Active Buffer Management in Datacenters

Vamsi Addanki, Maria Apostolaki, Manya Ghobadi, Stefan Schmid, Laurent Vanbever
Active Queue Management (AQM)

Buffer Management (BM)
Controlling queueing delay

ECN

eg., RED

AQM
Trimming eg., Cut payload

Controlling queueing delay

ECN eg., RED

AQM
Controlling queueing delay

- ECN: eg., RED
- Trimming: eg., Cut payload
- Delay: eg., CoDel

AQM
Controlling queueing delay

Stable drain time

ECN  Trimming  Delay

AQM

Controlling queueing delay
Controlling queueing delay

Controlling total buffer

BM

AQM

Stable drain time

ECN

Trimming

Delay

Controlling queueing delay
Controlling queueing delay

Controlling total buffer

Better isolation

Stable drain time

ECN
Trimming
Delay

AQM

BM

DT

Controlling queueing delay
Trimming

Controlling queueing delay

Controlling total buffer

Better isolation

Stable drain time

Burst absorption

AQM

BM

DT

ECN

Trimming

Delay

Controlling queueing delay

ABM
What is ABM?

- A novel Buffer Sharing algorithm
  for datacenter switches
What is ABM?

- A novel Buffer Sharing algorithm
  — Independent AQM and Buffer Management
What is ABM?

- A novel Buffer Sharing algorithm
- Independent AQM and Buffer Management
- AQM that depends on Buffer Management
What is ABM?

- A novel Buffer Sharing algorithm
- Independent AQM and Buffer Management
- Active Buffer Management
What is ABM?

- A novel Buffer Sharing algorithm
- Independent AQM and Buffer Management
- **Active Buffer Management**
  - *Isolation across traffic priorities (eg., different SLAs)*
What is ABM?

- A novel Buffer Sharing algorithm
- Independent AQM and Buffer Management
- **Active Buffer Management**
  - *Isolation across traffic priorities (eg., different SLAs)*
  - *Bounded queue drain time (Queueing delay)*
What is ABM?

- A novel Buffer Sharing algorithm
- Independent AQM and Buffer Management
- **Active Buffer Management**
  - Isolation across traffic priorities (eg., different SLAs)
  - Bounded queue drain time (Queueing delay)
  - Better burst absorption
Background on Buffer Sharing

- Both BM and AQM calculate *drop thresholds*
Background on Buffer Sharing

- Both BM and AQM calculate *drop thresholds*
- BM calculates a threshold for every queue in a *device*
  - function of the shared buffer space
Background on Buffer Sharing

- Both BM and AQM calculate drop thresholds
- BM calculates a threshold for every queue in a device
  - function of the shared buffer space
- AQM calculates thresholds for a single queue
  - function of queue statistics
Both BM and AQM calculate drop thresholds. BM calculates a threshold for every queue in a device function of the shared buffer space. AQM calculates thresholds for a single queue function of queue statistics. BM and AQM act independently.
Hierarchical Admission Control Scheme

Packet $\xrightarrow{BM}$ True $\xrightarrow{AQM}$ True $\xrightarrow{Enqueue}$

False $\xrightarrow{Drop}$ False $\xrightarrow{Drop}$
Hierarchical Admission Control Scheme

Packet \rightarrow BM \begin{cases} \text{False} \rightarrow \text{Drop} \\ \Psi_p^i(Q) > q_p^i \rightarrow \text{True} \rightarrow AQM \end{cases} \begin{cases} \text{False} \rightarrow \text{Drop} \\ \text{True} \rightarrow \text{Enqueue} \end{cases}
Hierarchical Admission Control Scheme

Packet

\[ \Psi_p^i(Q) > q_p^i \]

BM

True

Enqueue

\[ \Phi_p^i(q_p^i, \mu_p^i) > q_p^j \]

AQM

Drop

False

Drop

False
Hierarchical Admission Control Scheme

\[ \min \left( \frac{\Psi_p^i(Q)}{BM}, \frac{\Phi_p^i(q_p^i, \mu_p^i)}{AQM} \right) \]

Packet \rightarrow BM \rightarrow True \rightarrow Enqueue

False \rightarrow Drop

False \rightarrow Drop

BM

AQM
Large Buffers

Packet

BM

\[ \Psi^i_p(Q) > q^i_p \]

True

AQM

\[ \Phi^i_p(q^i_p, \mu^i_p) > q^i_p \]

True

Enqueue

False

Drop

False

Drop

Packet

True

False

ABM
Large Buffers

AQM becomes more important!

Packet → BM

\( \Psi_p^i(Q) > q_p^i \) → True

\( \Phi_p^i(q_p^i, \mu_p^i) > q_p^i \) → True

Enqueue

False → Drop

False → Drop
Shallow buffers

Packet

$\Psi^i_p(Q) > q^i_p$

False

Drop

BM

$\Phi^i_p(q^i_p, \mu^i_p) > q^i_p$

True

AQM

True

Enqueue

Drop
Shallow buffers

Buffer Management becomes more important!

Packet → BM

- $\Psi_p^i(Q) > q_p^i$
  - True → AQM
    - $\Phi_p^i(q_p^i, \mu_p^i) > q_p^i$
      - True → Enqueue
      - False → Drop
  - False → Drop

False → Drop
Drawbacks of Dynamic Thresholds (State-of-the-art BM)

Threshold = alpha x (Remaining shared buffer)

$$T^i_p(t) = \alpha_p \cdot (B - Q(t))$$
Drawbacks of Dynamic Thresholds (State-of-the-art BM)

Threshold = alpha x (Remaining shared buffer)

\[ T^i_p(t) = \alpha_p \cdot \left( B - Q(t) \right) \]

- Priority inversion (No isolation)
- Oblivious to buffer drain time
Drawbacks of Dynamic Thresholds (State-of-the-art BM)

No Isolation Properties

Occupied Buffer (%)

Priority Inversion

# Queues (Priority-1)
Drawbacks of Dynamic Thresholds (State-of-the-art BM)

Oblivious to drain rate

Ports: $i_1$, $i_2$
Drawbacks of Dynamic Thresholds (State-of-the-art BM)

Oblivious to drain rate

Ports: $i_1$, $i_2$
Benefits and Drawbacks of Existing Approaches

- **BM** can in-principle offer isolation across queues
  - oblivious to buffer drain time
- **AQM** can in-principle offer bounded queue drain time
  - cannot fundamentally satisfy the isolation property
Our Goals

- Isolation across traffic priorities
- Bounded drain time
- Better burst absorption
  - requires both isolation and bounded drain time
Controlling total buffer

Better isolation

Burst absorption

Stable drain time

- ECN
- Trimming
- Delay

AQM

Controlling queueing delay
Trimming

AQM

Controlling queueing delay

Better isolation

Burst absorption

Stable drain time

Controlling total buffer

BM

DT

ECN

Trimming

Delay

AQM

Controlling queueing delay

ABM

ABM
Active Buffer Management

\[ T^i_p(t) = \alpha_p \cdot \frac{1}{n_p} \cdot (B - Q(t)) \cdot \frac{\mu^i_p}{b} \]

Threshold per queue for port \( i \), priority \( p \)
ABM

Active Buffer Management

$$T_p^i(t) = \alpha_p \cdot \frac{1}{n_p} \cdot (B - Q(t)) \cdot \frac{\mu_p}{b}$$

Parameter

To be set for each priority
**Active Buffer Management**

\[ T_p^i(t) = \alpha_p \frac{1}{n_p} (B - Q(t)) \cdot \frac{\mu_p^i}{b} \]

# congested queues of priority p
Active Buffer Management

\[ T^i_p(t) = \alpha_p \cdot \frac{1}{n_p} \cdot (B - Q(t)) \cdot \frac{\mu^i_p}{b} \]

Remaining shared buffer
Active Buffer Management

\[ T_p^i(t) = \alpha_p \cdot \frac{1}{n_p} \cdot (B - Q(t)) \cdot \frac{\mu_p^i}{b} \]

Normalized dequeue rate
Active Buffer Management

\[ T^i_p(t) = \alpha_p \cdot \frac{1}{n_p} \cdot (B - Q(t)) \cdot \frac{\mu^i_p}{b} \]
Properties of ABM

- Upper bounds the buffer allocated to a priority (Prevents monopoly)

\[ B_p^{max} \leq \frac{B \cdot \alpha_p}{1 + \alpha_p} \]

Depends only on the parameter set for the corresponding priority
Properties of ABM

- Lower bounds the buffer allocated to a priority (Minimum buffer guarantee)

\[ B_{p}^{\text{min}} \geq \frac{B \cdot \alpha_p}{1 + \sum_{p \in \mathcal{P}} \alpha_p} \]

*Depends only on the parameter set for all priorities*
Properties of ABM

- Upper bounds the drain time for each priority
  \( \text{(Bounded queuing delays)} \)

\[ \Gamma \leq \frac{B \cdot \alpha_p}{(1+\alpha_p) \cdot b} \]

*Depends only on the parameter set for the corresponding priority and the port bandwidth*
Evaluation

- NS3 simulations
- Leaf-Spine topology (4:1 oversubscription)
- 9.6KB buffer-per-port-per-Gbps for all switches
  - Similar to Broadcom TridentII switch
- Websearch + incast workload
ABM Improves Short Flows FCTs

- DT
- FAB
- CS
- IB (AFD + Elephant trap)
- ABM

The diagram shows the improvement in 99-pct FCT slowdown with increasing load (%). Short flows are plotted against load percentage, with ABM showing a significant reduction in 99-pct FCT slowdown compared to other methods, indicating a 76% improvement.
ABM Improves Incast Flows FCTs

- DT
- FAB
- CS
- IB (AFD + Elephant trap)
- ABM

Incast flows

99-pct FCT slowdown

Load (%)

80%
Evaluation under Shallow Buffers and Advanced CC

- **ABM**
- **DT**
- **IB (AFD + Elephant trap)**

(Buffer per port per Gbps)

99-pct FCT slowdown

Trident2 8KB 7KB 6KB Tomahawk Tofino

**DCTCP**
Evaluation under Shallow Buffers and Advanced CC

(Buffer per port per Gbps)

99-pct FCT slowdown

Trident2 8KB 7KB 6KB Tomahawk Tofino

PowerTCP
Conclusion

- Existing approach of hierarchical buffer sharing is fundamentally limited to a single dimension
- ABM offers both isolation and stable drain time; and improves burst absorption
- ABM significantly improves the performance of incast flows
- ABM works well even under shallow buffers

https://github.com/inet-tub/ns3-datacenter
Thank you

https://github.com/inet-tub/ns3-datacenter