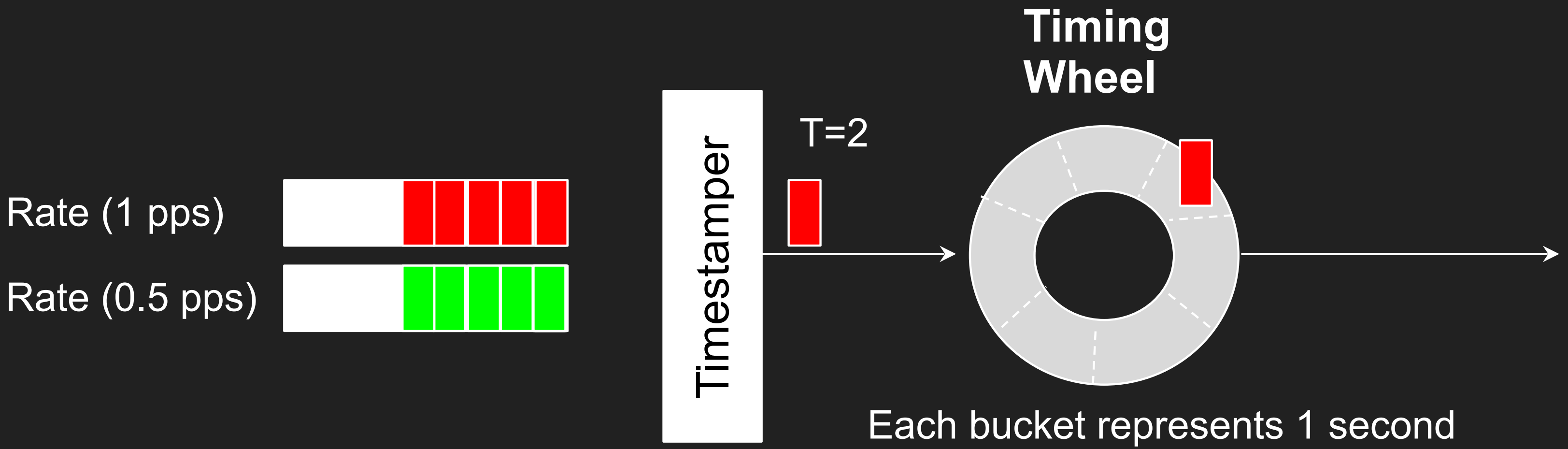


Example of Shaping using Carousel



A time step **0**

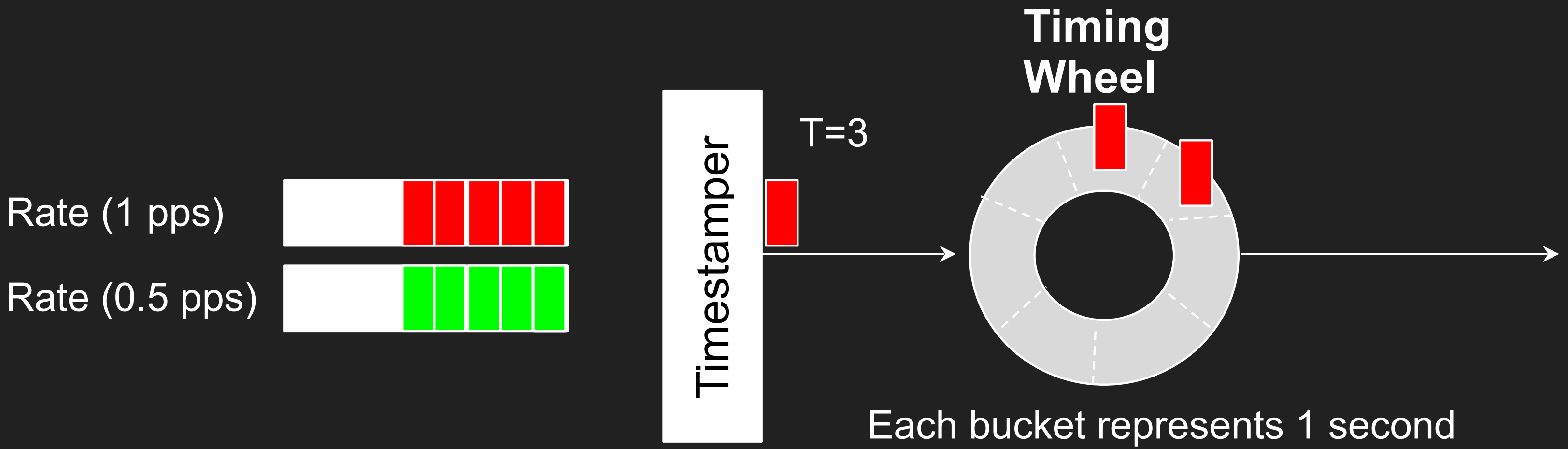
Example of Shaping using Carousel



Each bucket represents 1 second

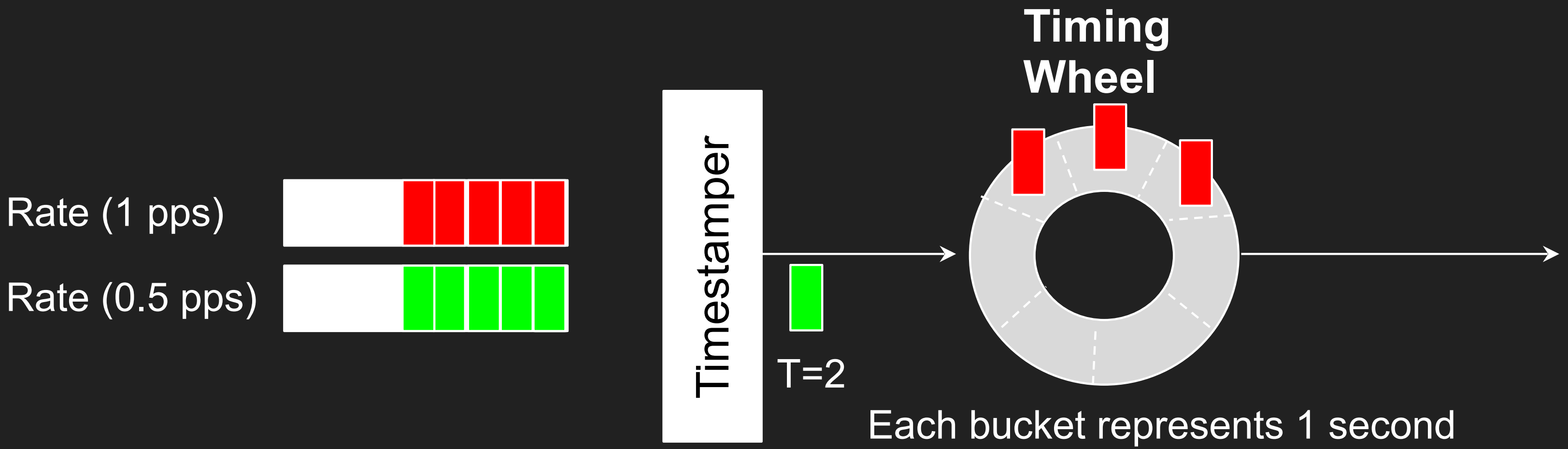
A time step **0**

Example of Shaping using Carousel



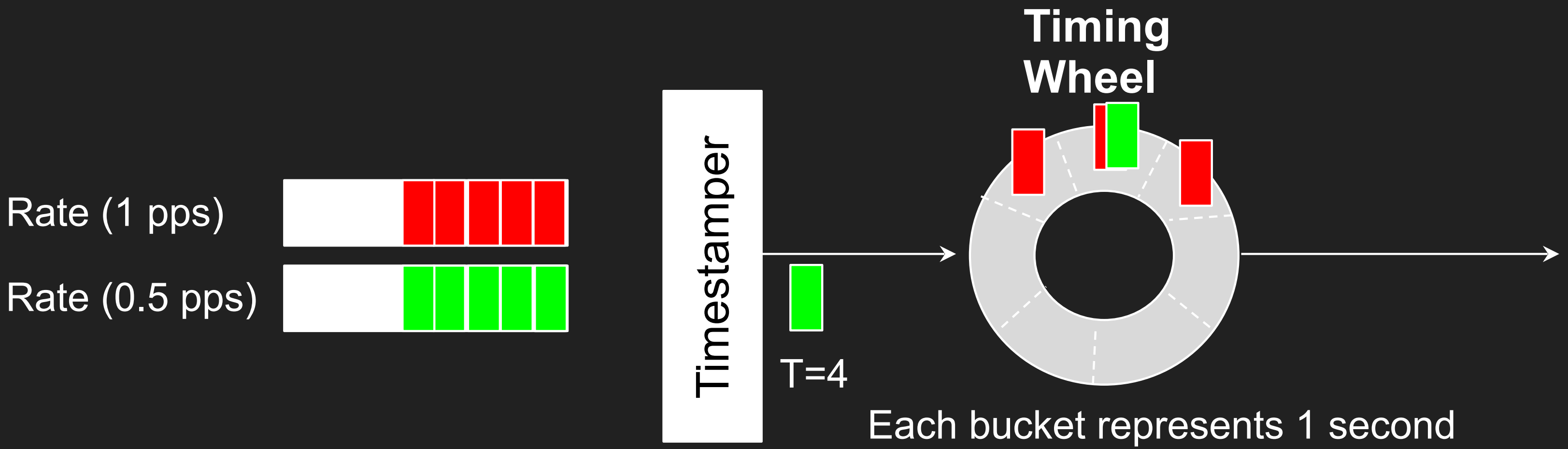
A time step **0**

Example of Shaping using Carousel



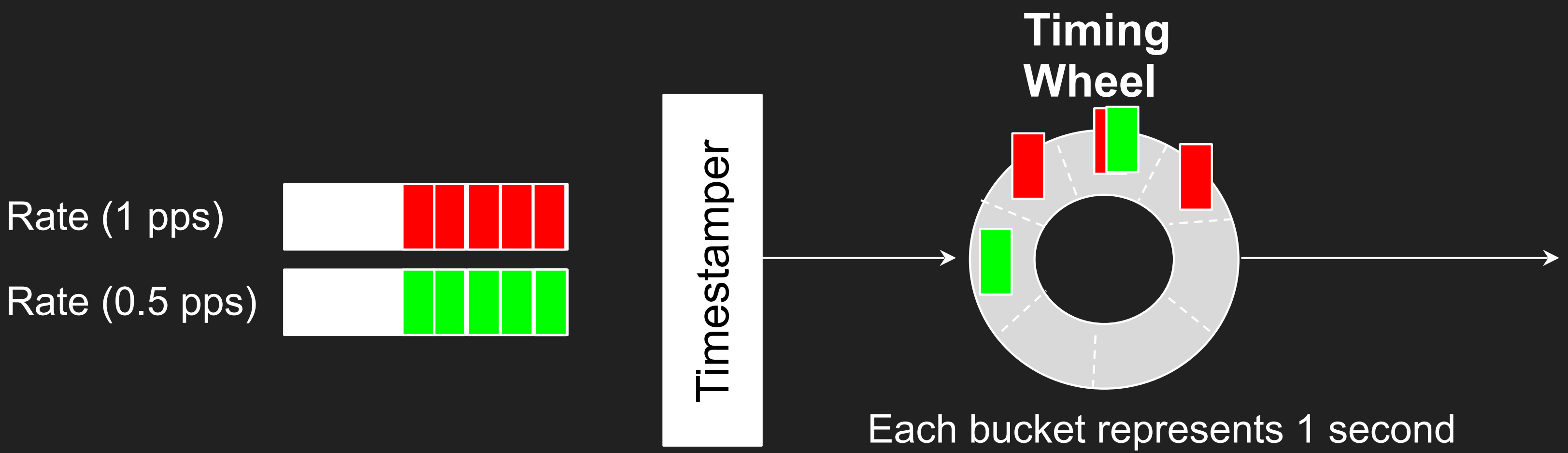
A time step **0**

Example of Shaping using Carousel



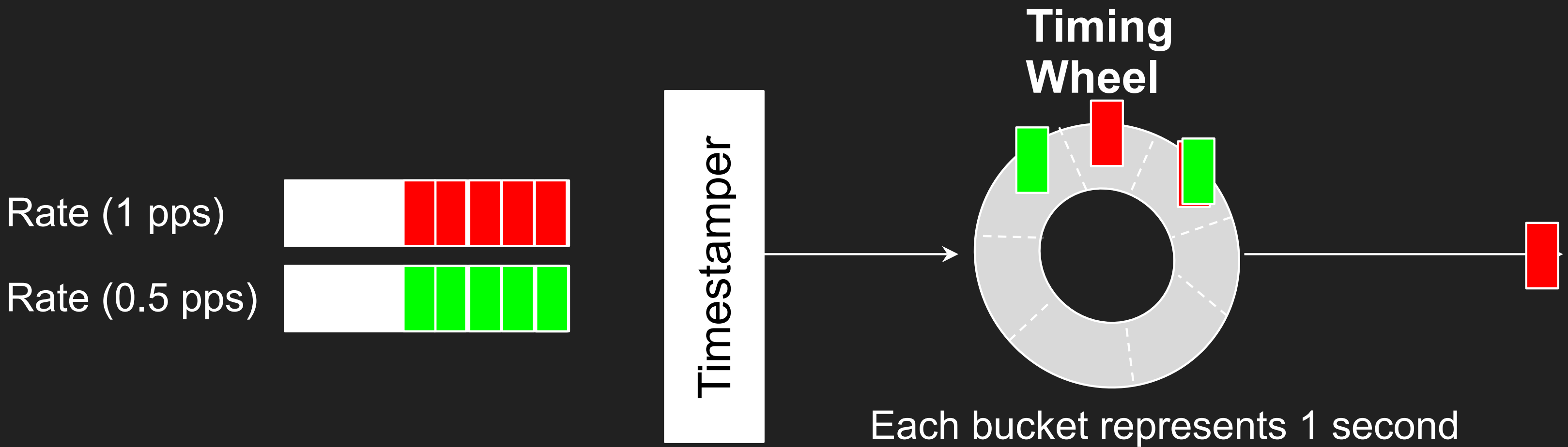
A time step **0**

Example of Shaping using Carousel



A time step **0**

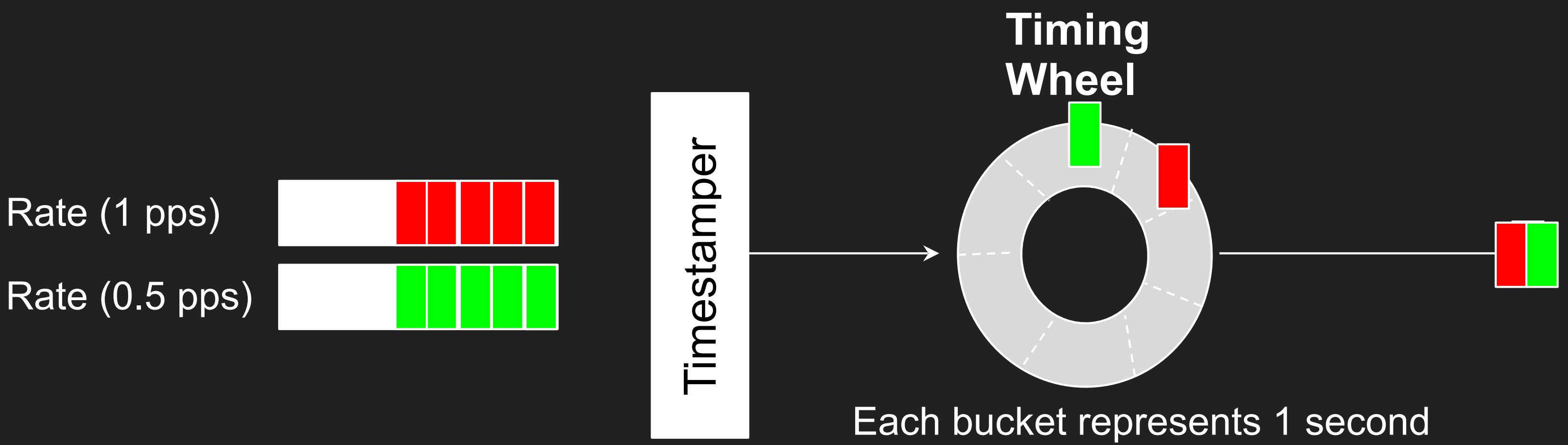
Example of Shaping using Carousel



Each bucket represents 1 second

A time step **1**

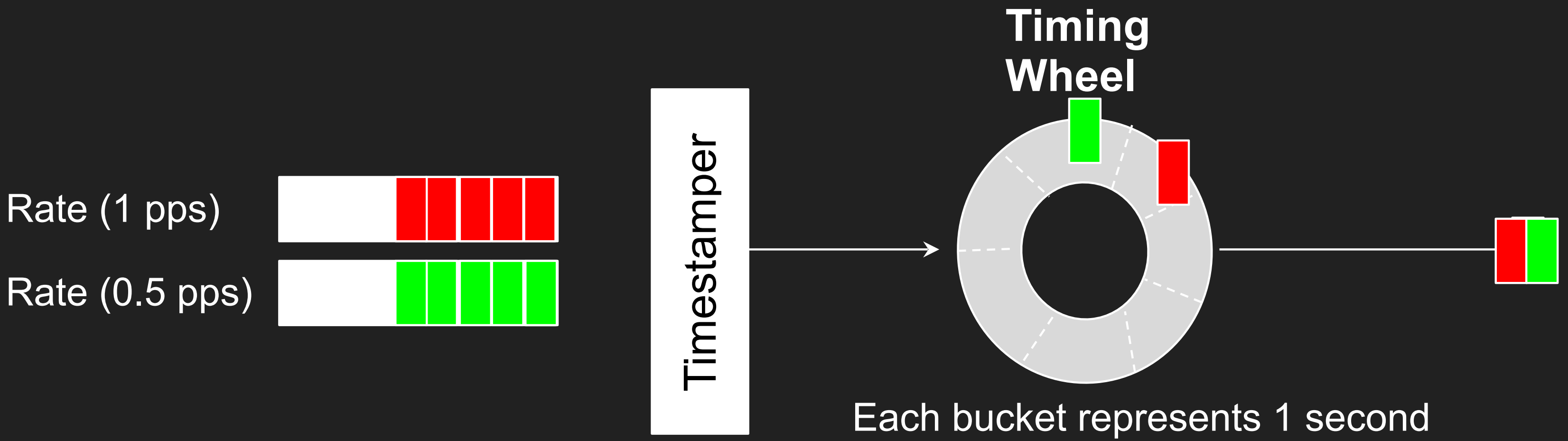
Example of Shaping using Carousel



Each bucket represents 1 second

A time step **2**

Example of Shaping using Carousel



Rate (1 pps)

Rate (0.5 pps)

Timestamper

Timing Wheel

Each bucket represents 1 second

Backpressure with Deferred Completion

The Value of Backpressure

- Without backpressure shaper queues get full with small number of flows causing

The Value of Backpressure

- Without backpressure shaper queues get full with small number of flows causing
  - Unnecessary drops (when the queue is full the queue tail drops)

The Value of Backpressure

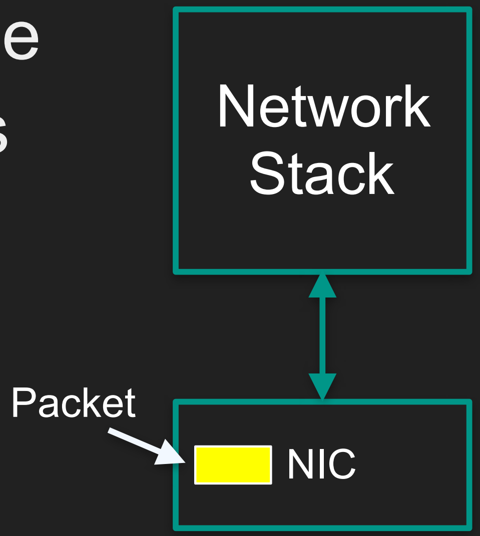
- Without backpressure shaper queues get full with small number of flows causing
  - Unnecessary drops (when the queue is full the queue tail drops)
  - Head of Line Blocking

The Value of Backpressure

- Without backpressure shaper queues get full with small number of flows causing
  - Unnecessary drops (when the queue is full the queue tail drops)
  - Head of Line Blocking
- Backpressure allows shapers to control sender rate and avoid overwhelming the shaper

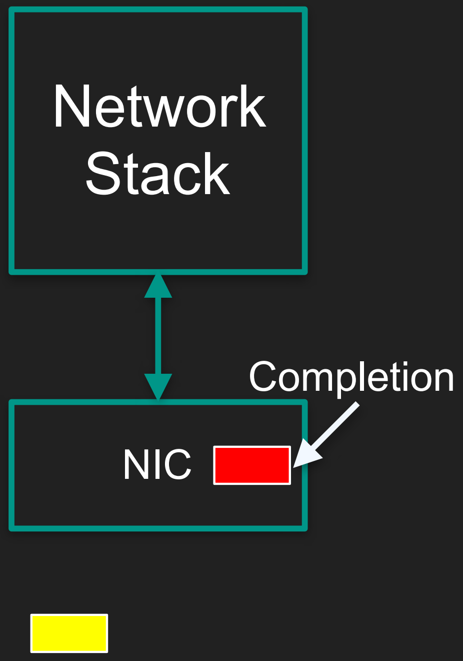
The Completion Signal

- Completions are signals from the NIC to the network stack to inform it that a packet has been transmitted



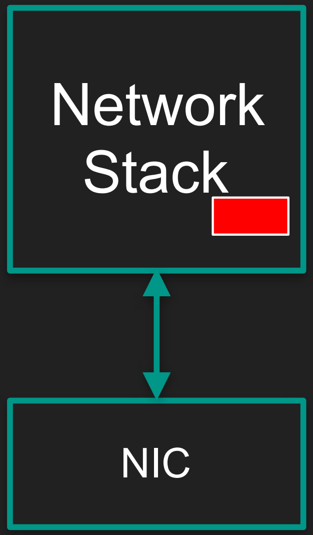
The Completion Signal

- Completions are signals from the NIC to the network stack to inform it that a packet has been transmitted



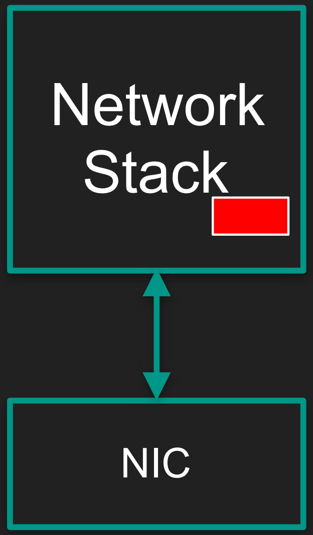
The Completion Signal

- Completions are signals from the NIC to the network stack to inform it that a packet has been transmitted



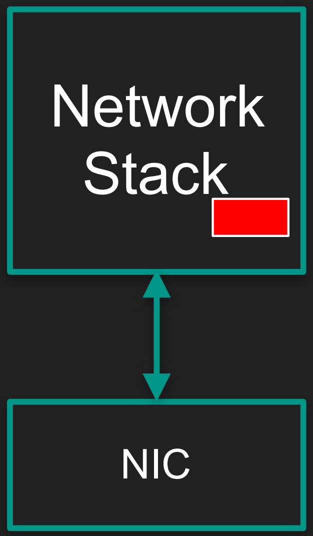
The Completion Signal

- Completions are signals from the NIC to the network stack to inform it that a packet has been transmitted
  - Completions are typically delivered in order



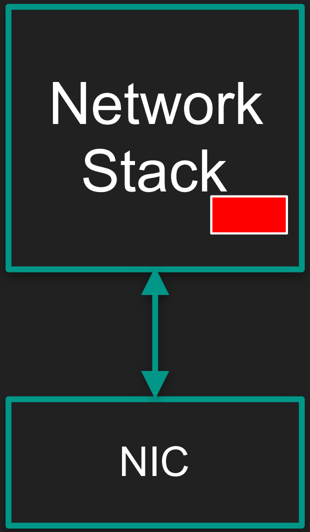
The Completion Signal

- Completions are signals from the NIC to the network stack to inform it that a packet has been transmitted
  - Completions are typically delivered in order
  - Completion should be controlled by the hypervisor not the virtual NIC



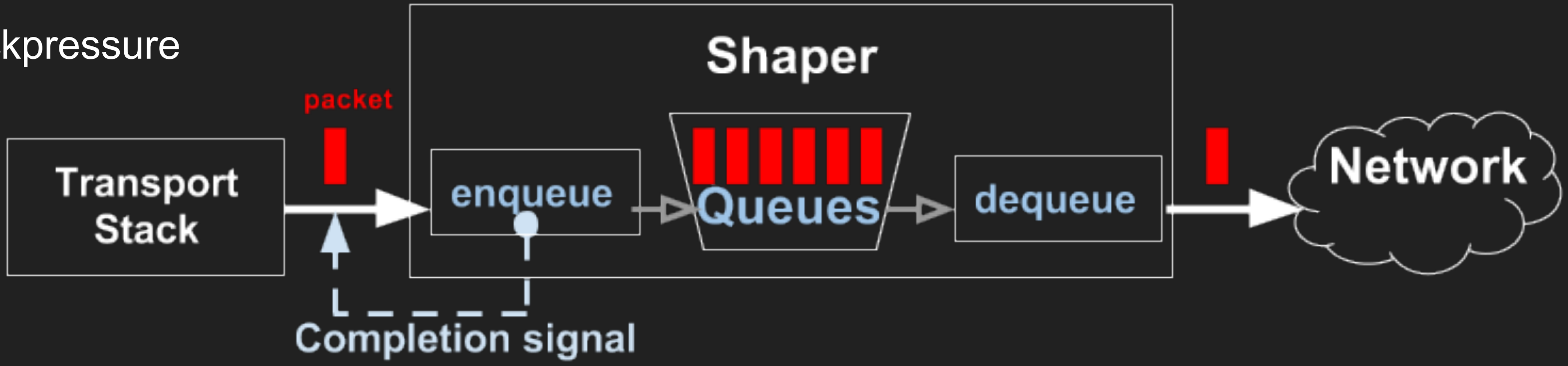
The Completion Signal

- Completions are signals from the NIC to the network stack to inform it that a packet has been transmitted
  - Completions are typically delivered in order
  - Completion should be controlled by the hypervisor not the virtual NIC
- Completions should be delivered out of order and completely controlled by Shapers



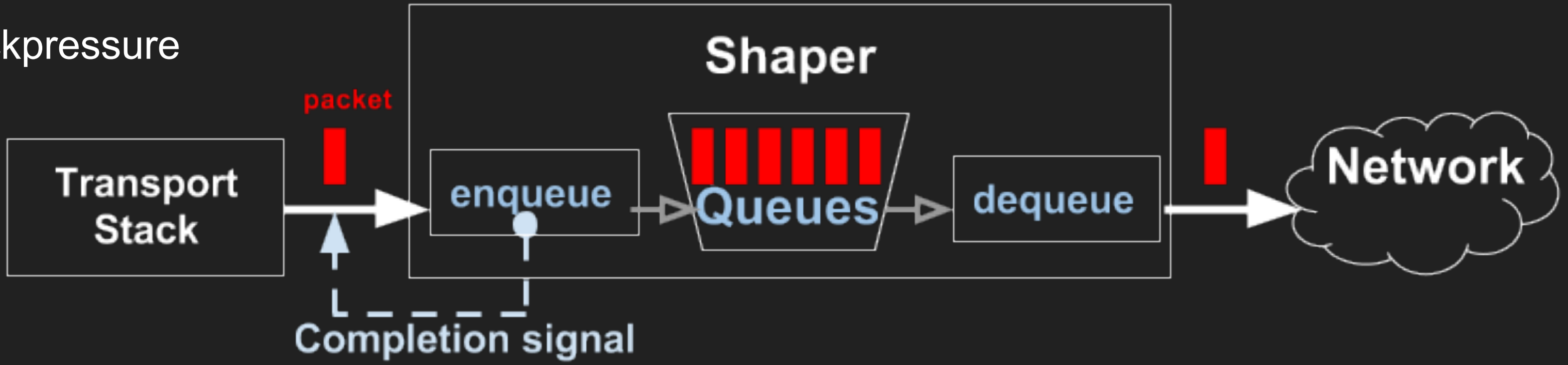
Backpressure with Deferred Completion

Without Backpressure

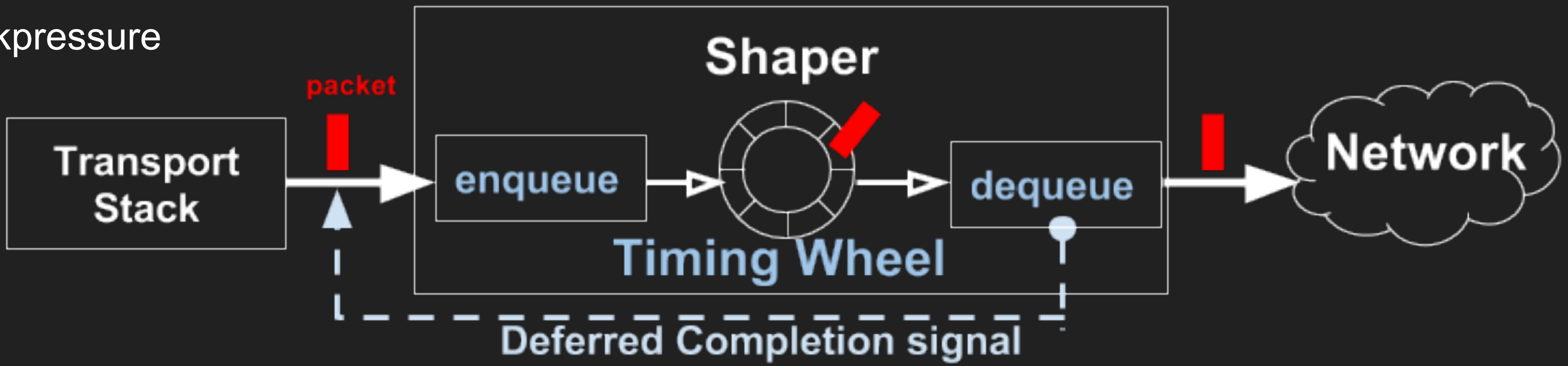


Backpressure with Deferred Completion

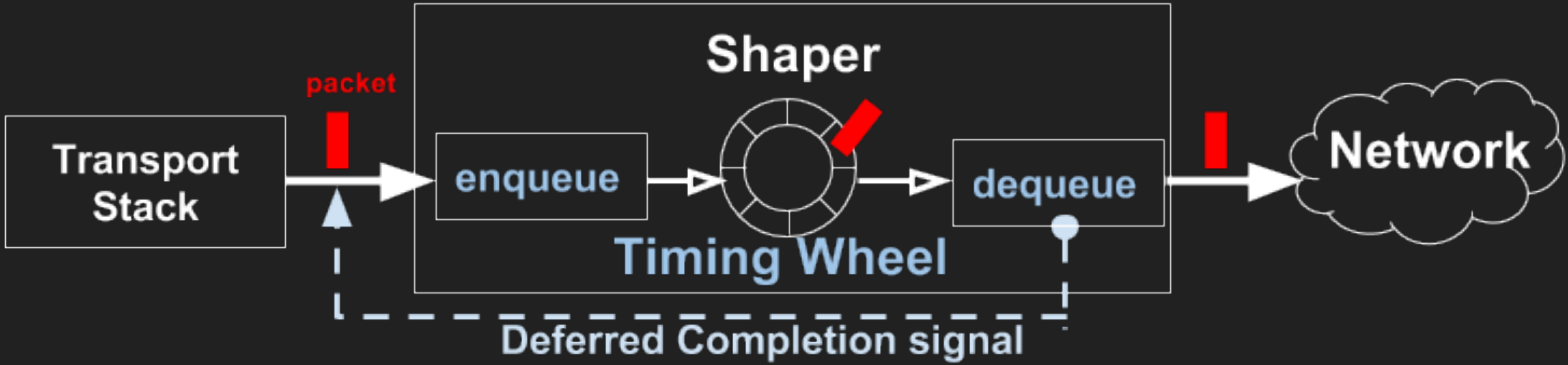
Without Backpressure

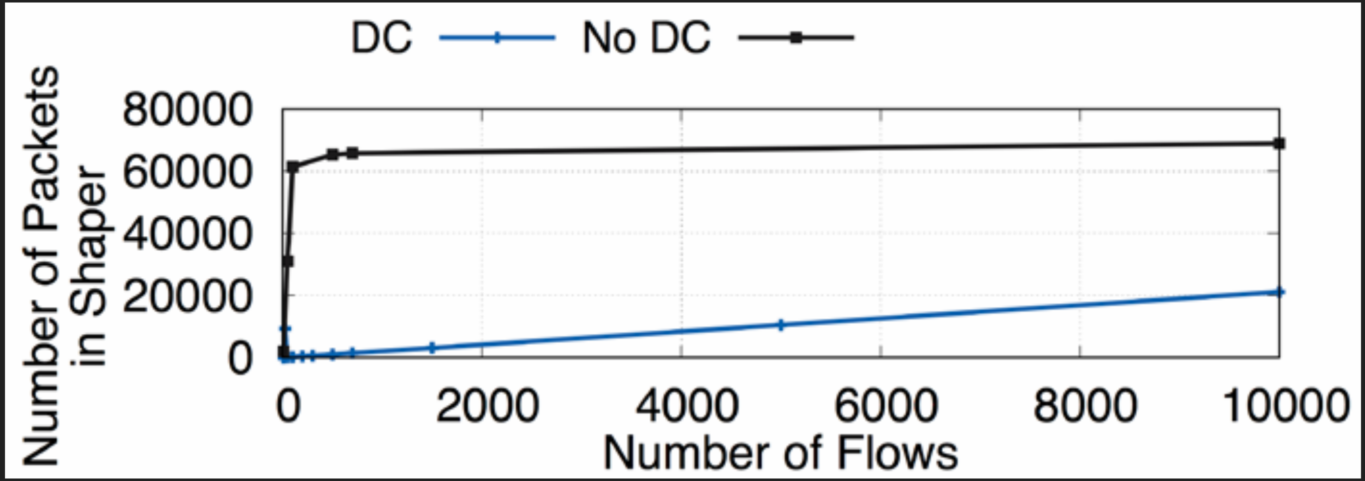


With Backpressure

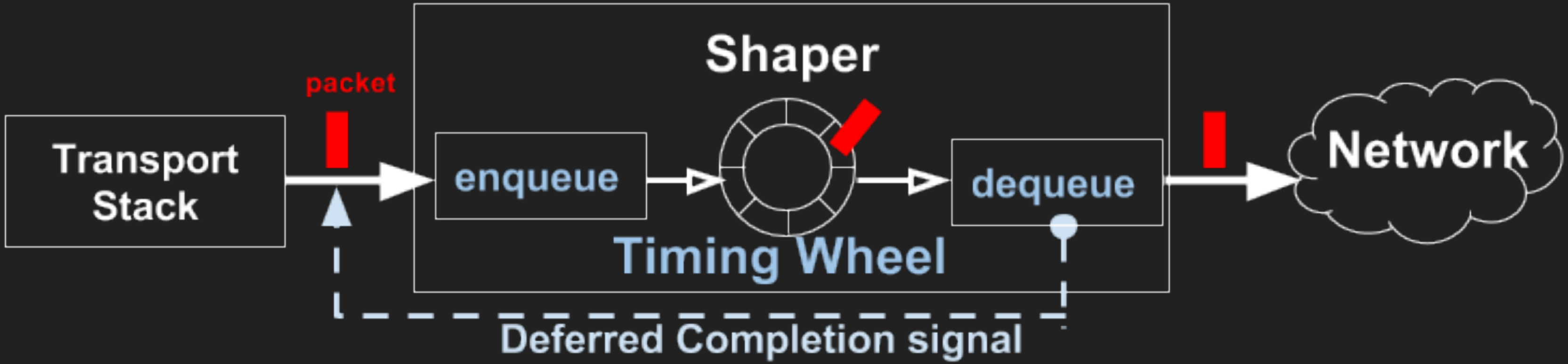


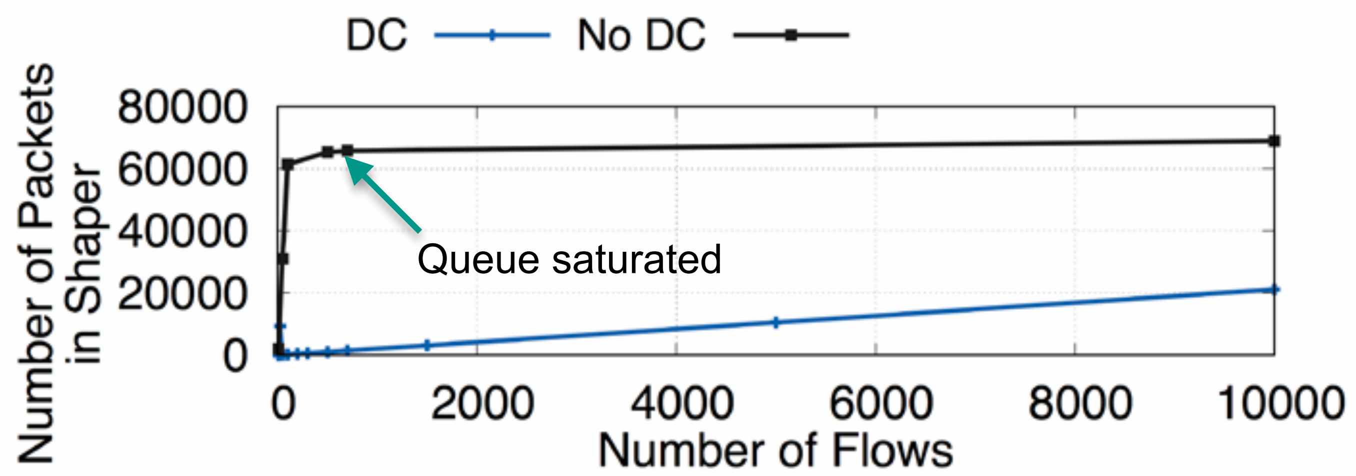
Backpressure with Deferred Completion



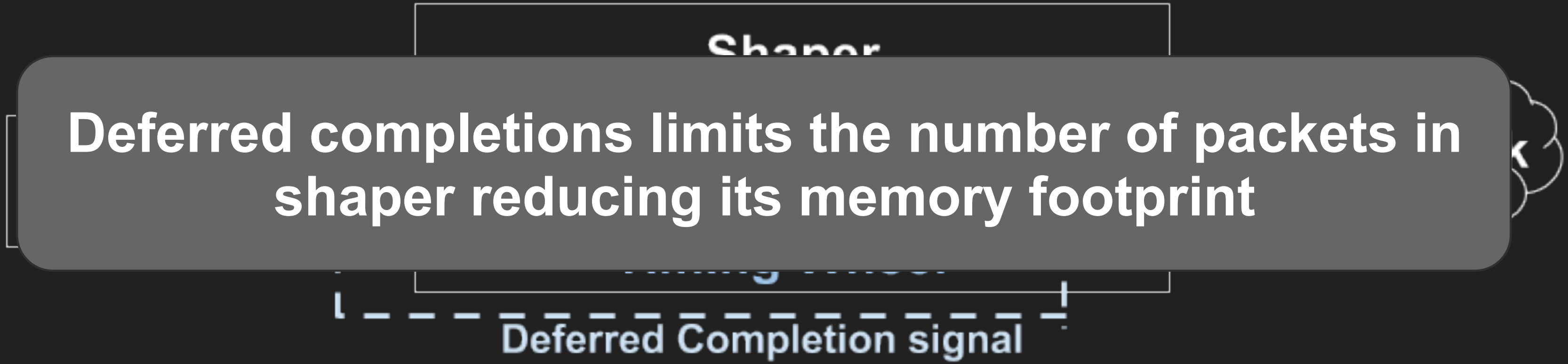


Backpressure with Deferred Completion

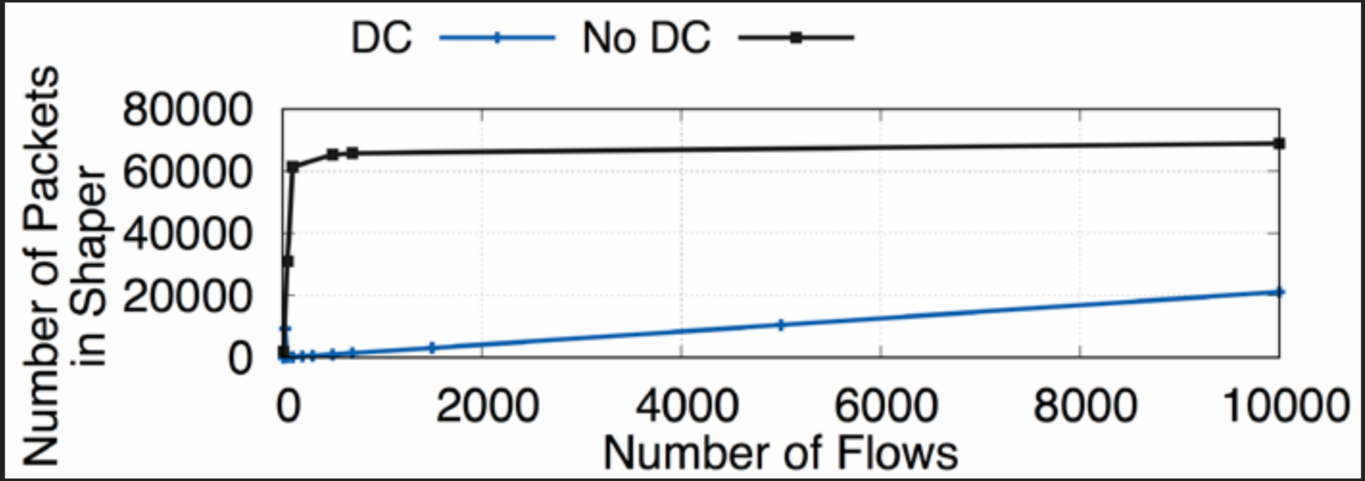




Backpressure with Deferred Completion



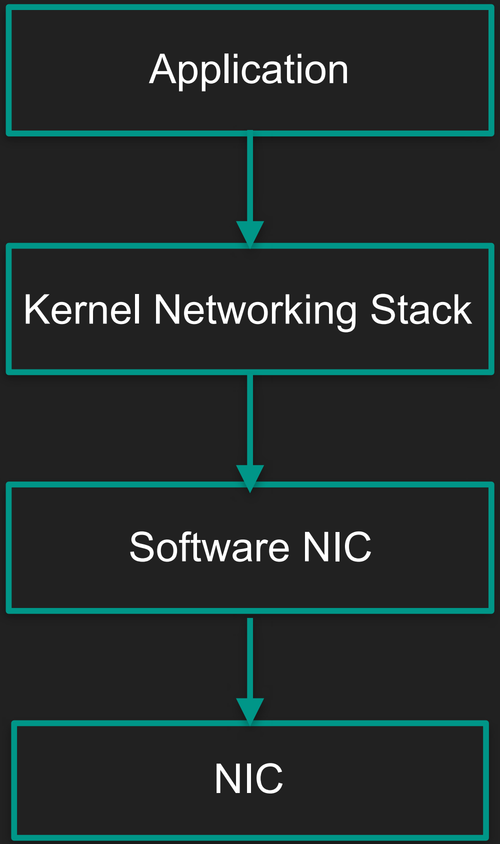
Deferred completions limits the number of packets in shaper reducing its memory footprint



Evaluation

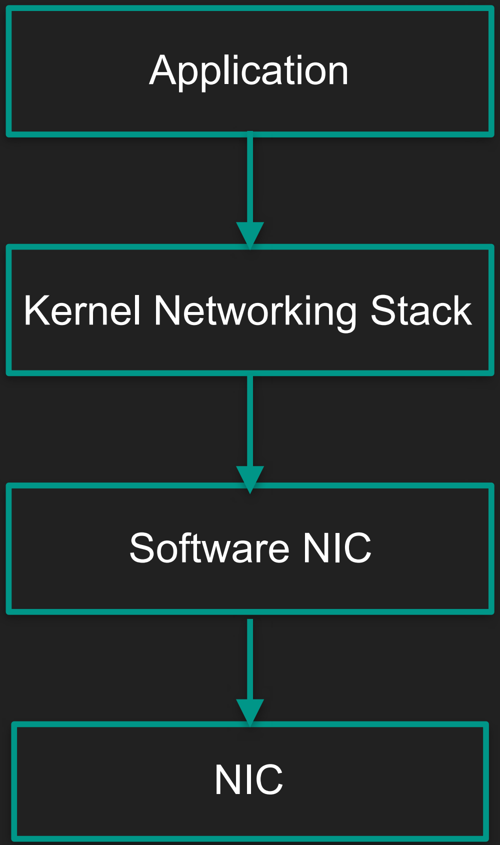
Evaluation Setup

- Carousel deployed within a Software NIC



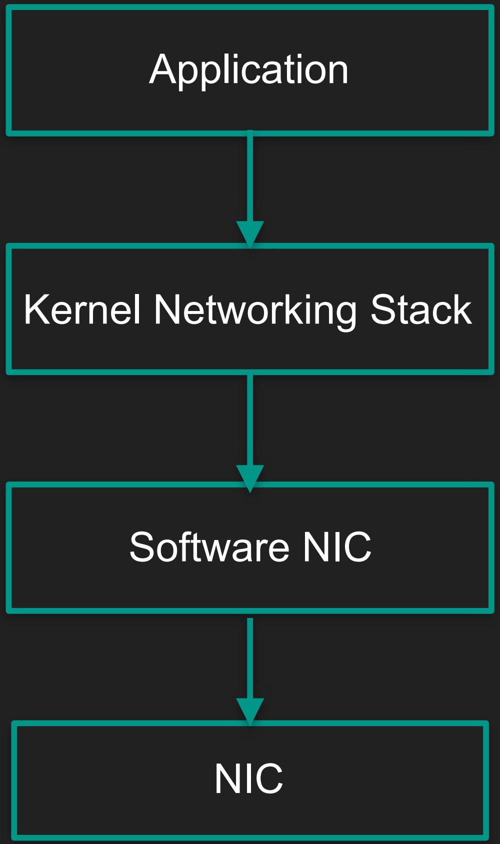
Evaluation Setup

- Carousel deployed within a Software NIC
- Evaluation on Youtube servers comparing Carousel and FQ/Pacing



Evaluation Setup

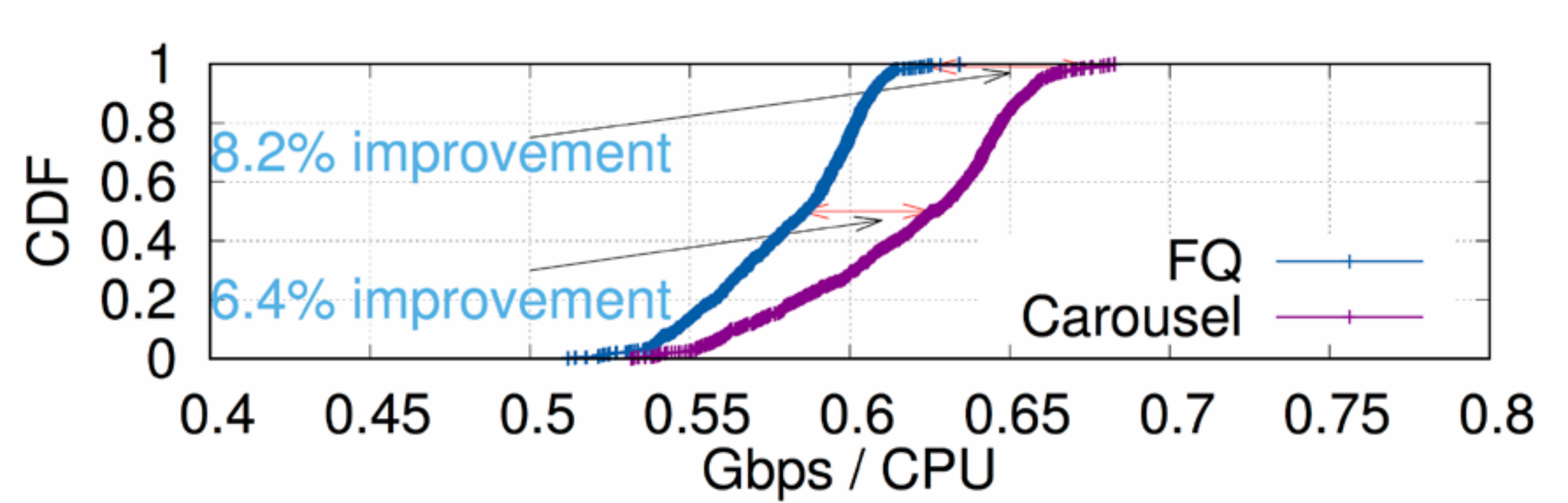
- Carousel deployed within a Software NIC
- Evaluation on Youtube servers comparing Carousel and FQ/Pacing
- Each server handles up to 50k sessions concurrently



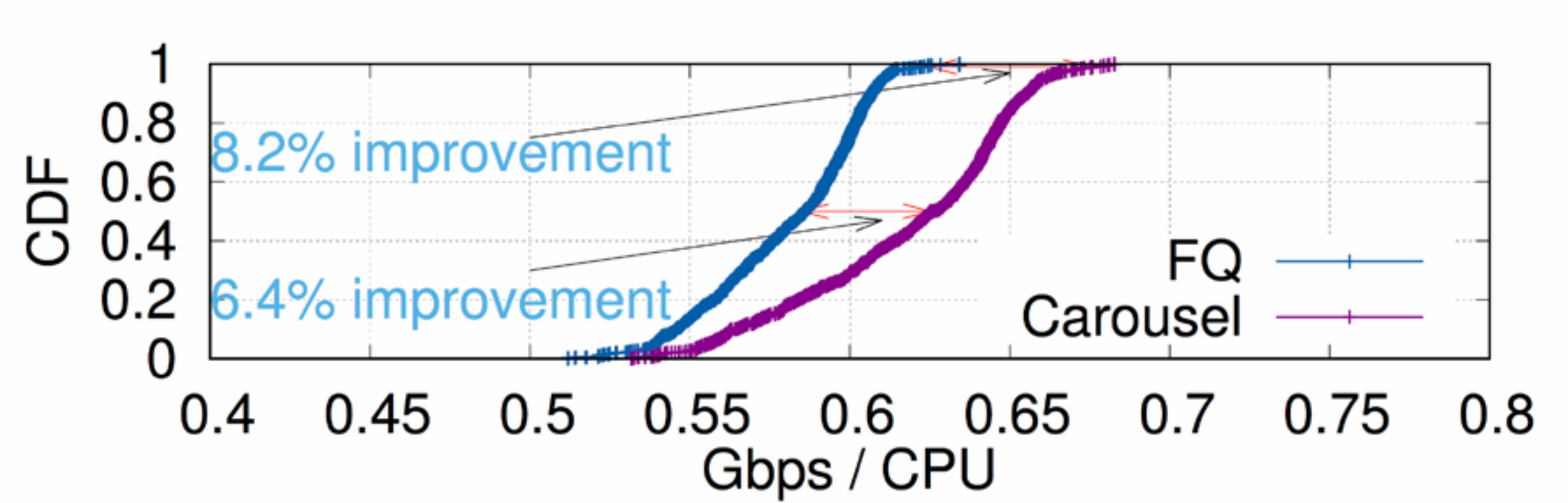
Evaluation Metric

- Measures Gbps served per CPU utilization
  - Metric used is Gbps/CPU (higher is better)
  - Compare machines with similar CPU utilization
  - Measurements performed during peak 12-hours per day
- Evaluation is performed for:
  - Overall CPU utilization
  - Software NIC utilization

Overall CPU Utilization

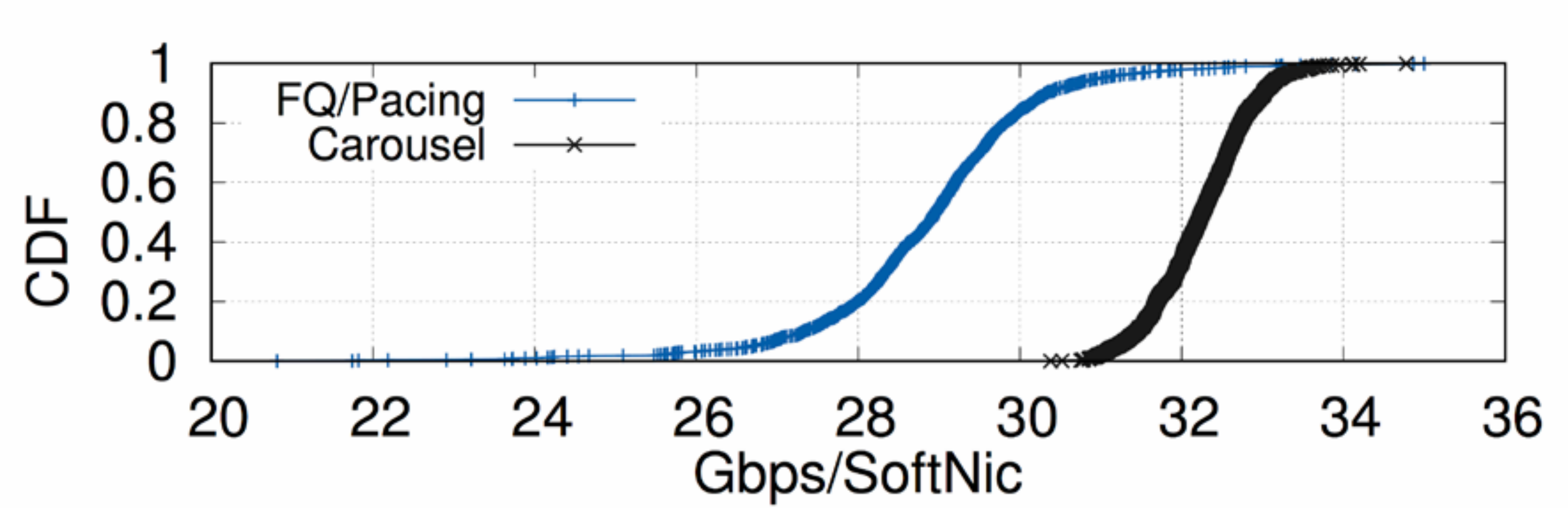


Overall CPU Utilization

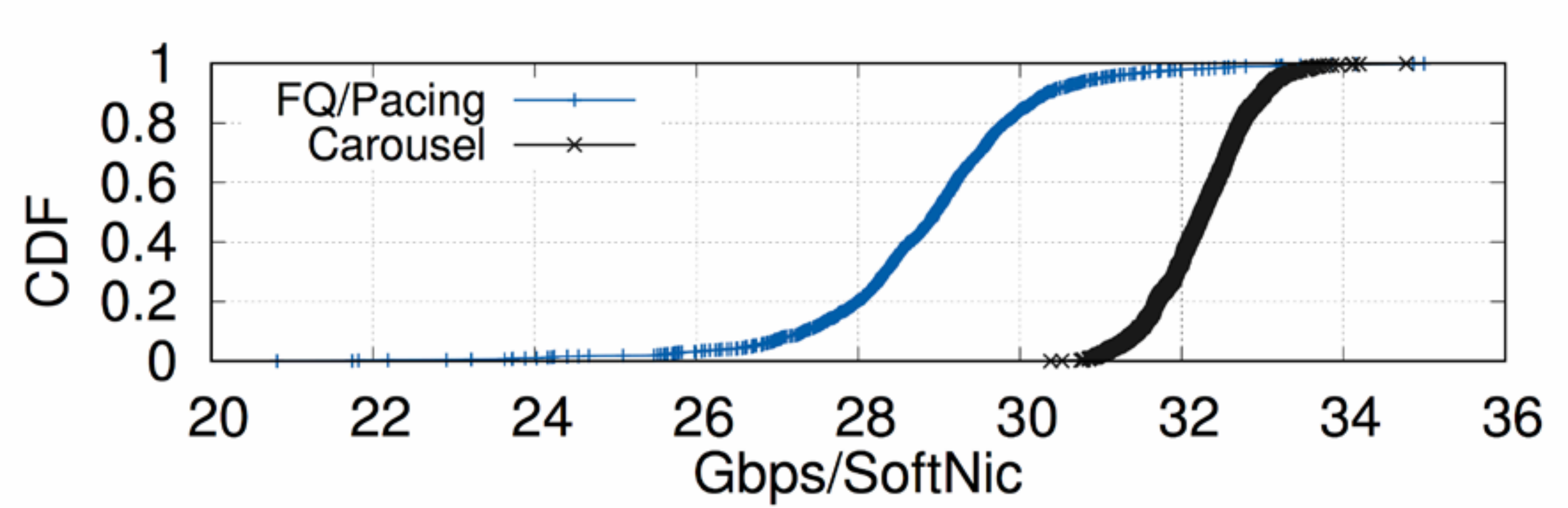


Carousel saves up to 8.2% of overall CPU utilization (5.9 cores on a 72 core machine)

SoftNIC Utilization

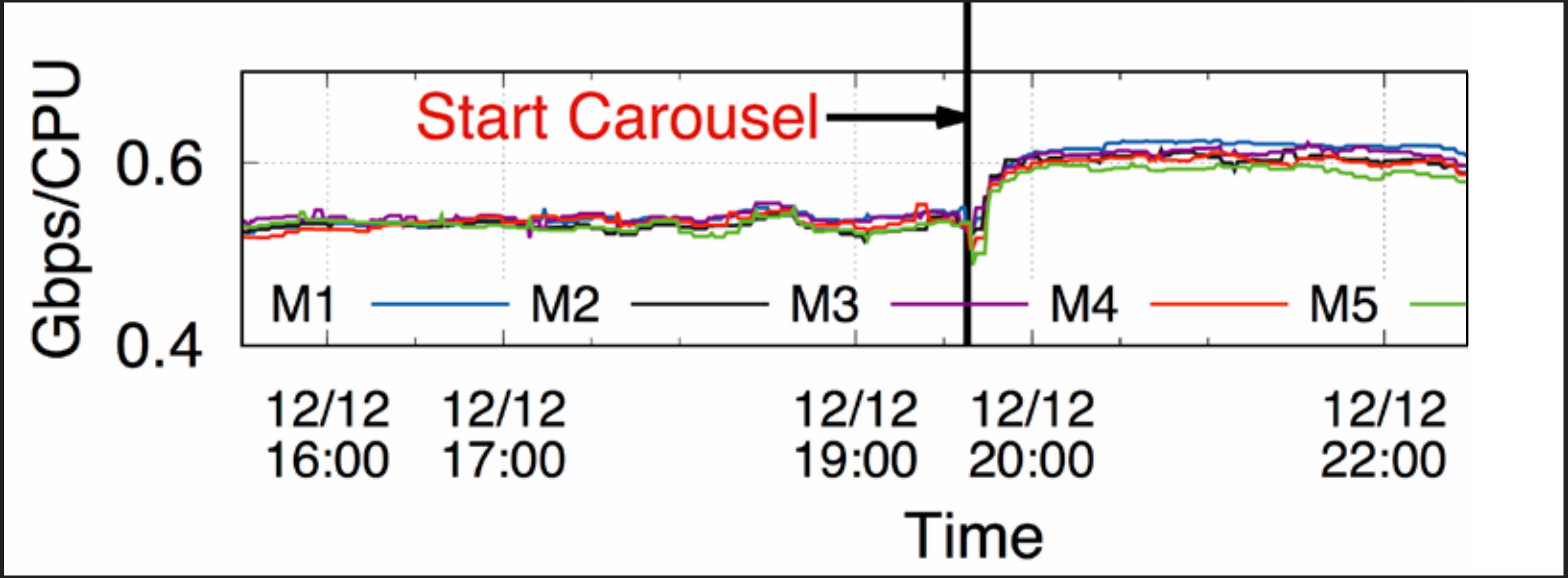


SoftNIC Utilization



Carousel improves even Software NIC utilization by 12% by increasing size of batches of packets enqueue in the Software NIC

Evaluation Summary



Performance improvement when Carousel starts on 5 different machines

Conclusion

- Carousel allows networks operators for the first time to shape tens of thousands of flows individually
- Carousel advantages make a strong case for providing single-queue shaping and backpressure in kernel, userspace stacks, hypervisors, and hardware

Questions?