南京郵電大學
Nanjing University of Posts and Telecommunications

科研写作学习作业3

汇报人：1023041110 王文怿

Augmented Queue: A Scalable In-Network Abstraction for Data Center Network Sharing

Xinyu Crystal Wu*
Rice University

Zhuang Wang*
Rice University

Weitao Wang
Rice University

T. S. Eugene Ng
Rice University

研究背景：

- 核心探究点：云数据中心网络中物理先进先出（FIFO）队列的问题
- 在今天的数据中心网络中，交换机通常配备有物理队列。这些物理队列可以吸收流量突发性以减少数据包丢失。并且许多拥塞控制（CC）算法依赖于物理队列提供的拥塞信息（如物理队列长度和排队延迟）以实现低延迟和高网络利用率。但在流量管理过程中，由于物理队列无法**为不同的流量成分提供精确的带宽**，导致了整个系统性能的不可预测性和效率低下。

Motivation：

- 目标在于解决当前数据中心网络中使用的物理队列无法为不同的流量成分提供精确带宽保障的问题。由于物理队列的这一限制，不同应用程序的流量在共享同一个物理队列时可能会互相干扰，导致性能波动极大；物理队列也不能为虚拟机提供确切的入站和出站带宽保障。

AQ设计目标：

- 来自不同应用程序的流量会干扰物理队列（如TCP+UDP）

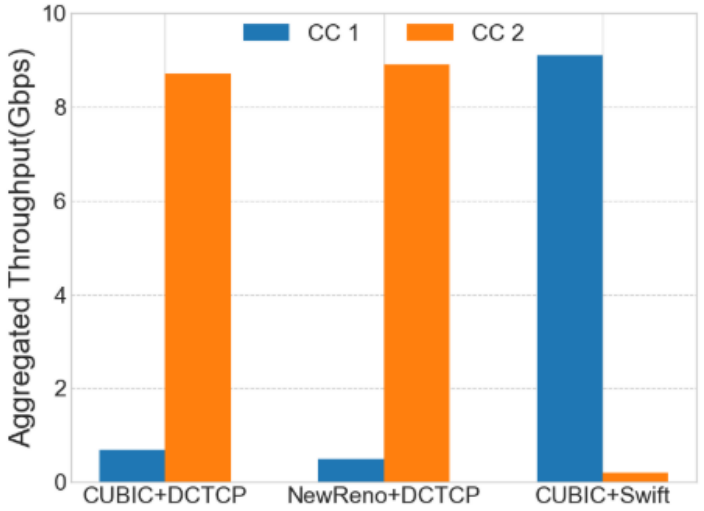
→目标1：为不同的应用提供网络隔离，每个应用有特定的带宽需求；

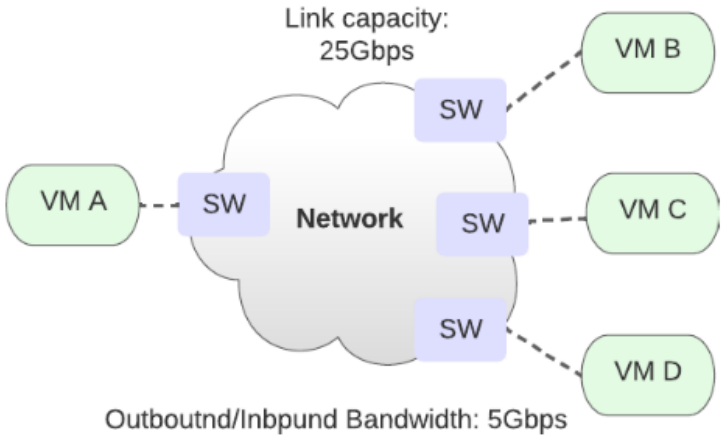
- 不同CC算法的流量难以在物理队列中共存（如吞吐优先和低时延优先）

→目标2：多个实体通过不同CC算法，可以公平地共享网络资源；

- 物理队列无法保证虚拟机的出站和入站带宽

→目标3：为不同的虚拟机提供双向带宽保证





AQ设计实现：

- 挑战：如何通过**可编程交换机**及其提供的**有状态内存**，捕获不同应用程序，不同CC算法和虚拟机流量的差异并进行速率控制。
- **AQ的数学模型**基于一种特定的流量控制逻辑，用于计算和调整流量的发送率：

  —定义**分配率**与**到达率**：分配率 (R)：为特定流量成分**预留的带宽（上界）**；

  到达率 r(t)：流量成分在时间 t 时刻的**实际使用带宽**；

  $$|r(t) - R| < \epsilon$$

  —**差异函数 d(t, t+Δ)**：函数计算在时间间隔 [t, t+Δ] 内到达率 r(t) 与分配率 R 之间的积分差异

  $$d(t, t+\Delta) = \int_t^{t+\Delta} (r(t) - R)\, dt$$

  —**A-Gap 函数 A(t)**：A-Gap 在时间 t 的值，是从初始状态开始，考虑到时间 t 之前所有积分差异的累积。初始状态 A(0)=0，对于每个后续的时间点 t，A-Gap 更新为：$\max\{0, A(t) + d(t, t+\Delta)\}$。

AQ设计实现：

- **定义**：AQ是一种可扩展的网络抽象；
- **核心：A-Gap的计算**——设计了一个精确的流算法，在数据包层面通过连续监测**到达速率与分配速率的差值**，并将其积分得到 A-Gap。A-Gap 为正，表明实际流量速率超出了分配，而零或负值则表明流量速率未超出分配。
- **调整策略**：如果 A-Gap 增大（即流量超出分配），AQ 会动态减少发送速率，反之则增加。

Algorithm 1: The streaming algorithm

Input: *aq* is an augmented queue. *aq.rate* is its allocated rate and *aq.gap* is its A-Gap.

Function A_Gap(*aq*, *pkt*):
- $\Delta = pkt.time - aq.last\_time$
- $aq.gap = \max(0, aq.gap - \Delta * aq.rate) + pkt.size$
- $aq.last\_time = pkt.time$
- **return** *aq.gap*

Algorithm 2: The framework for traffic control

Function Generate_NFB(*aq*, *pkt*):
- **if** *aq.gap* > *aq.limit* **then**
  - *aq.gap* = *aq.gap* − *pkt.size*
  - Drop(*pkt*)
- **else**
  - **if** *aq.CC* = *ECN_type* **then**
    - Apply_ECN_Marking_Actions(*aq*, *pkt*)
  - **else if** *aq.CC* = *delay_type* **then**
    - Apply_Update_Delay_Actions(*aq*, *pkt*)
  - **end**
- **end**

■ AQ实际部署：

- **控制平面（SDN）**：负责整体的网络管理策略，包括流量分配、A-Gap 参数的设置和监控；
- **数据平面：**直接处理通过交换机的数据包，根据控制平面的策略执行具体的流量控制操作。

Figure 4: Applying AQ abstraction to networks.

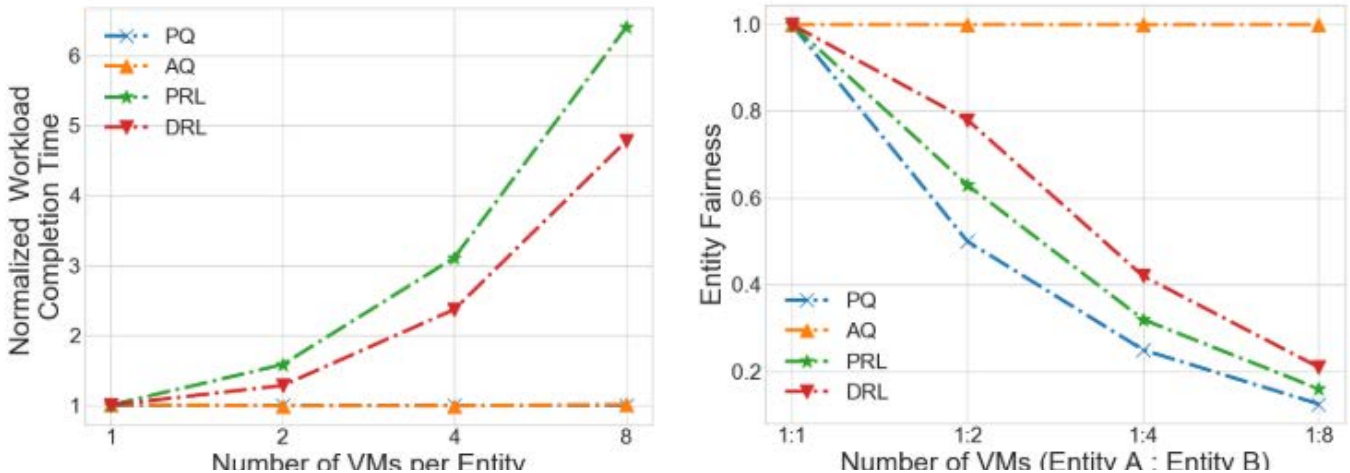
■ 有状态内存——动态寄存器共享机制

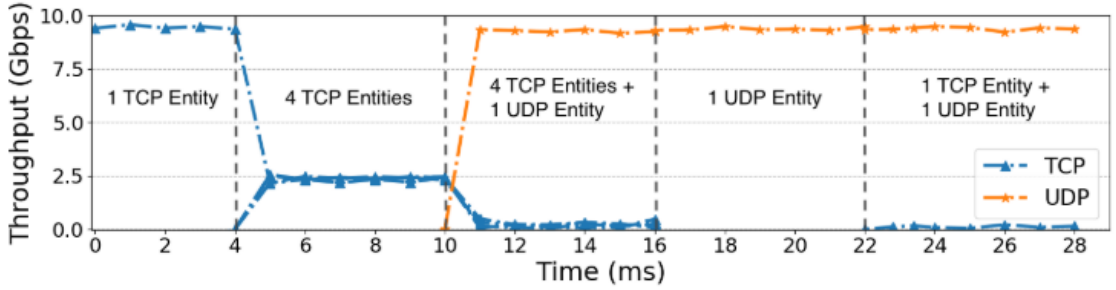
	AQ request	AQ configuration
Bandwidth demand	✓	
CC fields	✓	✓
Position profile	✓	
AQ ID		✓
AQ rate		✓
AQ limit		✓
AQ gap		✓
AQ last_time		✓

Table 1: Fields in AQ request and AQ configuration.

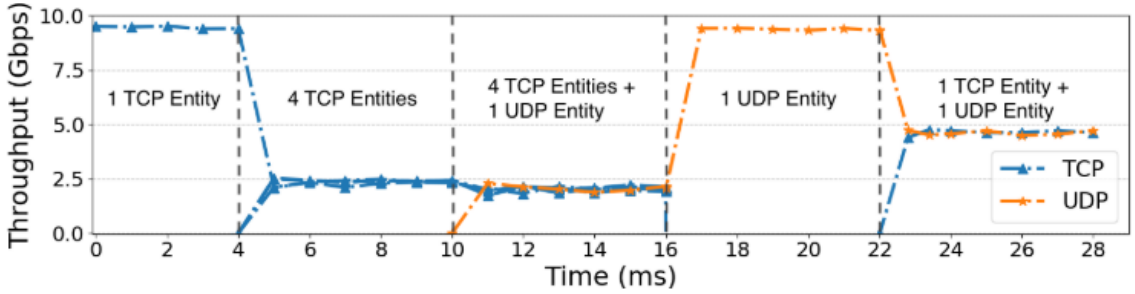
■ 实验部署以及实验结果：

- **拓扑**： NS3-BMv2软件交换机 +CloudLab
- **链路速率**：10Gbps
- **传播时延**：10us
- **流量模型**：WebSearch
- **拥塞控制算法**： CUBIC、NewReno、DCTCP、Swift、Illinois





(a) Using PQ

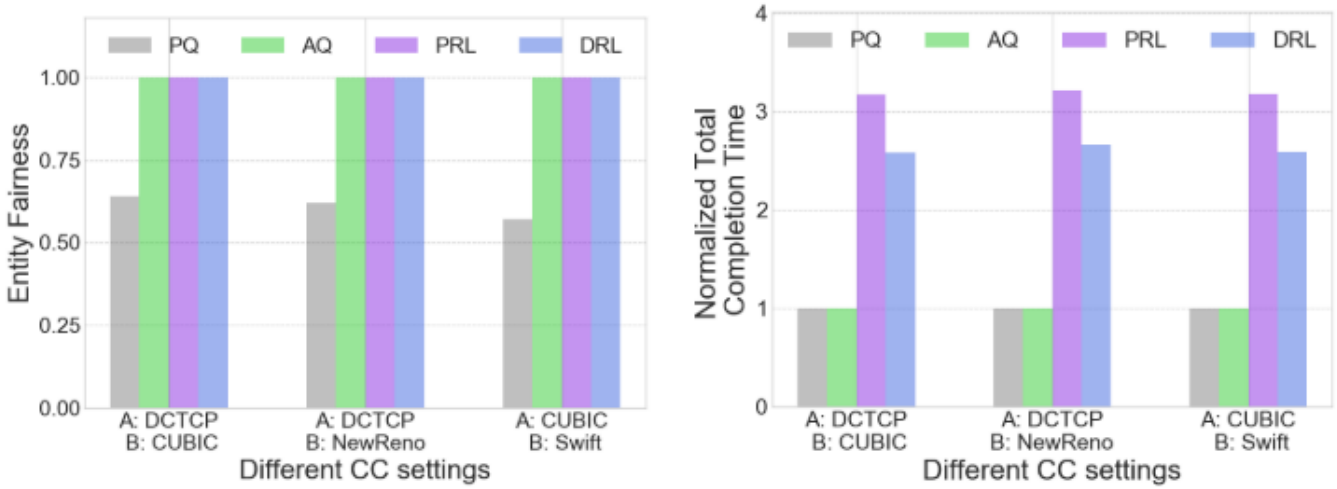


(b) Using AQ

Network Performance of CCs

■ 实验部署以及实验结果：

- **拓扑**： NS3-BMv2软件交换机 +CloudLab
- **链路速率**：10Gbps
- **传播时延**：10us
- **流量模型**：WebSearch
- **拥塞控制算法**： CUBIC、NewReno、DCTCP、Swift、Illinois



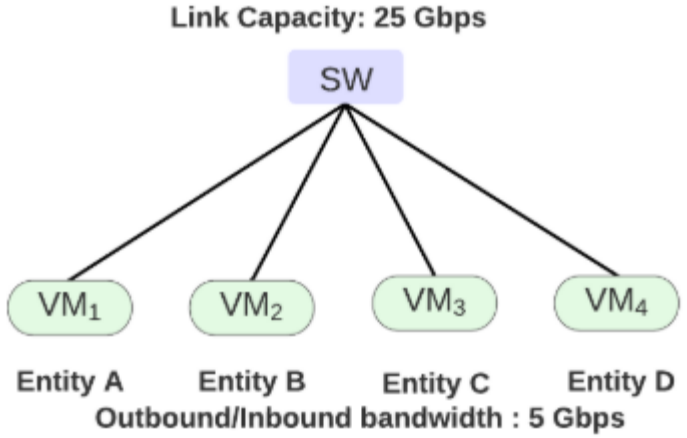
(a) The entity fairness. (b) The total completion time.

Congestion control	PQ	AQ
5 CUBIC+5 CUBIC	4.7Gbps+4.7Gbps	4.7Gbps+4.7Gbps
5 CUBIC+5 DCTCP	0.7Gbps+8.7Gbps	4.6Gbps+4.7Gbps
5 NewReno+5 DCTCP	0.5Gbps+8.9Gbps	4.7Gbps+4.7Gbps
5 Illinois+5 DCTCP	1.7Gbps+7.7Gbps	4.6Gbps+4.7Gbps
5 CUBIC+5 Swift	9.1Gbps+0.2Gbps	4.7Gbps+4.6Gbps
5 DCTCP+5 Swift	9.2Gbps+0.1Gbps	4.7Gbps+4.6Gbps
10 DCTCP+5 NewReno	9.1Gbps+0.3Gbps	4.7Gbps+4.7Gbps
10 DCTCP+5 Swift	9.2Gbps+0.1Gbps	4.7Gbps+4.6Gbps
1 UDP+3 CUBIC	8.9Gbps+0.1Gbps	2.4Gbps+2.3Gbps
+3 DCTCP+3 Swift	+0.2Gbps+0.1Gbps	+2.4Gbps+2.2Gbps

Table 2: Throughput of entities with different CC settings.

■ 实验部署以及实验结果：

- **拓扑**：Tofino交换机+八核2Ghz cpu服务器（划分为4台虚拟机）
- **链路速率**：25Gbps
- **流量模型**：WebSearch
- **拥塞控制算法**：CUBIC、NewReno、DCTCP、Swift、Illinois



Approaches	Outbound Rate Range	Inbound Rate Range
Ideal	5Gbps	5Gbps
PQ-testbed	23.1Gbps ~ 23.6Gbps	23.2Gbps ~ 23.6Gbps
PRL-testbed	4.8Gbps ~ 5.1Gbps	14.6Gbps ~ 15.3Gbps
DRL-testbed	3.1Gbps ~ 4.9Gbps	3.3Gbps ~ 4.8Gbps
AQ-testbed	4.9Gbps ~ 5.2Gbps	4.8Gbps ~ 5.2Gbps
AQ-simulator	4.8Gbps ~ 5.3Gbps	4.9Gbps ~ 5.1Gbps

Table 3: The outbound and inbound rates of VM A with different approaches.

- **额外的资源开销：**在Tofino测试平台上的数据平面资源使用情况显示，仅占用每种资源类型很小的百分比；每个AQ需要15个字节的内存

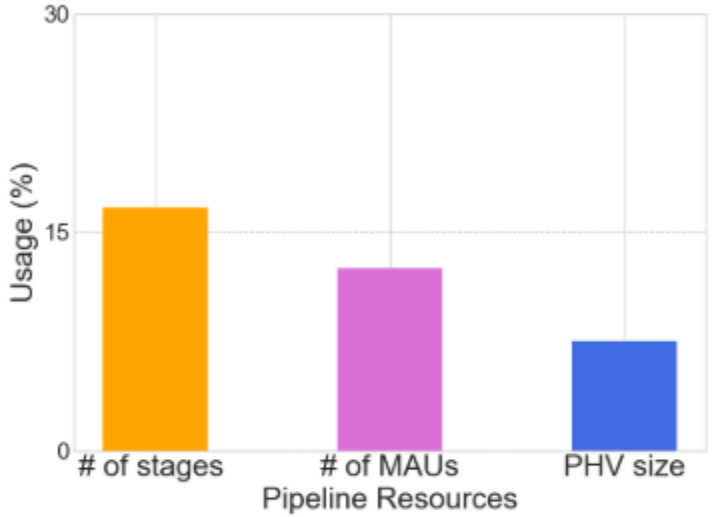


Figure 11: Usage of data plane resources on Tofino switch.

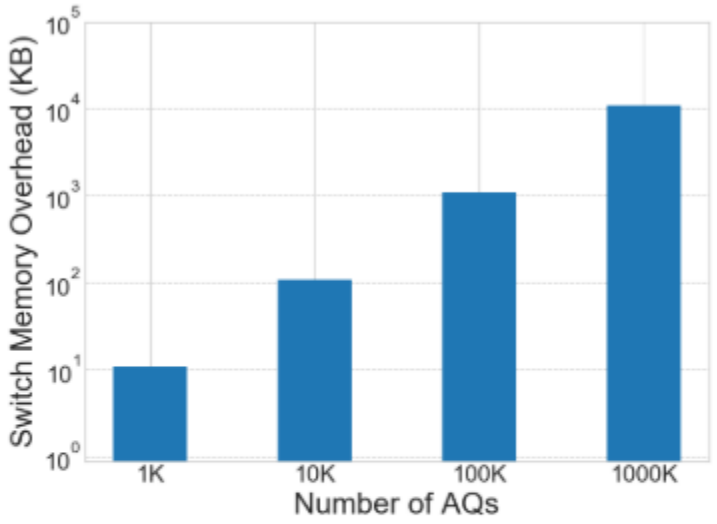


Figure 12: Memory consumption with different numbers of traffic constituents.

■ **结论：**

1. 精确的带宽保证：AQ能够为数据中心网络中的不同流量实体提供精确的带宽保证，这解决了传统物理队列技术在带宽分配精确性上的局限。

2. 高效的流量隔离：实验结果显示，AQ有效地隔离了不同应用和服务的流量，防止了资源争抢和性能干扰，从而提高了网络的整体稳定性和可预测性。

3. 拥塞控制算法的支持：AQ通过为不同算法提供必要的网络反馈（如ECN标记和延迟信息），使它们能够在同一网络环境中共存，从而优化了整体的网络性能。

4. 可扩展性和实用性：论文中的测试和评估证明，AQ是一个高度可扩展的解决方案，能够支持大规模的数据中心网络。其设计允许简单地扩展到包含数百万个流量实体的环境，而不会对性能造成显著影响。

感谢!