REVERIE

Low Pass Filter-Based Switch Buffer Sharing for Datacenters with RDMA and TCP Traffic

Vamsi Addanki, Wei Bai, Stefan Schmid, Maria Apostolaki

Technische Universität Berlin   Microsoft   Technische Universität Berlin   PRINCETON UNIVERSITY

Traditional Datacenter Networking

- TCP-based applications
- Host-networking consumes CPU clock cycles (a lot!!)
- Loss-tolerant traffic

REVERIE

Modern Datacenter Networking

- ~~TCP-based applications~~
  - RDMA-based applications
- ~~Host networking consumes CPU clock cycles (a lot!!)~~
  - Host networking is offloaded to the NIC
  - NIC implements the entire networking stack
- ~~Loss-tolerant traffic~~
  - Lossless traffic
  - Requires Priority Flow Control (PFC)

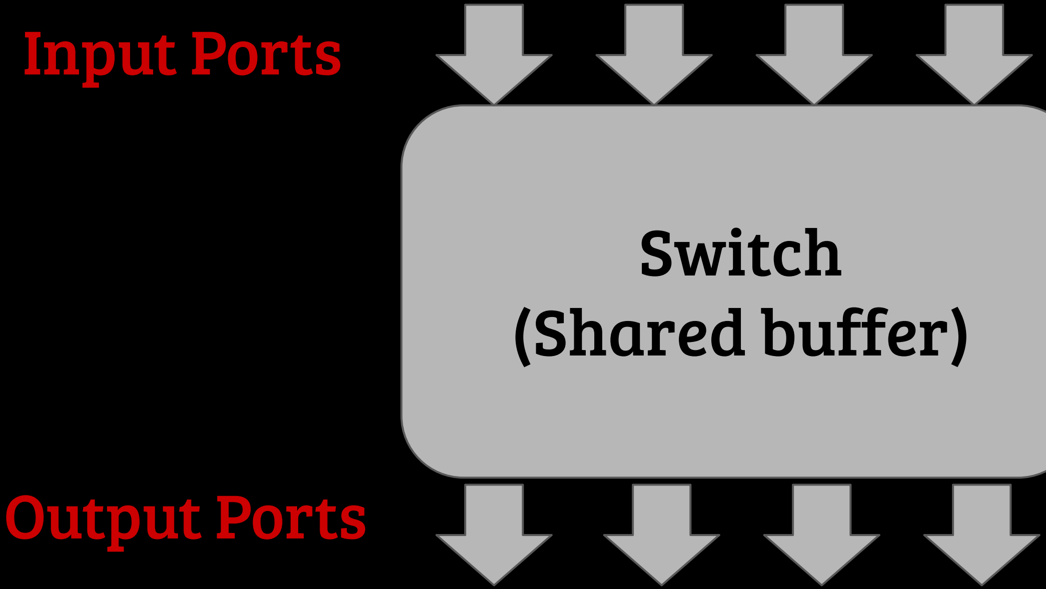
REVERIE

Production Datacenter Networks

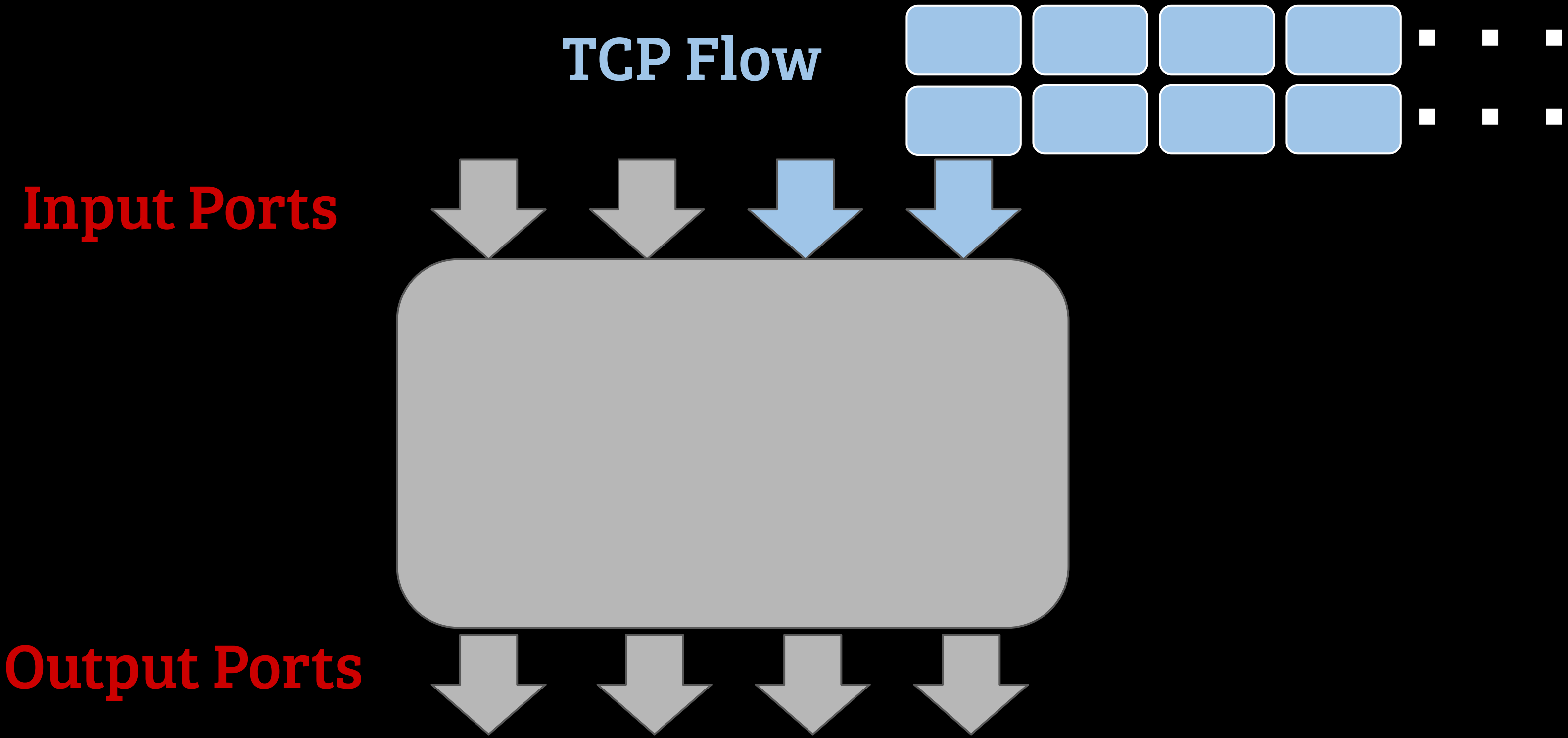
- A mix of RDMA and TCP traffic
- Switches use shared buffers
- Both RDMA and TCP *share* the limited buffer space at each switch in the network

REVERIE

Switch Buffer Sharing with TCP

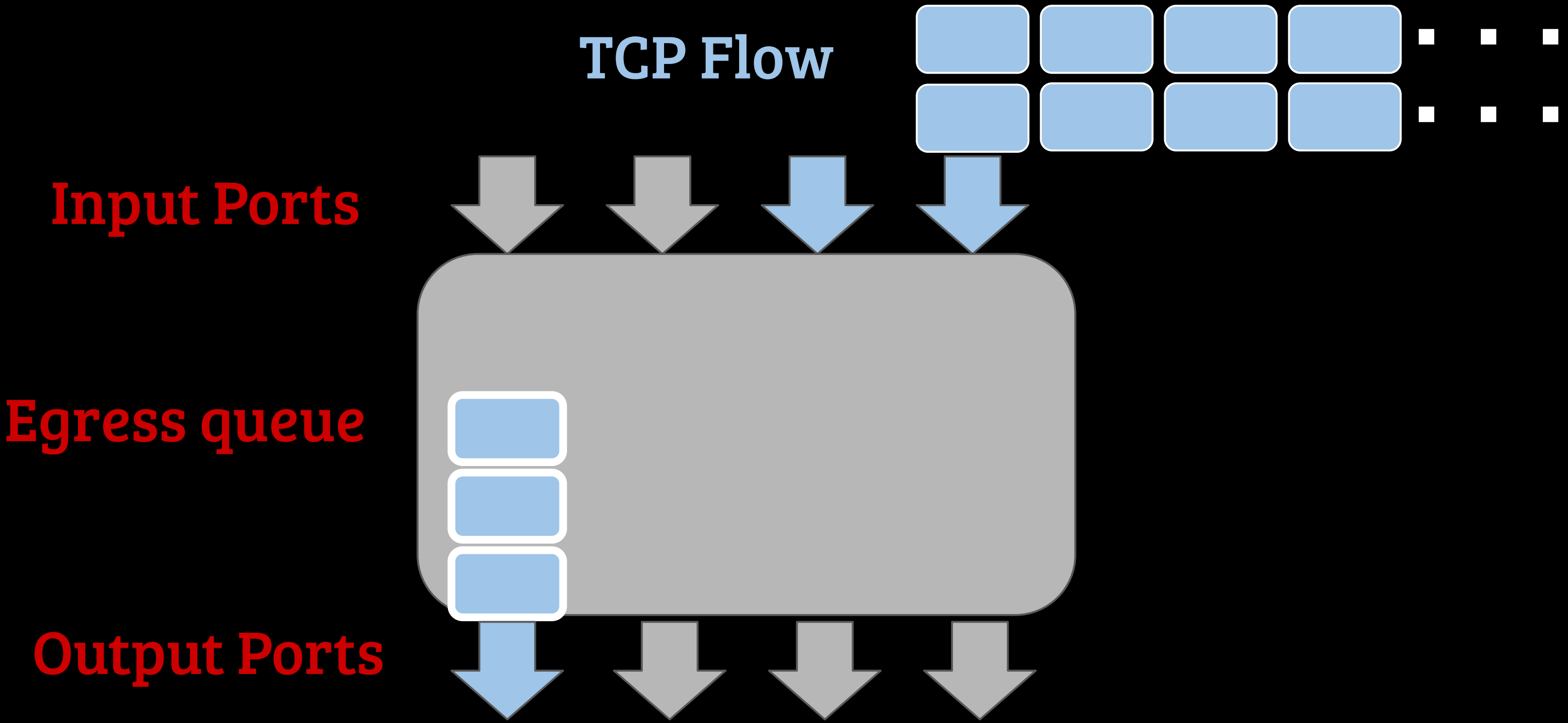


Switch Buffer Sharing with TCP



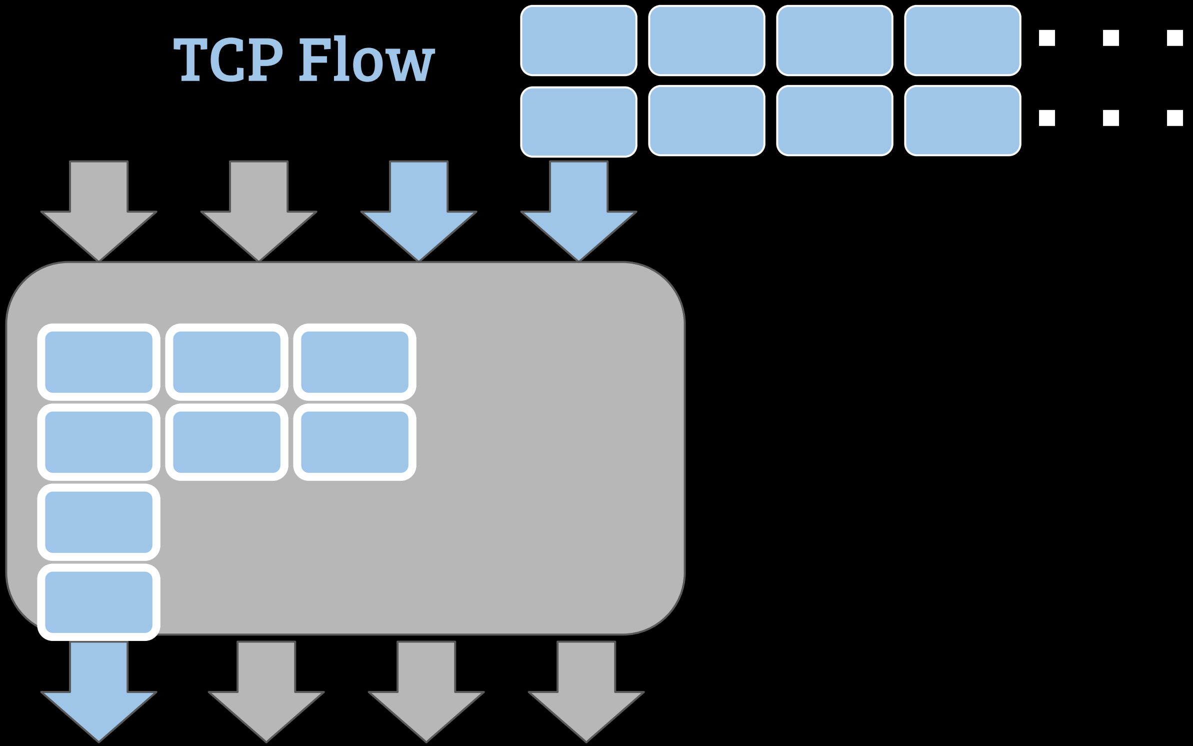
REVERIE

Switch Buffer Sharing with TCP



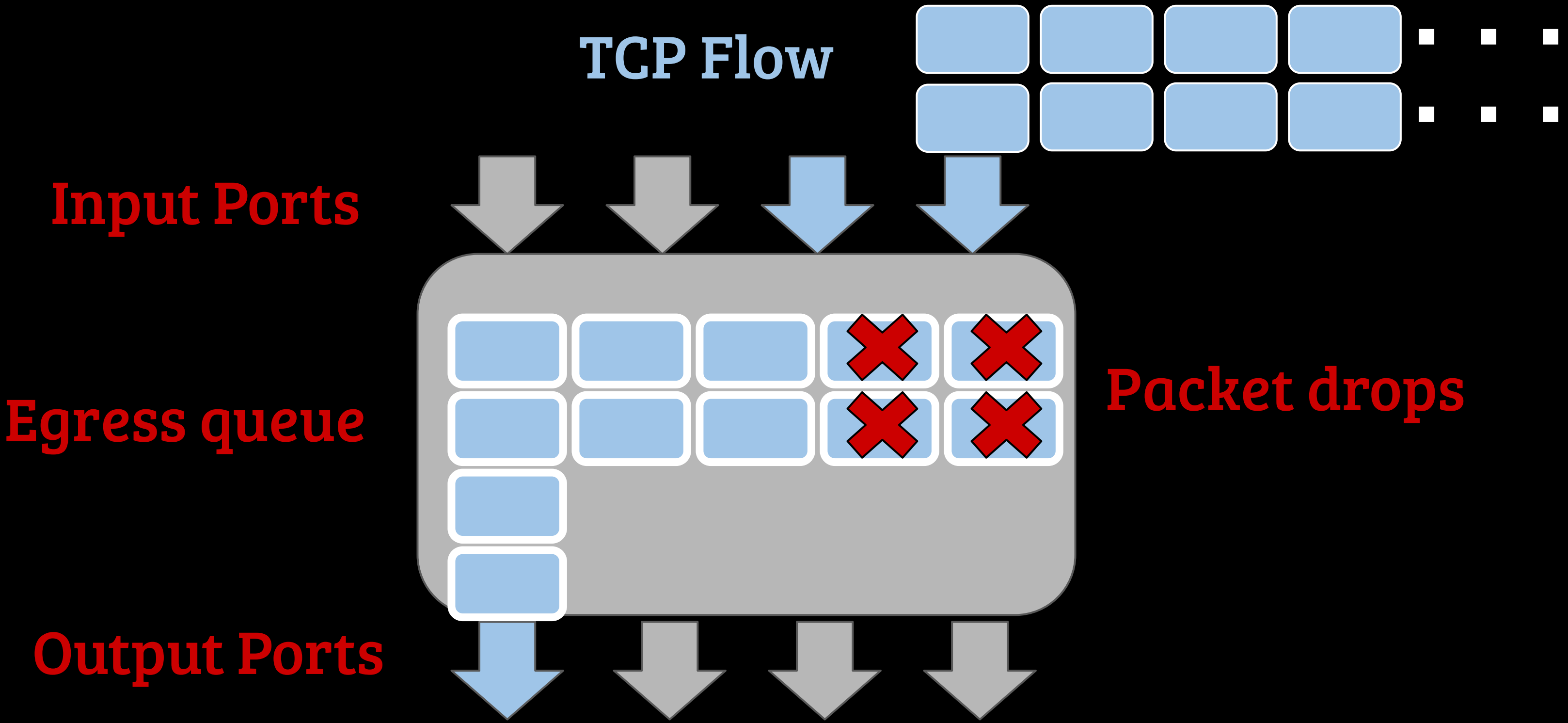
REVERIE

Switch Buffer Sharing with TCP



REVERIE

Switch Buffer Sharing with TCP

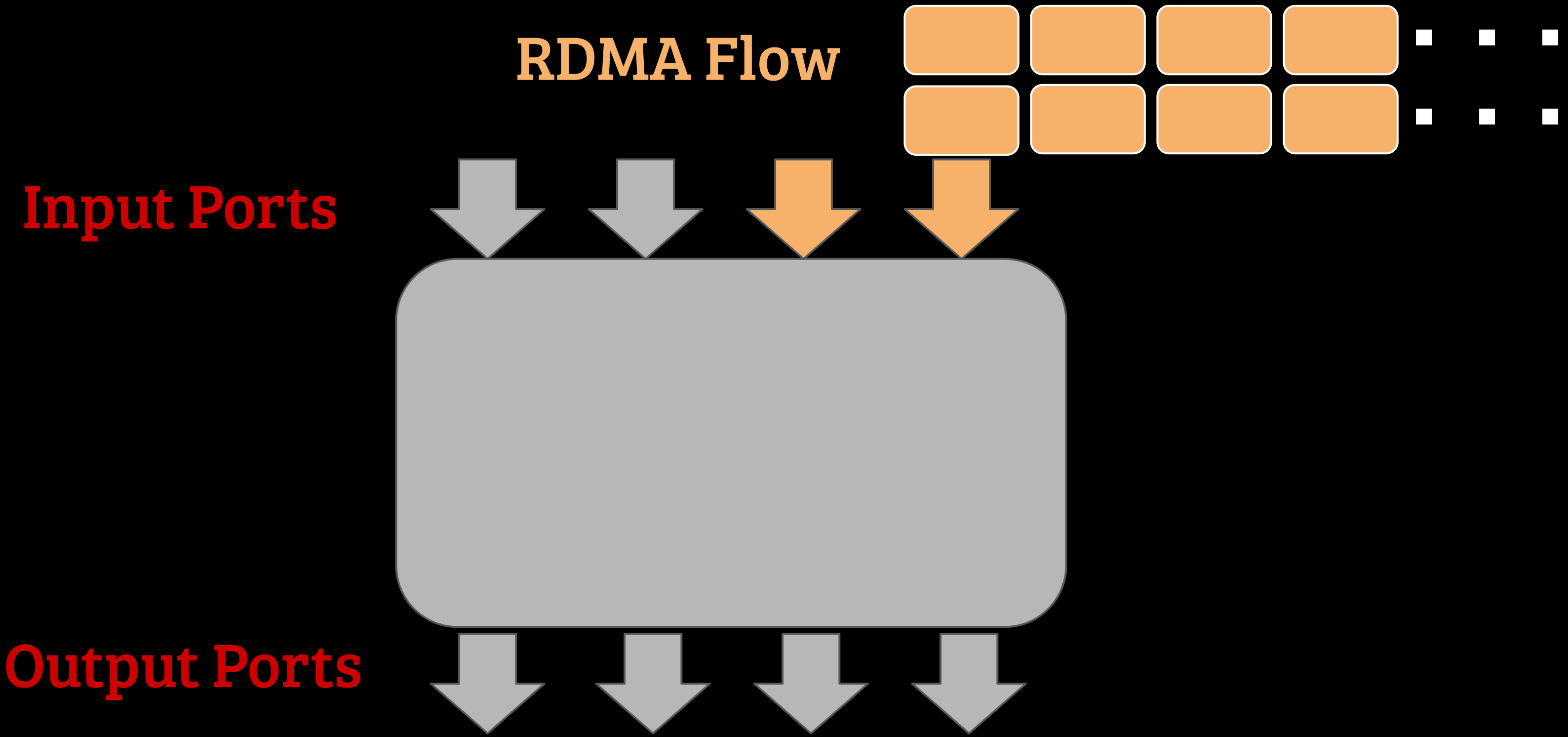


Switch Buffer Sharing with TCP

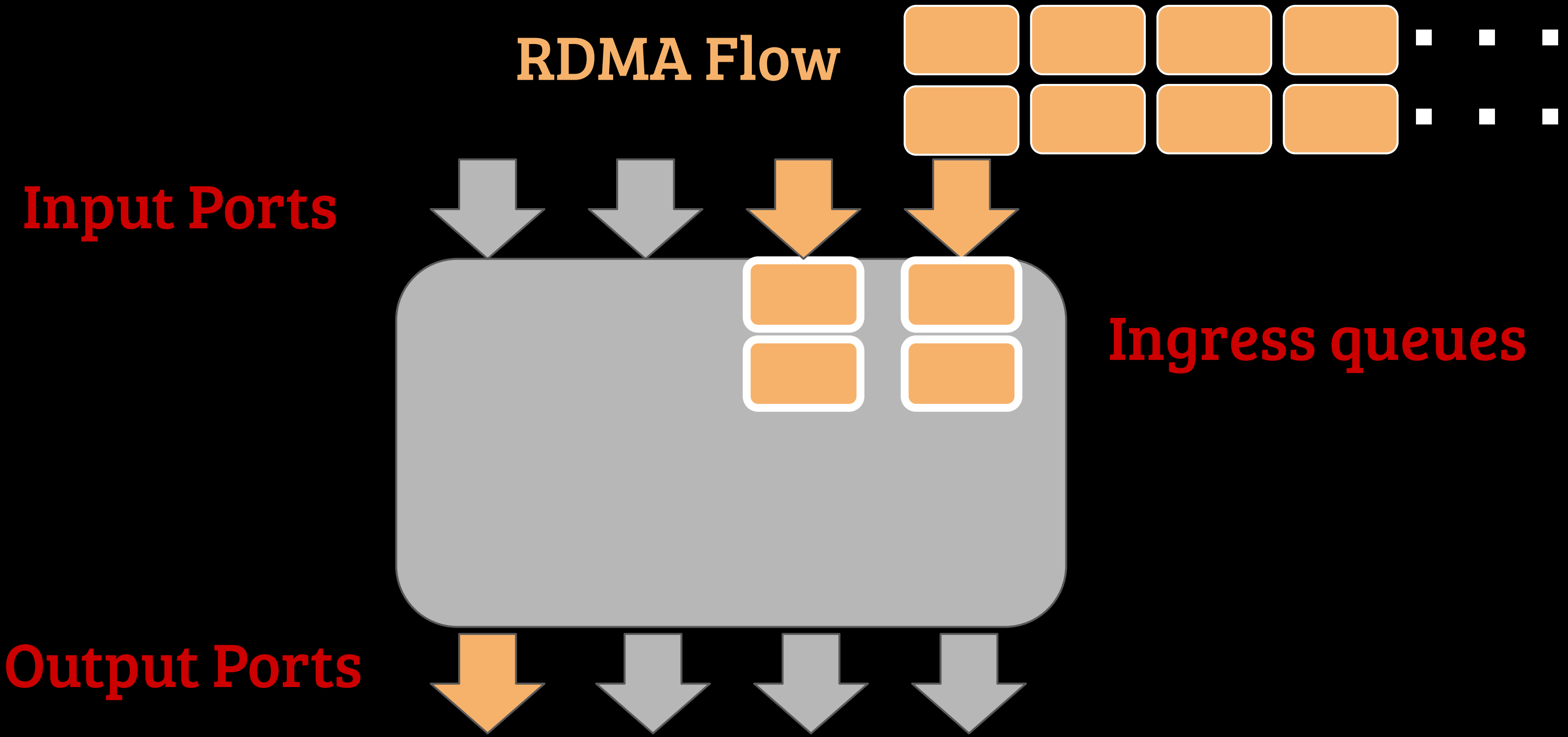
- Based on egress queue lengths and packet drops
- A buffer sharing algorithm assigns a threshold for each egress queue in a switch
- Packet accepted: Threshold > Queue length (egress)
- Packet dropped: Threshold < Queue length (egress)

REVERIE

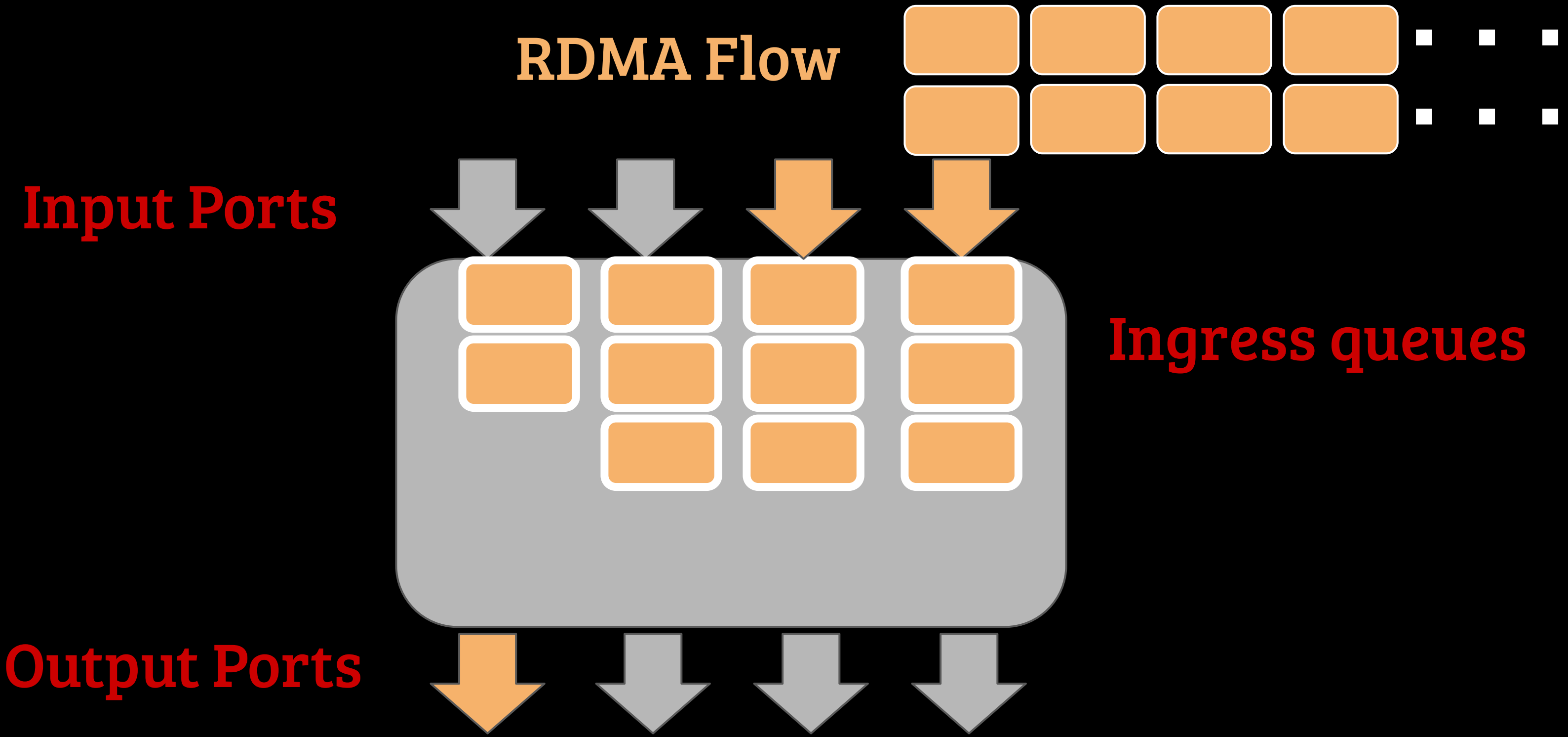
Switch Buffer Sharing with RDMA



Switch Buffer Sharing with RDMA

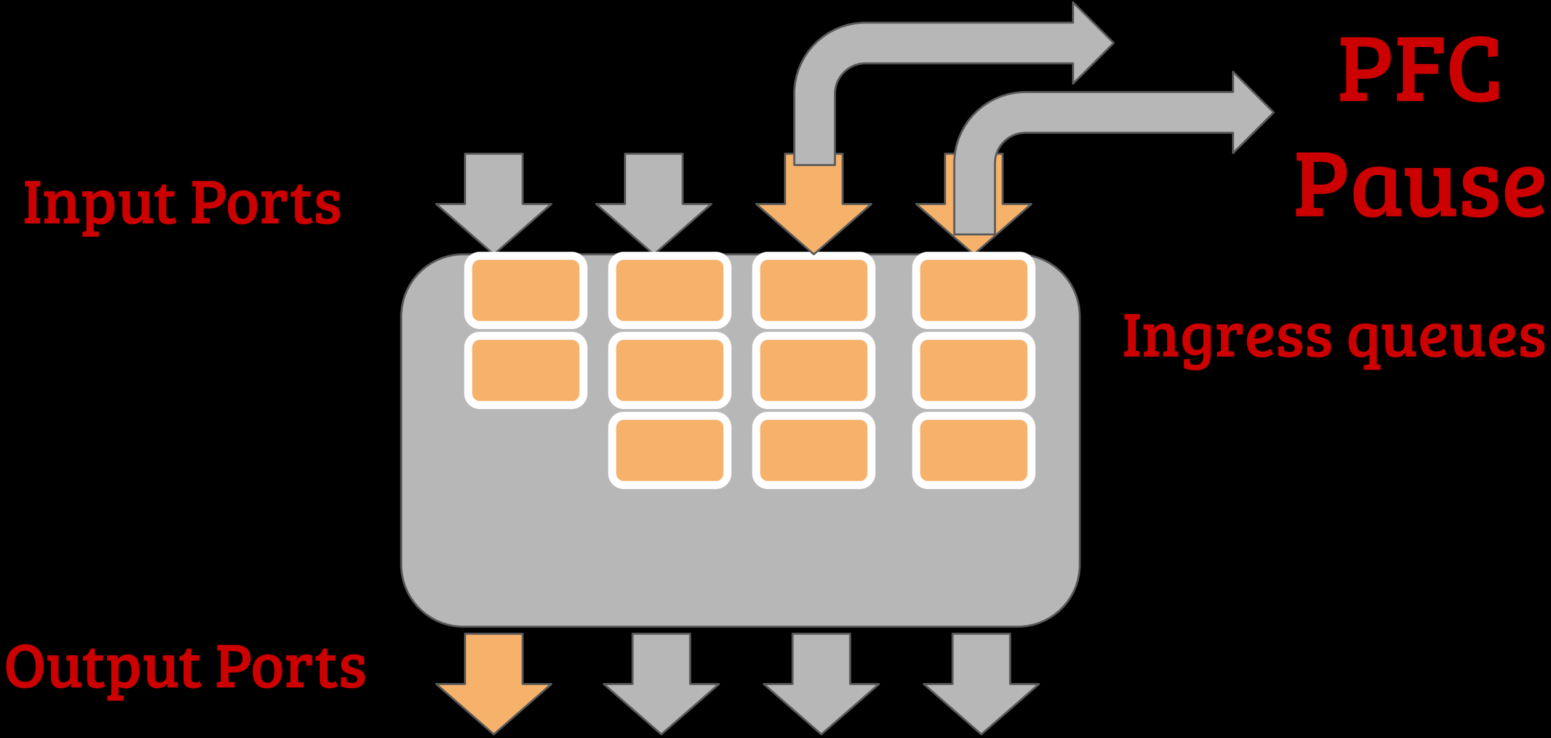


Switch Buffer Sharing with RDMA



REVERIE

Switch Buffer Sharing with RDMA



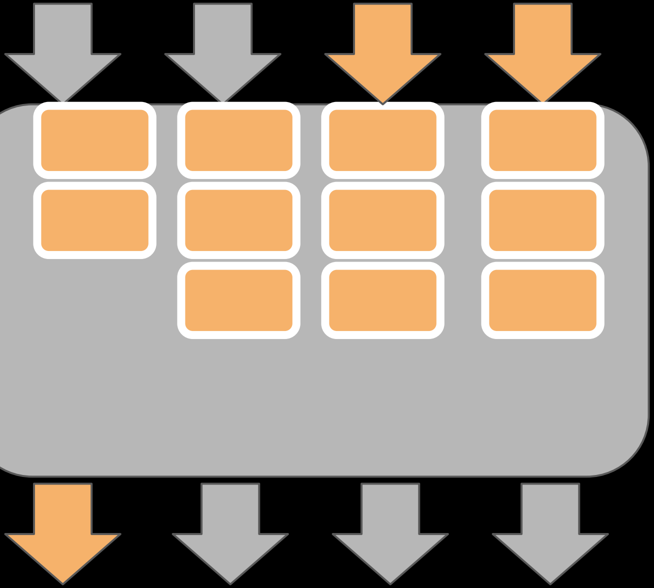
REVERIE

Switch Buffer Sharing with RDMA

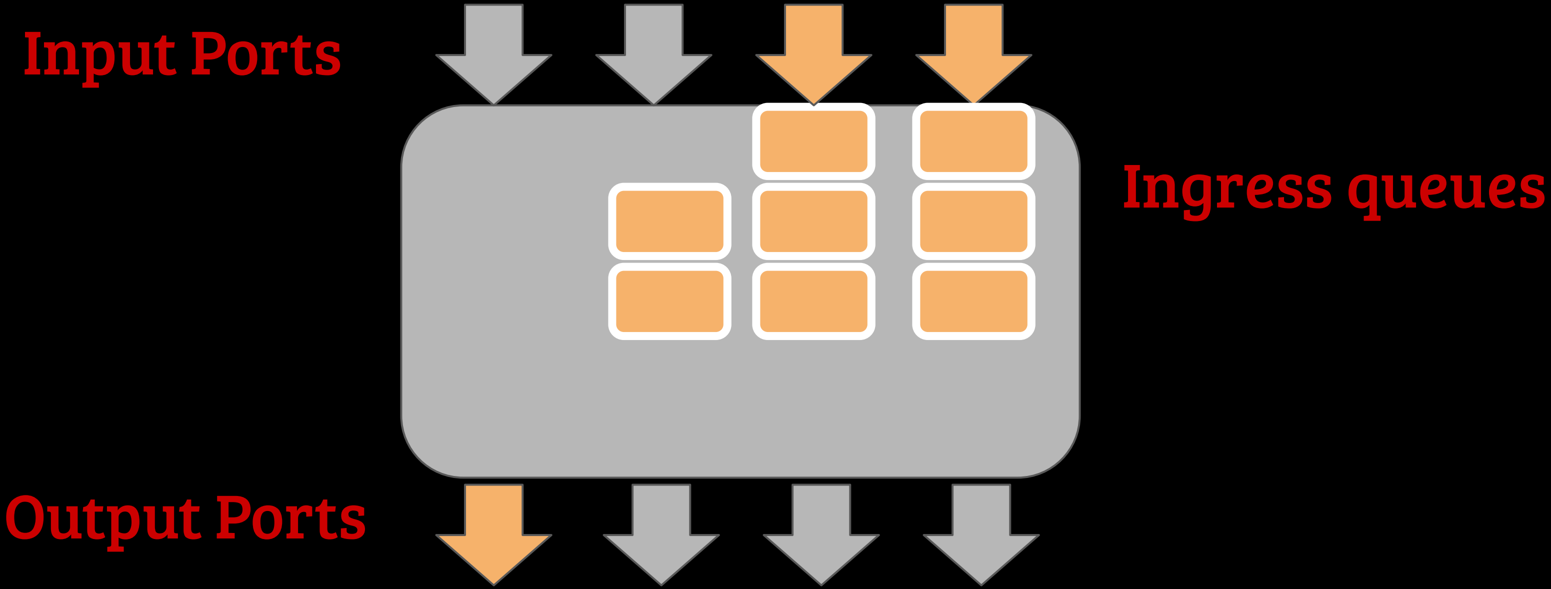
Input Ports

Ingress queues

Output Ports



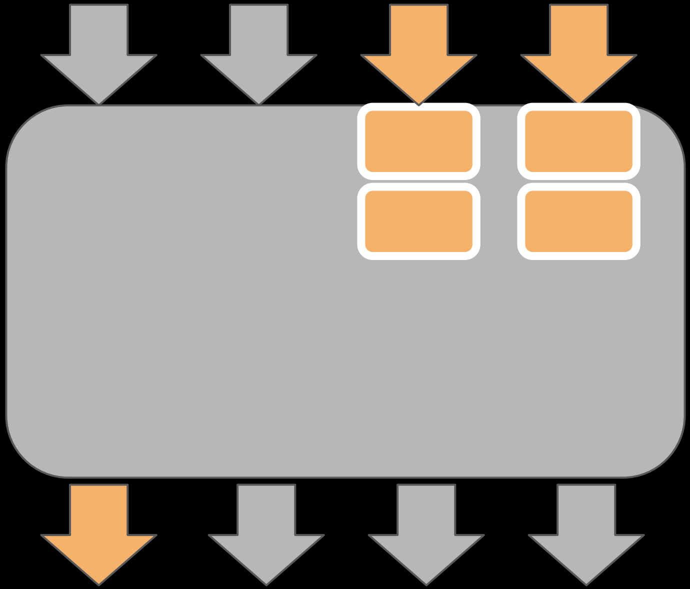
Switch Buffer Sharing with RDMA



REVERIE

Switch Buffer Sharing with RDMA

Input Ports

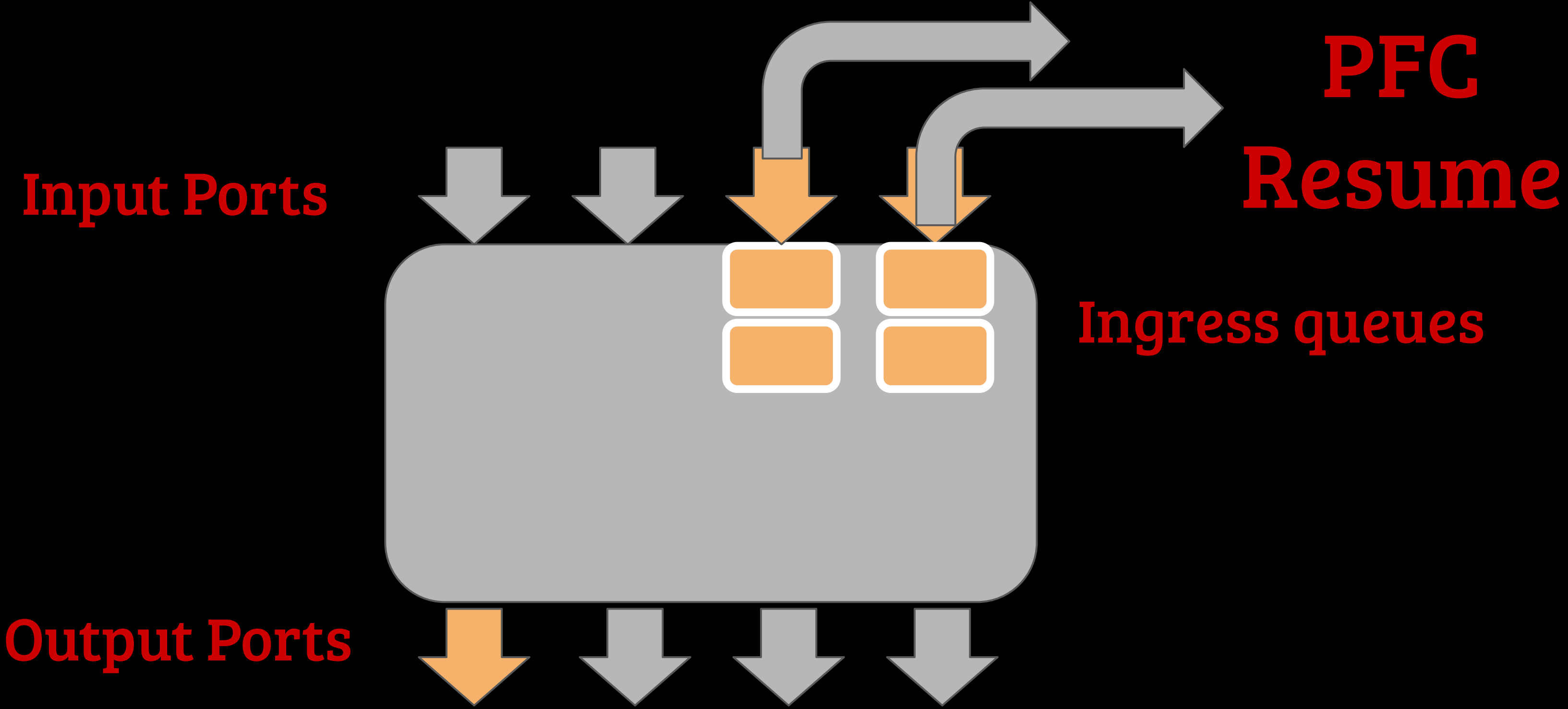


Ingress queues

Output Ports

REVERIE

Switch Buffer Sharing with RDMA



Switch Buffer Sharing with RDMA

- ~~Based on egress queue lengths and packet drops~~
- Based on ingress queue lengths and PFC

REVERIE

Switch Buffer Sharing with RDMA

- Based on ingress queue lengths and PFC
- A buffer sharing algorithm assigns a threshold for each ~~egress~~ ingress queue in a switch

Switch Buffer Sharing with RDMA

- Based on ingress queue lengths and PFC
- A buffer sharing algorithm assigns a threshold for each ingress queue in a switch
- ~~Packet accepted: Threshold > Queue length (egress)~~
- Packets are always accepted

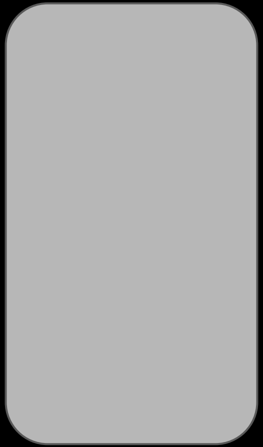
Switch Buffer Sharing with RDMA

- Based on ingress queue lengths and PFC
- A buffer sharing algorithm assigns a threshold for each ingress queue in a switch
- Packets are always accepted
- ~~Packet dropped: Threshold < Queue length (egress)~~
- PFC Pause: Threshold < Queue length (ingress)

Problem: Switch Buffer Sharing with RDMA + TCP

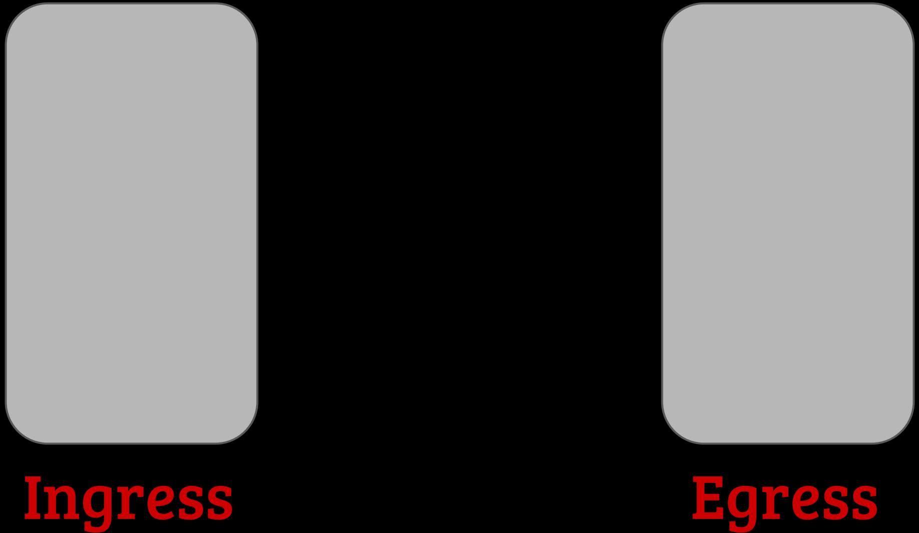
- Harmful interactions between RDMA and TCP
- Unfair buffer allocation
- Poor burst absorption

Background: SONiC Buffer Model



Background: SONiC Buffer Model

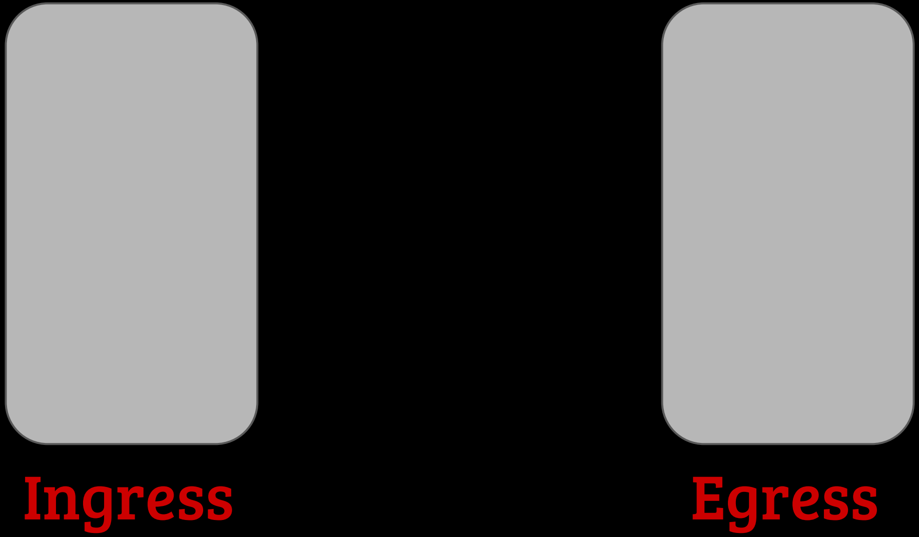
- Two (logical) views of the buffer



REVERIE

Background: SONiC Buffer Model

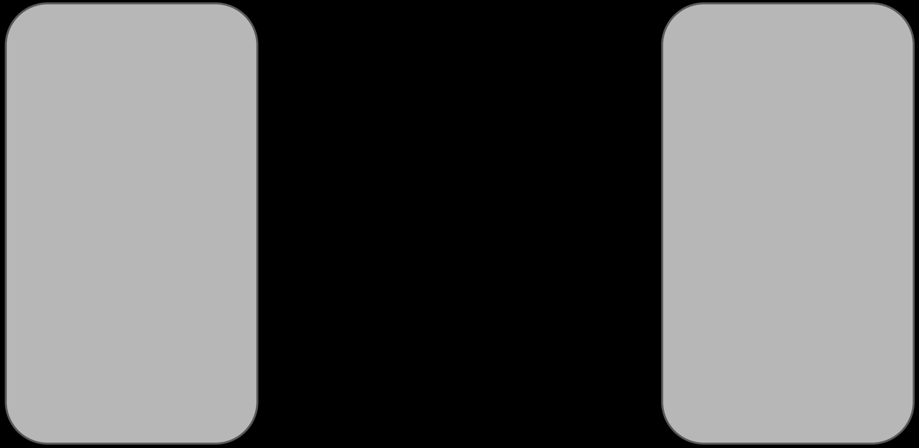
- Every packet is accounted both in ingress and egress



REVERIE

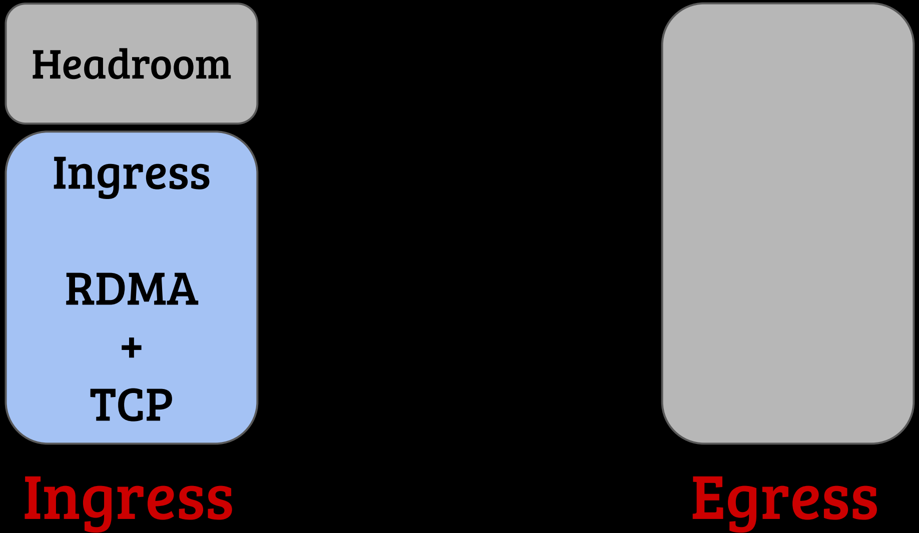
Background: SONiC Buffer Model

- Buffer is logically divided into pools



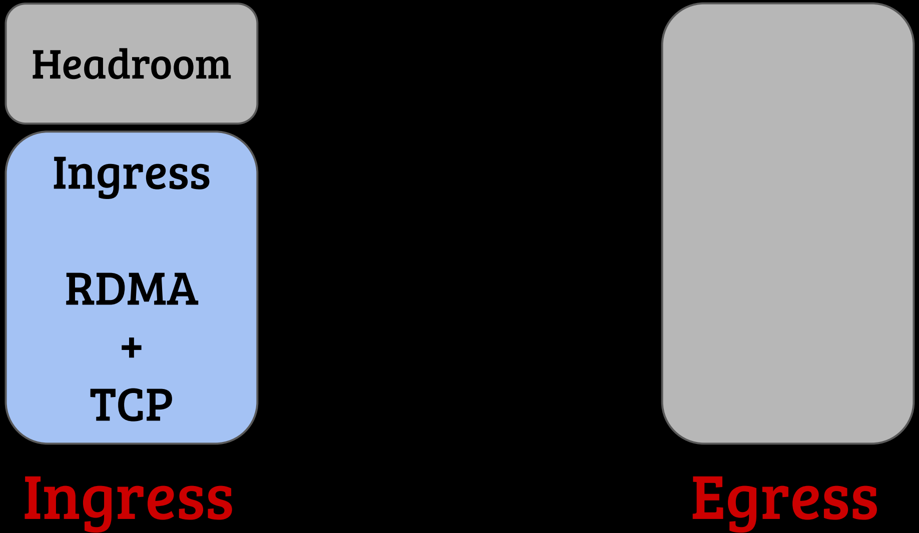
Background: SONiC Buffer Model

- Ingress pool is shared by both RDMA and TCP



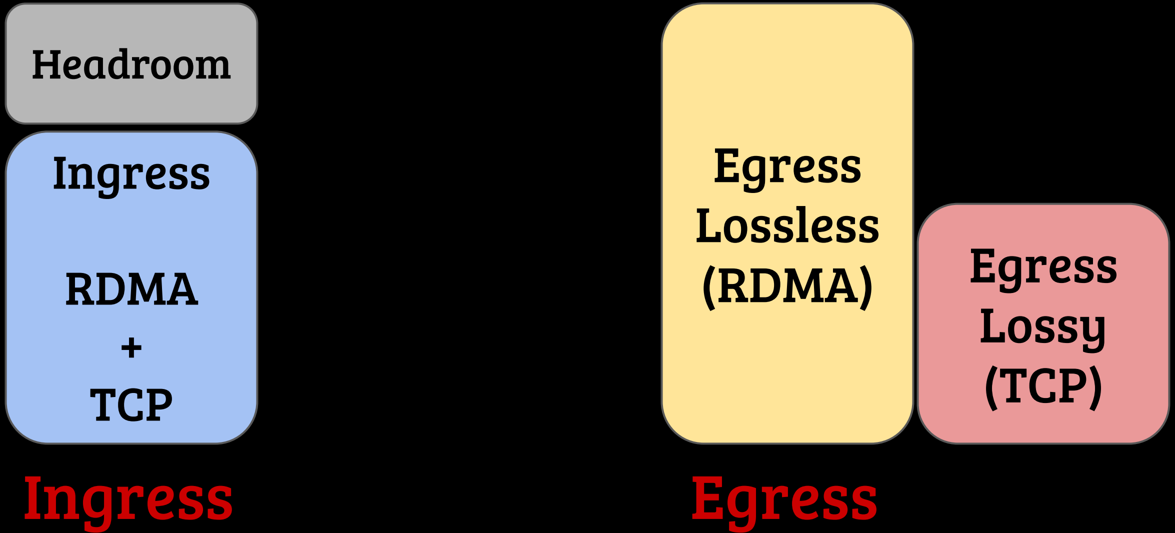
Background: SONiC Buffer Model

- Headroom pool in the ingress is reserved for RDMA



Background: SONiC Buffer Model

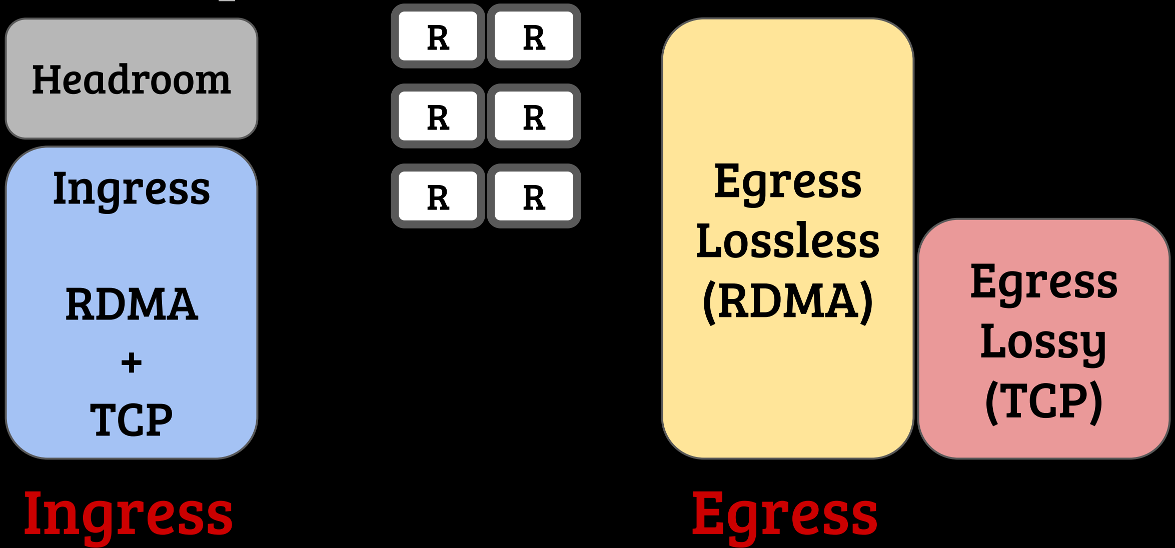
- Egress lossless (RDMA) and Egress lossy (TCP) pools



REVERIE

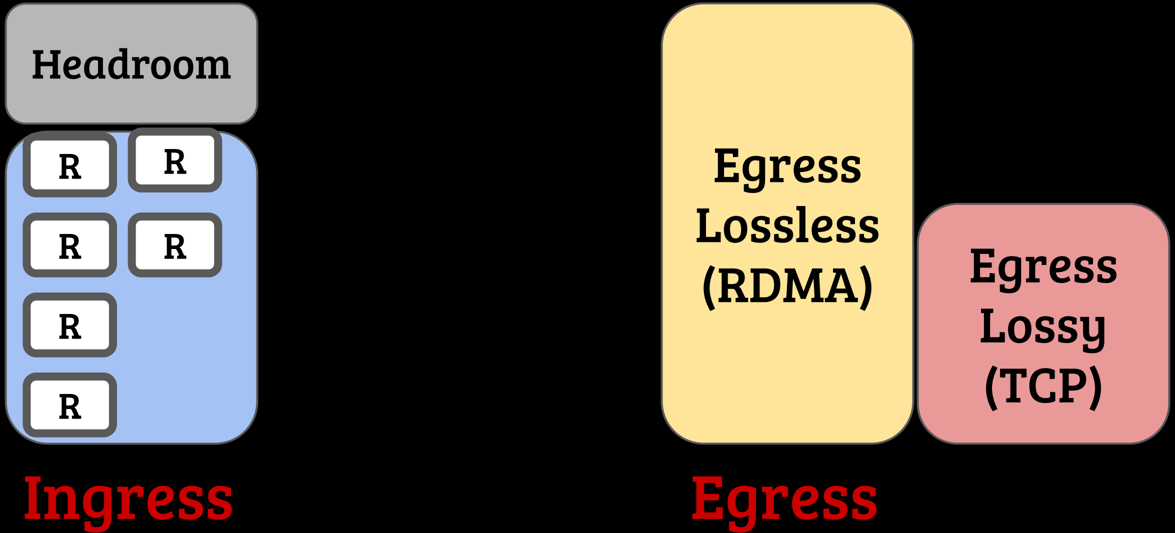
Background: SONiC Buffer Model

- Example: RDMA packets



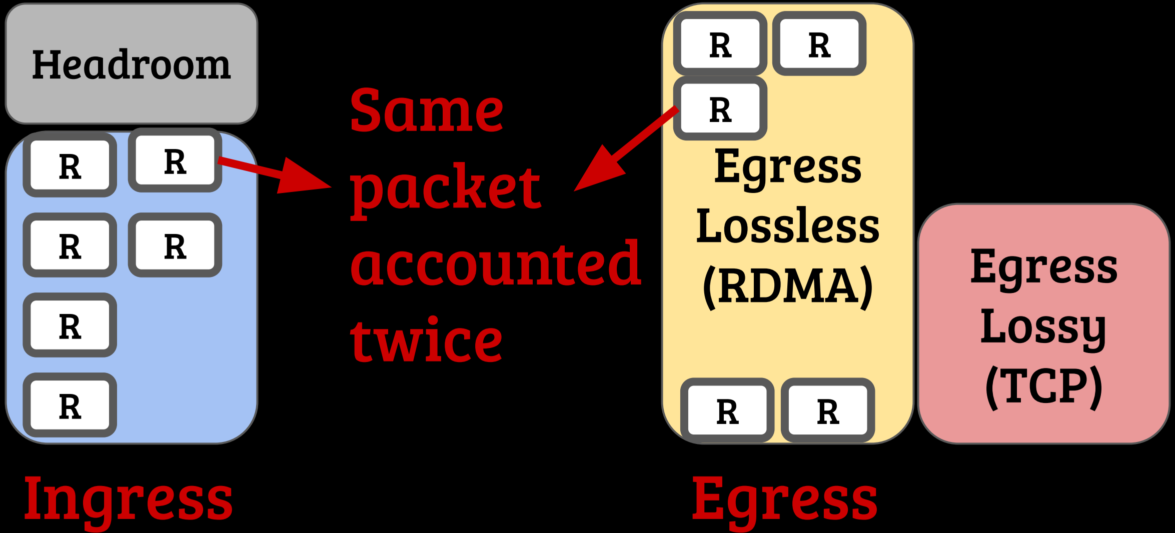
Background: SONiC Buffer Model

- Example: RDMA packets



Background: SONiC Buffer Model

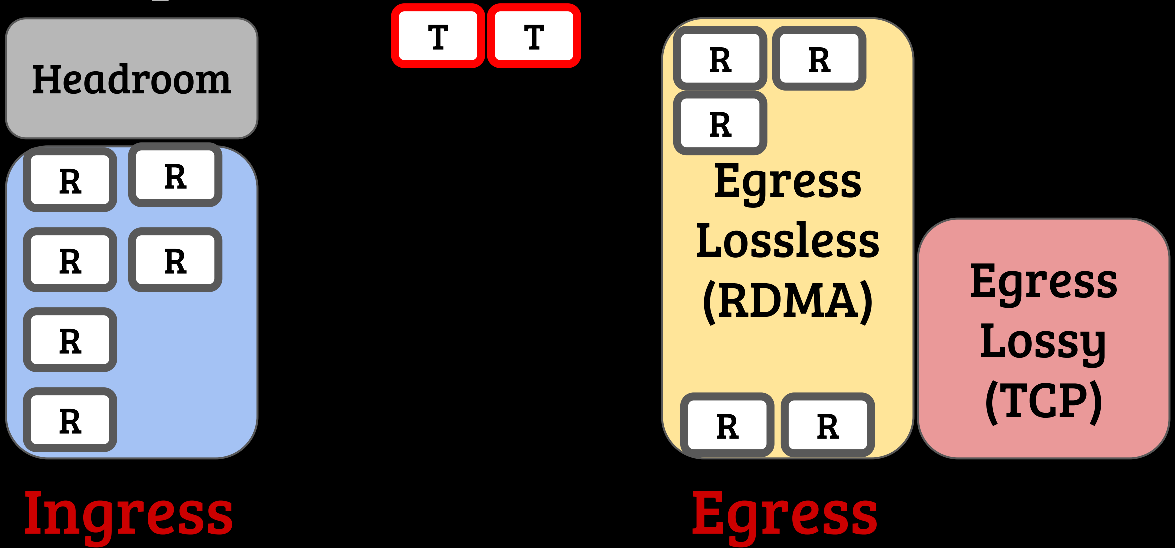
- Example: RDMA packets



REVERIE

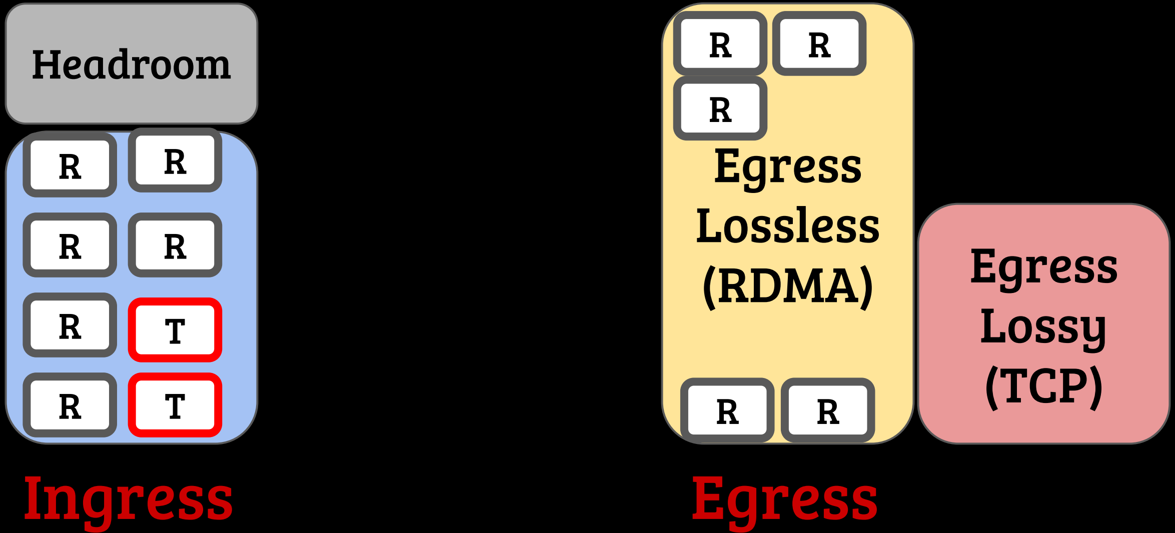
Background: SONiC Buffer Model

- Example: TCP packets



Background: SONiC Buffer Model

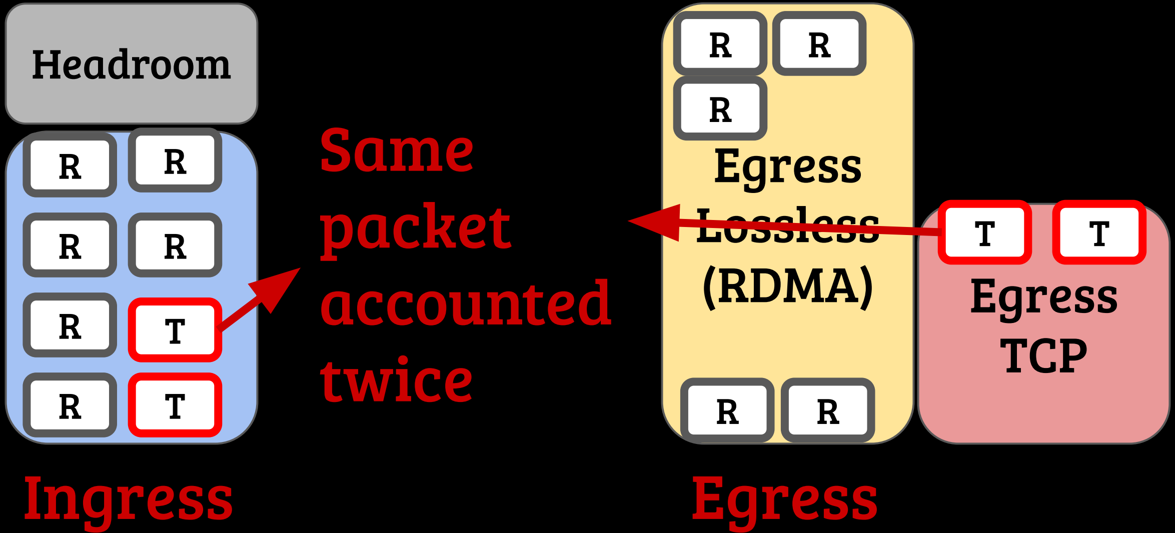
- Example: TCP packets



REVERIE

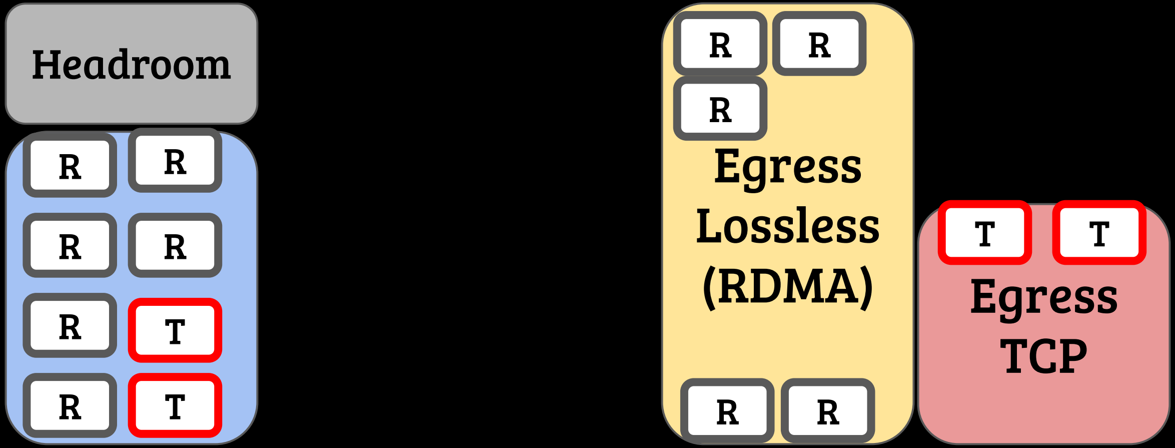
Background: SONiC Buffer Model

- Example: TCP packets



Background: SONiC Buffer Model

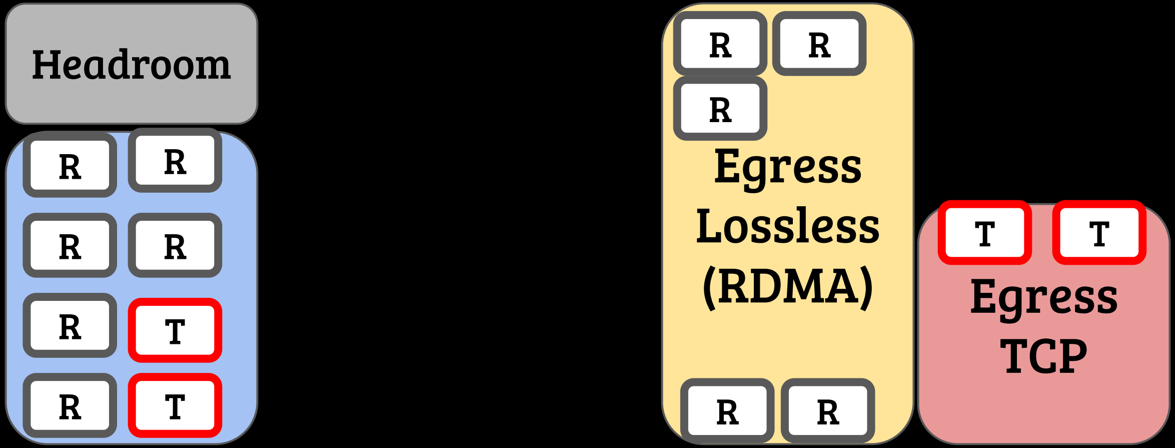
- Admission Control: Dynamic Thresholds in each pool



Ingress Egress

Background: SONiC Buffer Model

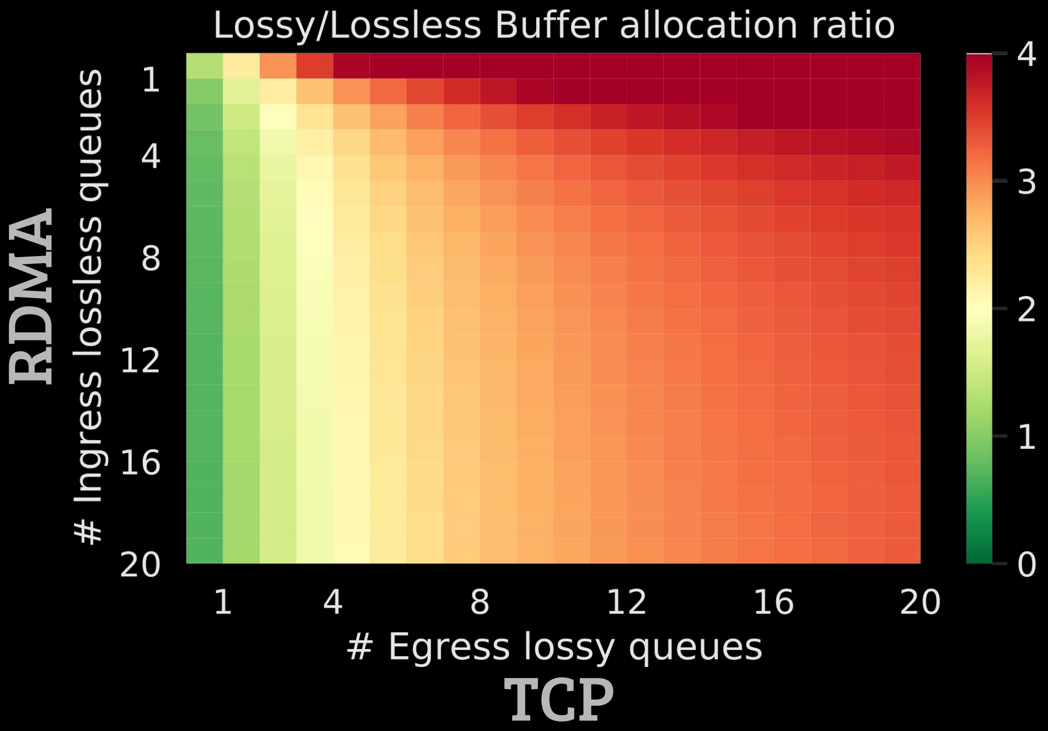
- Admission Control: $\alpha \times$ Remaining pool size



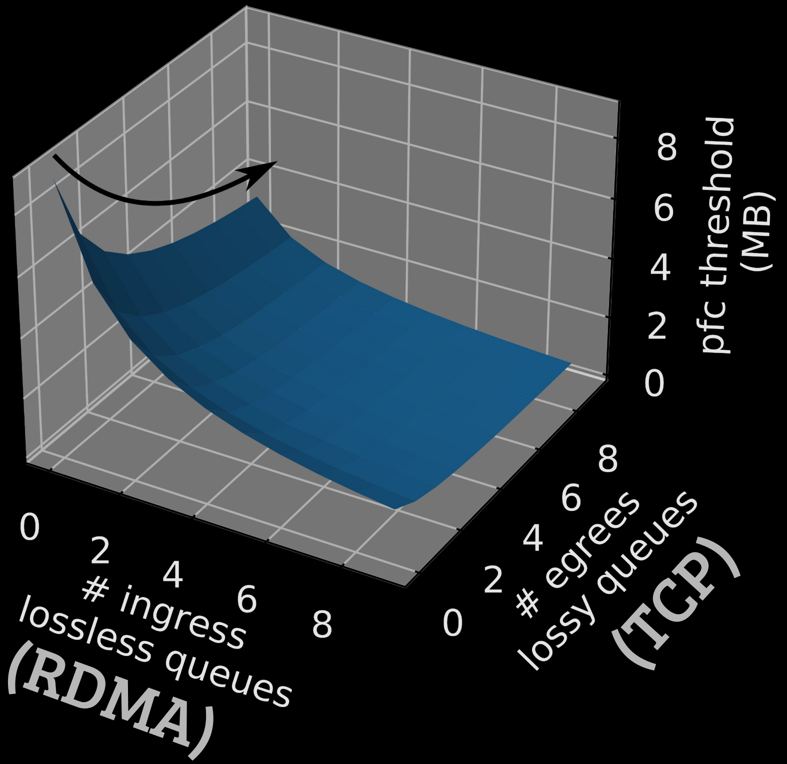
Ingress

Egress

Problem 1: TCP Gets More Buffer than RDMA under Contention

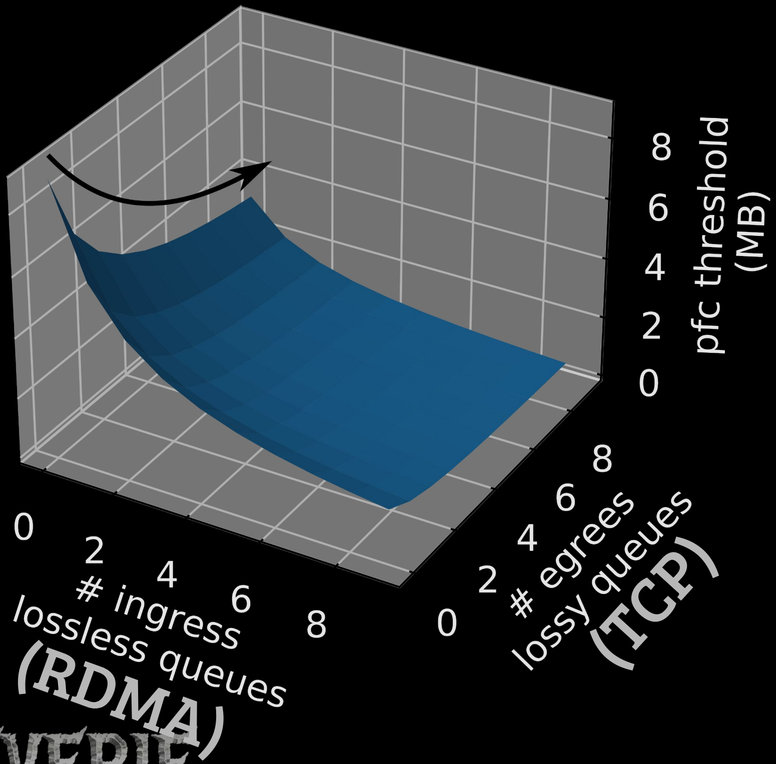


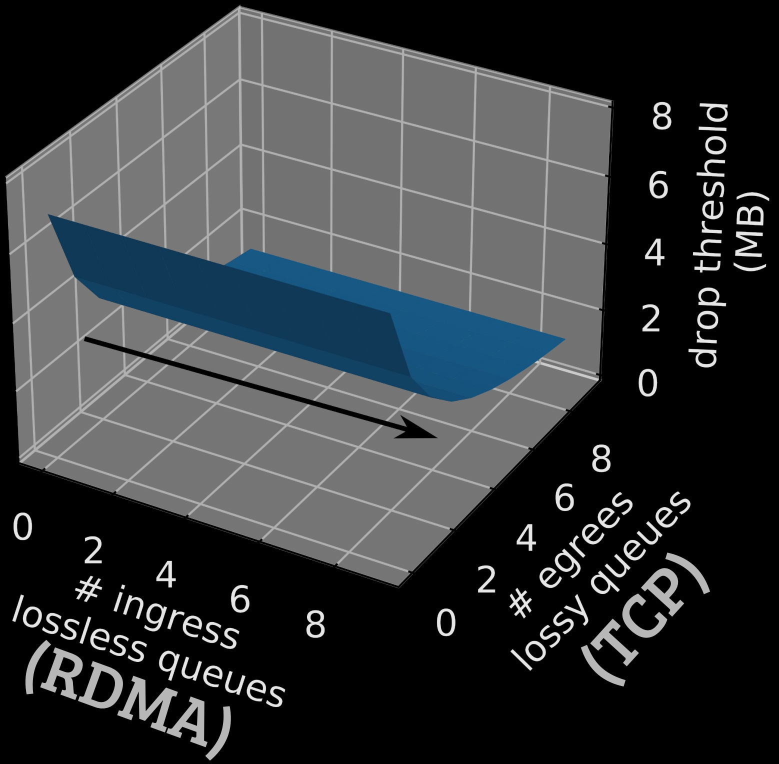
Problem 2: RDMA PFC Pause Due to TCP's Buffer Occupancy



Reverie

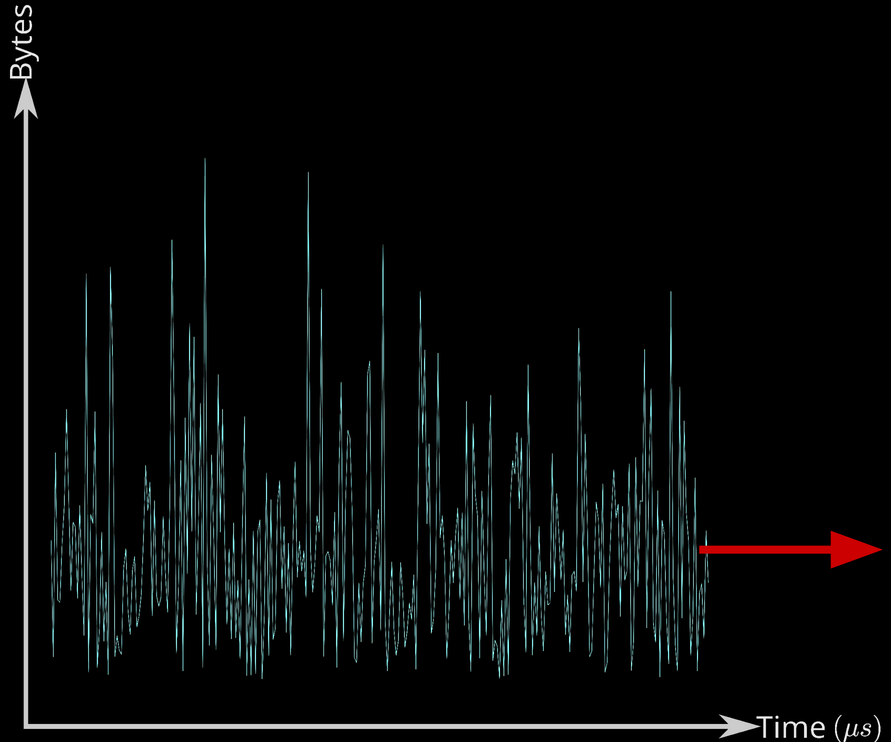
Problem 2: PFC Pause Due to TCP’s Buffer Occupancy





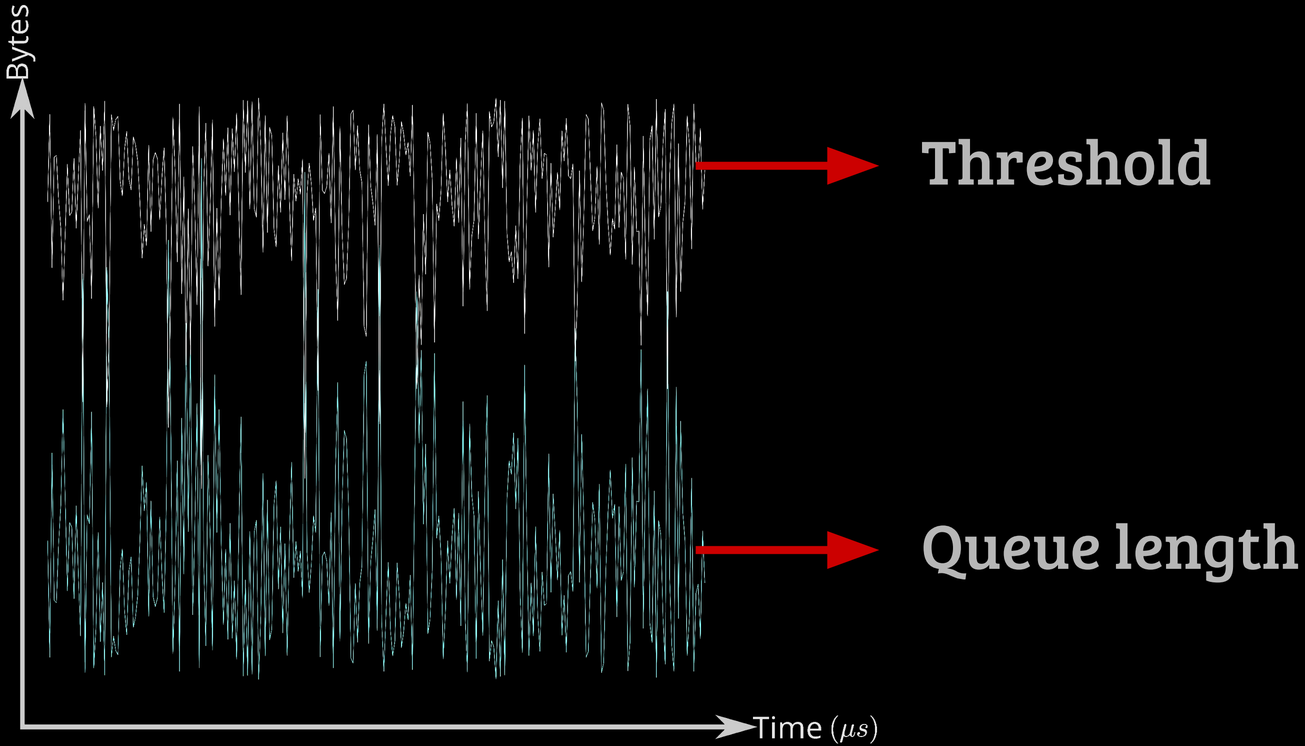
REVERIE

Problem 3: Poor Burst Absorption



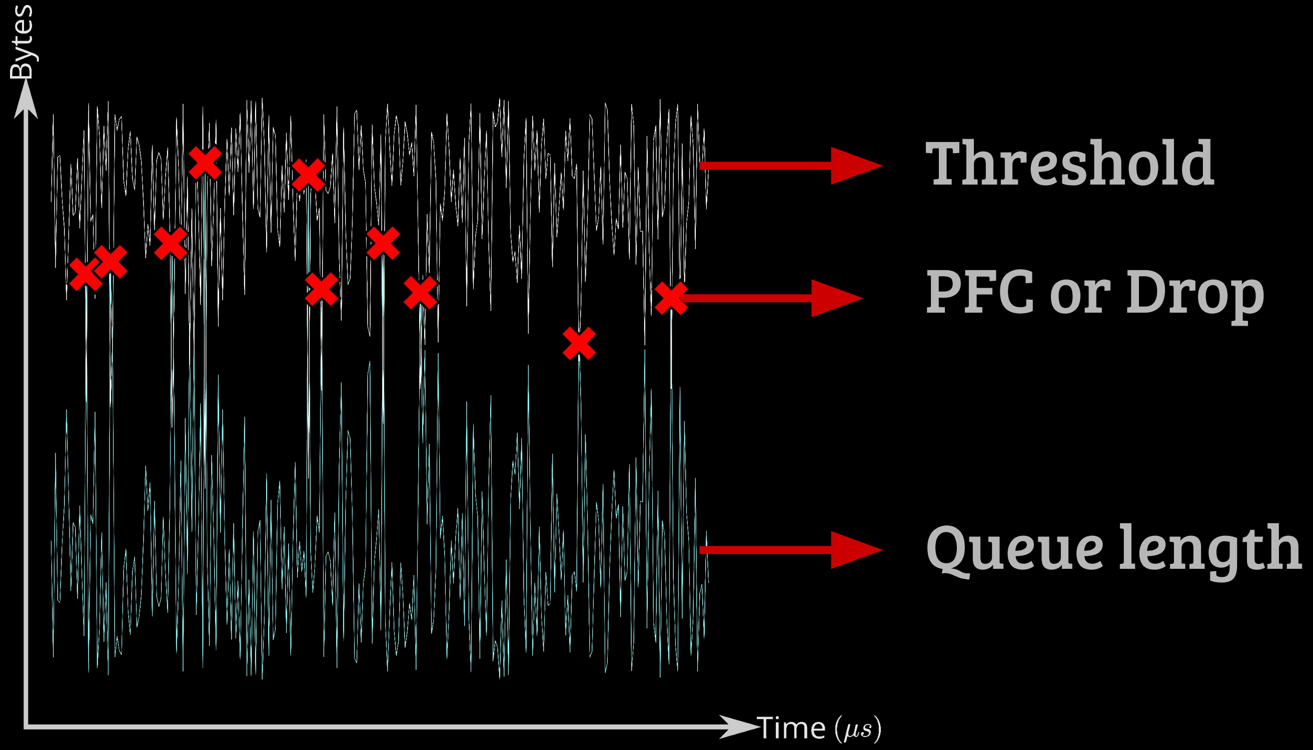
Queue length

Problem 3: Poor Burst Absorption

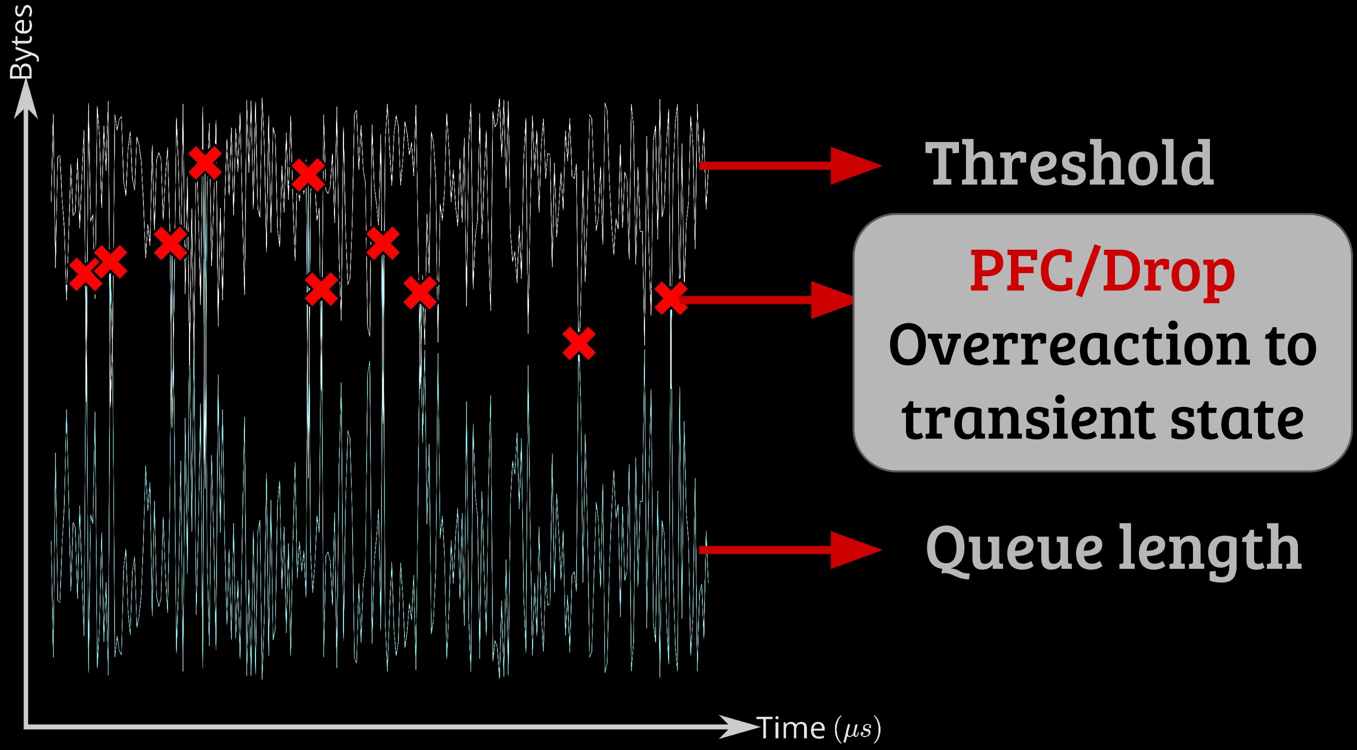


REVERIE

Problem 3: Poor Burst Absorption



Problem 3: Poor Burst Absorption



REVERIE

Can we isolate RDMA and TCP while improving burst absorption?

REVERIE

Reverie

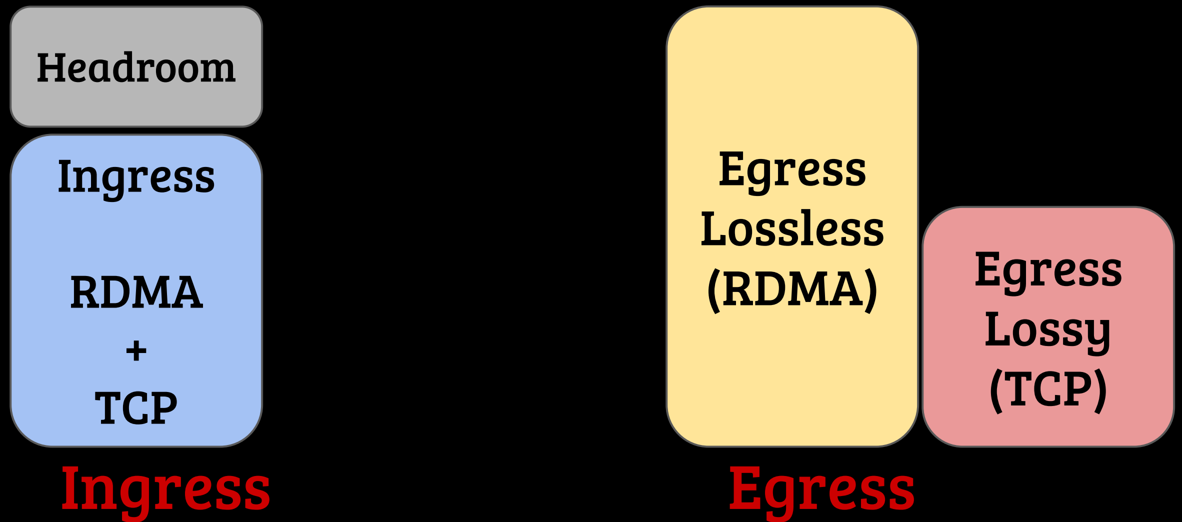
- Achieves isolation across RDMA and TCP
- Improves burst absorption

Reverie

- Single shared buffer pool for RDMA and TCP

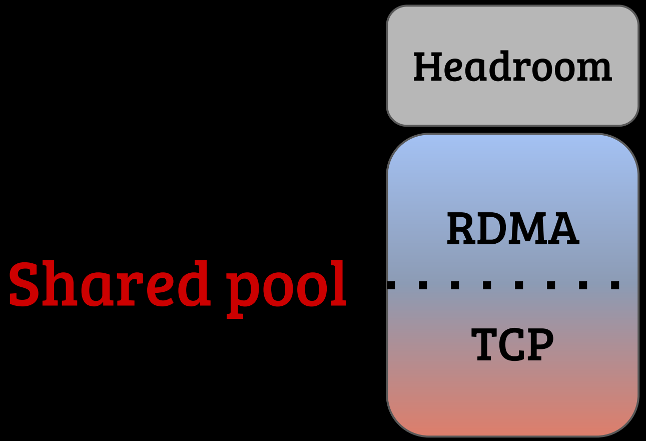
Reverie

- Single shared buffer pool for RDMA and TCP



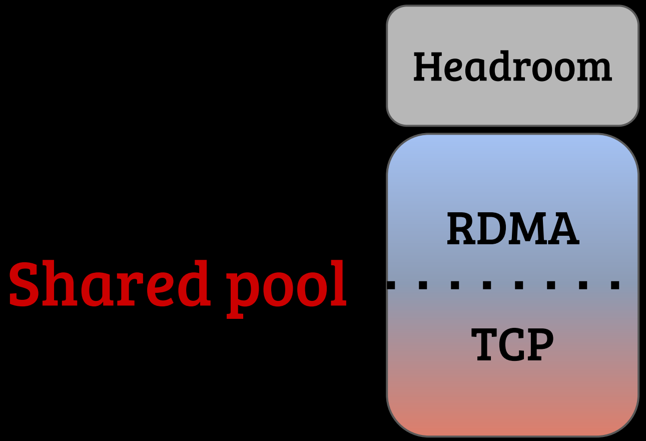
Reverie

- Single shared buffer pool for RDMA and TCP



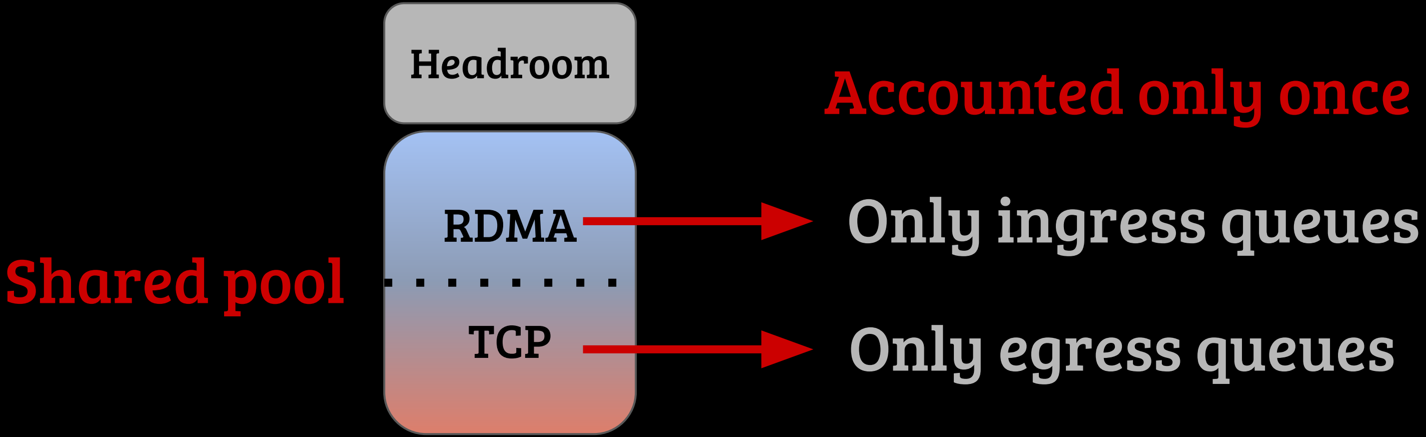
Reverie

- Single shared buffer pool for RDMA and TCP
- Consolidated ingress and egress buffer views
  - Birds-eye view of the buffer



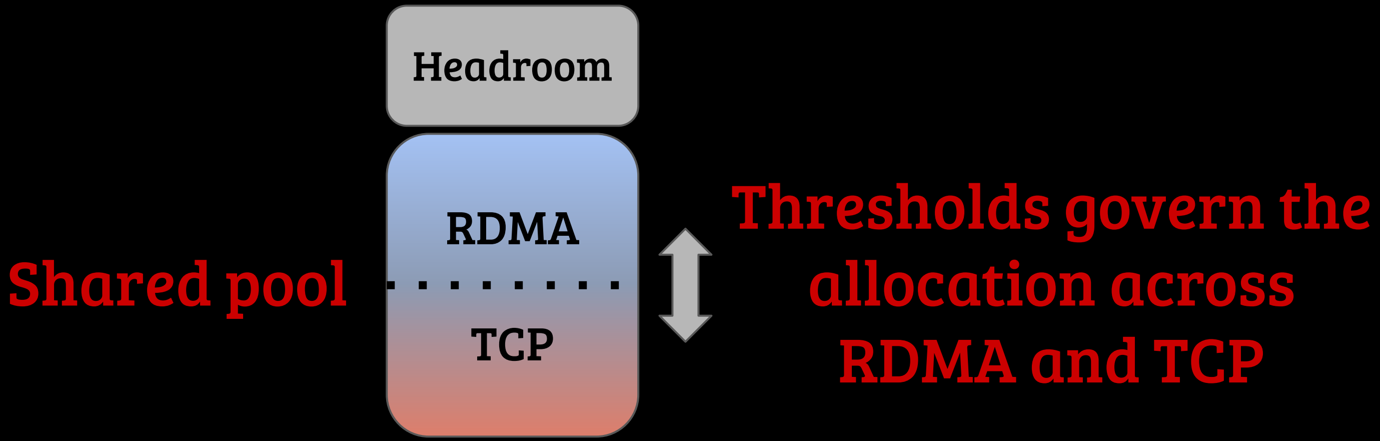
Reverie

- Single shared buffer pool for RDMA and TCP
- Consolidated ingress and egress buffer views
  - Birds-eye view of the buffer



Reverie

- Single shared buffer pool for RDMA and TCP
- Consolidated ingress and egress buffer views
    - Birds-eye view of the buffer



Reverie

- Threshold: $\boxed{\alpha_p}$ □ (Remaining shared pool) □ $\frac{1}{n_p}$

Configurable parameter for each queue
e.g., α_r for RDMA (ingress queues) and
α_t for TCP (egress queues)

Reverie

- Threshold: α_p □ $\boxed{\text{(Remaining shared pool)}}$ □ $\frac{1}{n_p}$

$\downarrow$

Shared pool size — total shared occupancy

Reverie

- Threshold: α_p □ (Remaining shared pool) □ $\frac{1}{\boxed{n_p}}$

$$\big\downarrow$$

Number of congested queues of type p

Reverie

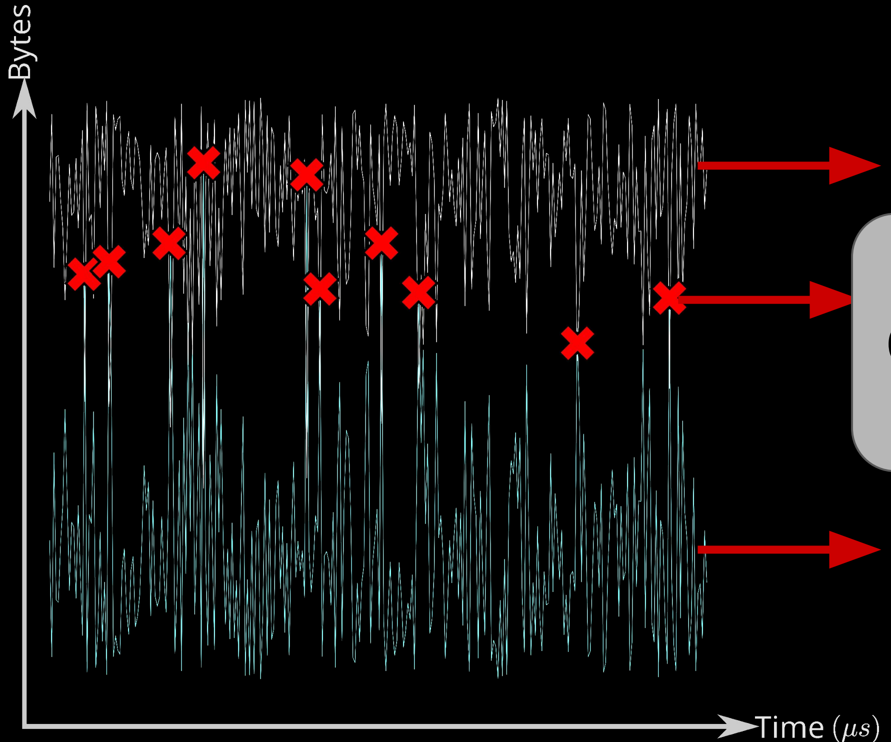
- Threshold: $\underbrace{\alpha_p \;\square\; (\text{Remaining shared pool}) \;\square\; \frac{1}{n_p}}_{\text{RDMA vs TCP}}$

Isolation ✅
Fair allocation ✅

Reverie

- Single shared buffer pool for RDMA and TCP
- Consolidated ingress and egress buffer views
  - Birds-eye view of the buffer
- Low pass filter-based admission control
  - High burst absorption

Reverie

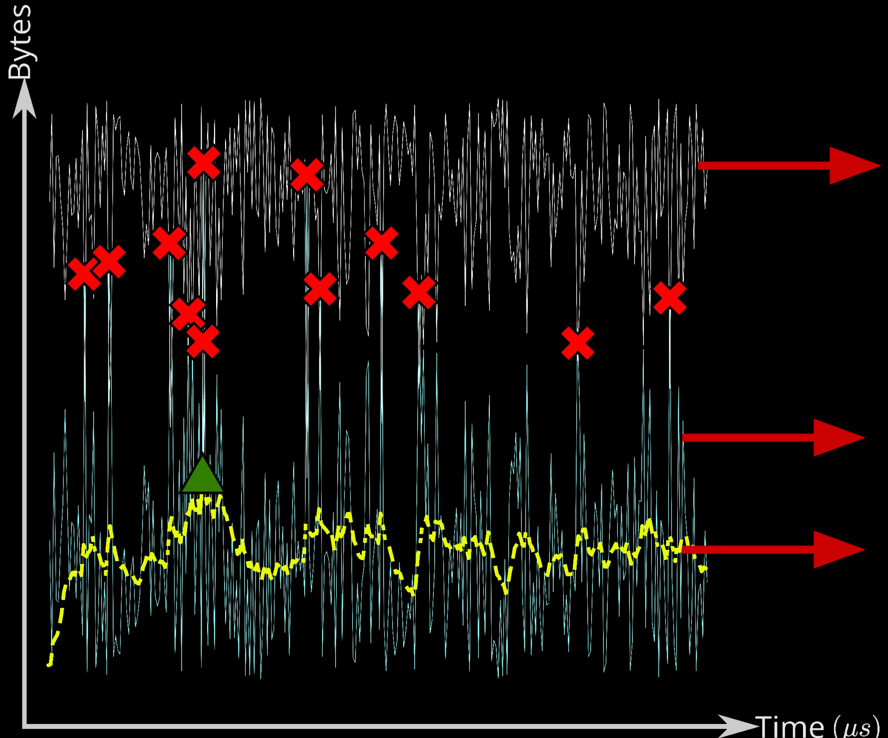


Threshold

PFC/Drop
Overreaction to transient state

Queue length

Reverie



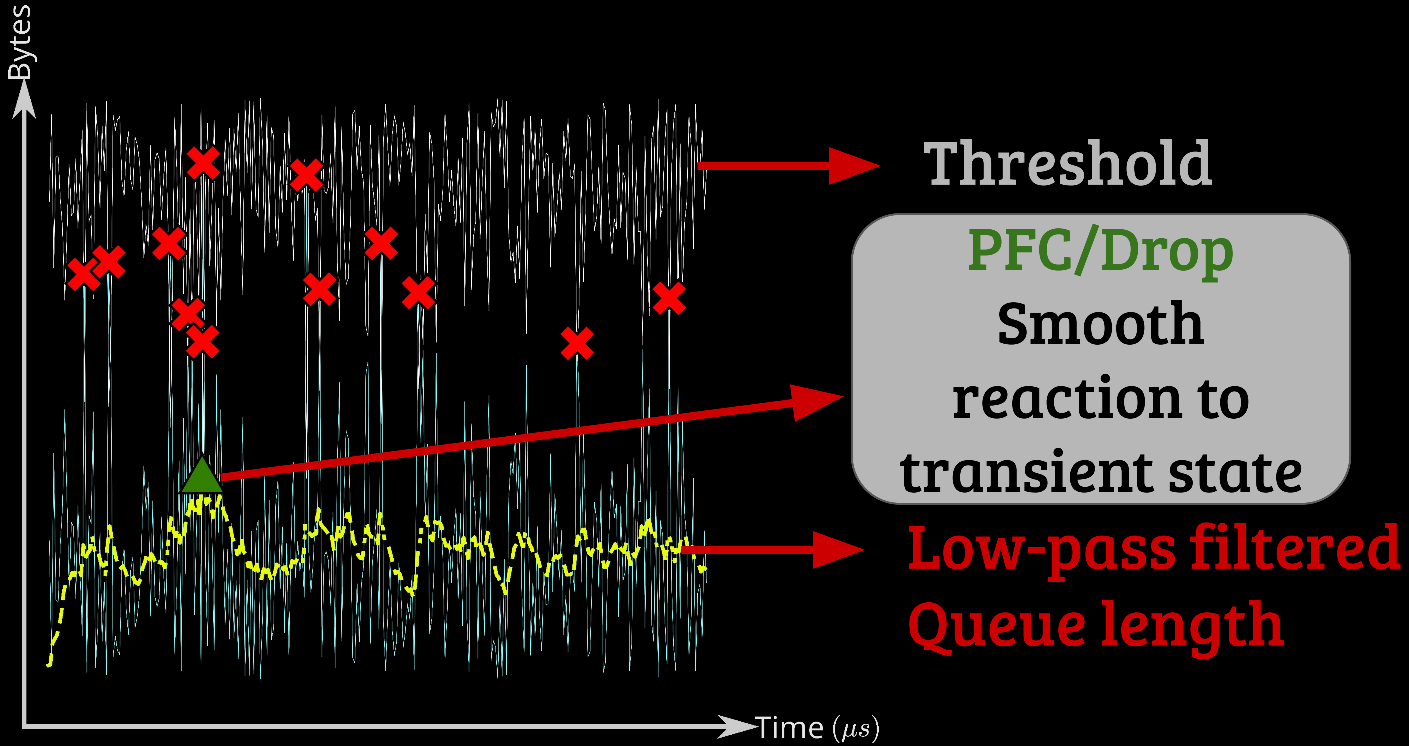
Threshold

~~Queue length~~

Low-pass filtered Queue length

REVERIE

Reverie


Bytes
Time (μs)
Threshold
PFC/Drop
Smooth reaction to transient state
Low-pass filtered Queue length

Reverie's properties

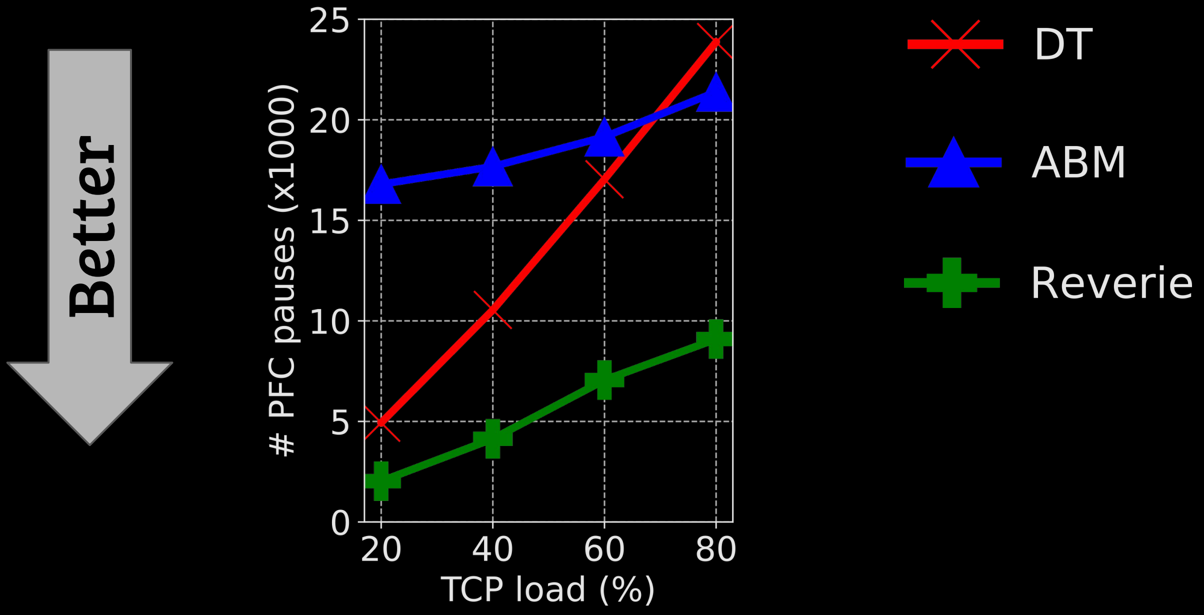
- Fair allocation across RDMA and TCP
- Steady-state isolation
- Improved burst absorption
- (Formal proofs in the paper)

REVERIE

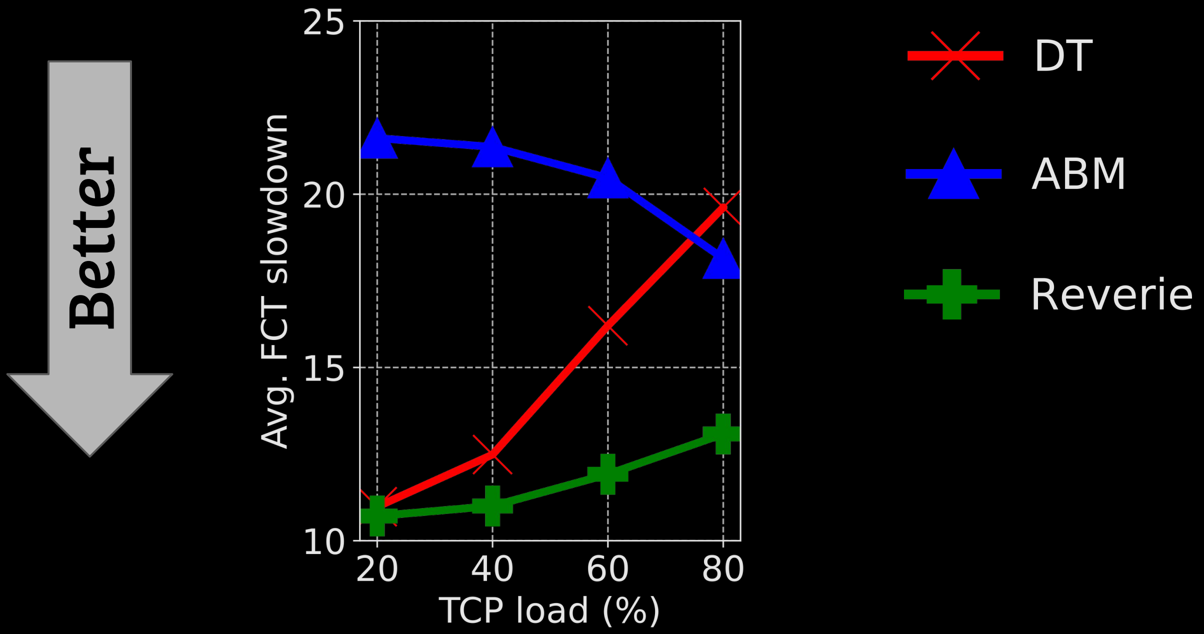
Evaluation

- Packet-level simulations using NS3
- 256 servers, 4 spine switches and 16 ToR switches
- 25Gbps NICs
- Websearch workload + Synthetic incast workload
- Shared buffer at the switches
  - Dynamic Thresholds (SONiC model)
  - ABM (SONiC model)
  - Reverie

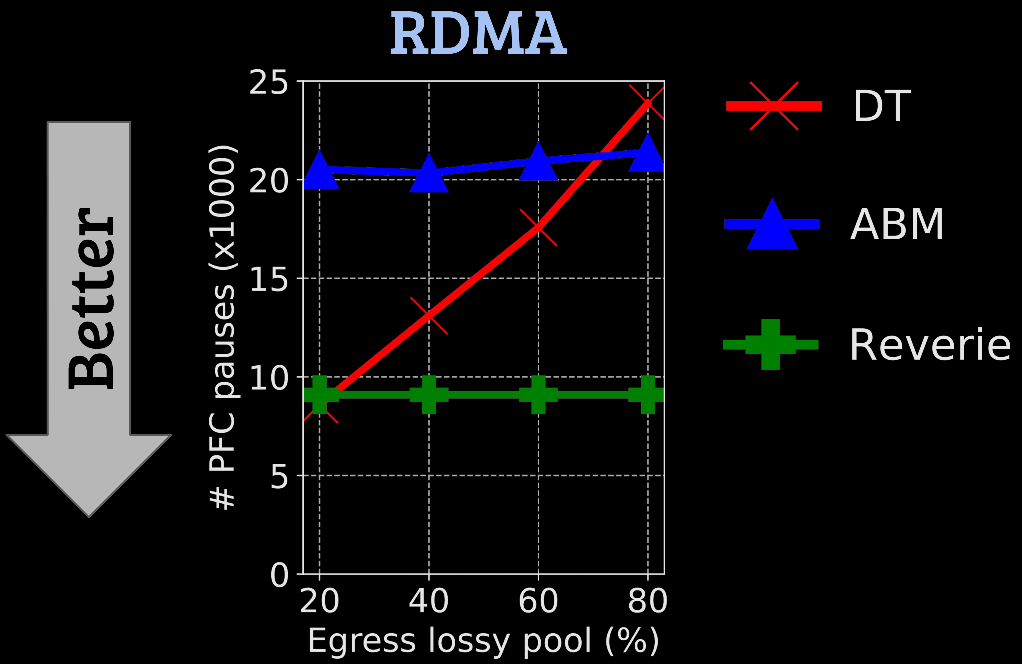
Reverie Reduces the Interactions Between TCP and RDMA

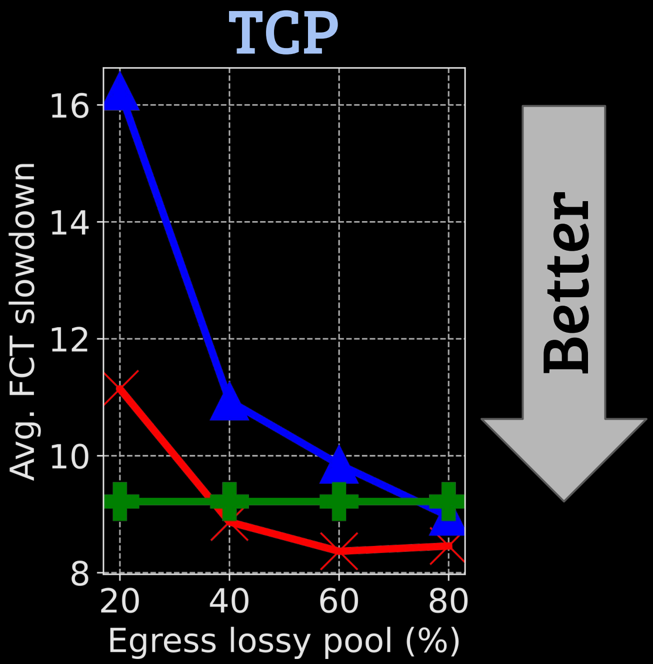


Reverie Improves Burst Absorption for RDMA



Reverie Improves the Performance of both RDMA and TCP





REVERIE

Conclusion

- Existing buffer sharing techniques cannot serve the diverse buffer needs of RDMA and TCP
- Reverie achieves isolation between RDMA and TCP
- Reverie improves burst absorption for RDMA and TCP
- Reverie improves flow completions for RDMA and TCP
- Source code: https://github.com/inet-tub/ns3-datacenter

Vamsi Addanki
vamsi@inet.tu-berlin.de
@Vamsi_DT

Wei Bai
wbai@nvidia.com
@baiwei96642217

Stefan Schmid
stefan.schmid@tu-berlin.de
@schmiste_ch

Maria Apostolaki
apostolaki@princeton.edu
@maria__apo

Thank You