Performance Analysis of Binary Marking Congestion Control Protocol

Bhavin T. Sorathiya[1] Yogesh H. Zambare[2]
[1]Student, M.E., [2]Assistant professor
[1,2]Electronics & Communication Department
[1,2]Marwadi Education Foundation's Faculty of P.G. Studies & Research in Engineering & Technology

Abstract--- Binary Marking Congestion Control protocol (BMCC) obtain high resolution estimates of congestion by combining the explicit congestion notification (ECN) marks of multiple packets, and using this to guide multiplicative increase, additive increase, multiplicative decrease (MI-AI-MD) window adaption method. BMCC achieved high utilization, high throughput and negligible packet loss even on high bandwidth delay product networks. We show the comparison of BMCC with XCP protocol by ns-2 simulator.

I. INTRODUCTION

The Transmission Control Protocol (TCP) has been instrumental to the successful development and growth of the Internet and its applications. However, recent advances in wired and wireless communications technology have led to a tremendous growth in the range of path bandwidth-delay products (BDPs) in the Internet. There has simultaneously been an increase in the diversity of applications carried over the Internet (e.g., voice over IP, video conferencing, and social networking). These advances have stressed the congestion control algorithm in TCP and the need for more efficient, fair, robust, and easy-to-deploy congestion control protocols is increasingly important. So for this we use BMCC [1] protocol. There are several methods of obtaining high resolution congestion estimate using existing ECN [4] bits of streams of packet. Higher resolution estimate can be obtain using side information in packet headers to indicate how to interpret the ECN mark of given packet. Adaptive deterministic packet marking (ADPM) [2] implicitly adapt its effective quantization resolution based on the dynamics of the value and obtains a resolution of 1/n after receiving n packets, for n up to 2^{16} or beyond. ADPM provide MSE that is several order of magnitudes smaller than the estimator based on random marking of packets [5] or deterministic marking with static quantization [6].

II. PROTOCOL

BMCC estimate congestion in terms of load factor which is weighted sum of the link utilization and queuing delay. The maximum load factor of flow's path is communicated using ADPM to the senders, which respond to the MI-AI-MD. The components of BMCC are as follows.

A. *BMCC Router*

A BMCC router divides time in to intervals of length t_p and computes load factor in each interval as

$$f = \lambda + k_1 q_{av} / \gamma_l C_l t_p \qquad (1.1)$$

Where, λ is the bytes received during t_p, C_l is the link capacity in bytes/second, $\gamma_l \leq 1$ is the target utilization, q_{av} is an estimate of the average queue length in bytes, and $k_1 \leq 1$ controls how fast to drain the queue.

The router conveys its load factor to the sender by applying ADPM to packets' ECN bits as follows.ECN bits on an unmarked packet are initially $(10)_2$ and routers set these bits to $(11)_2$ to indicate congestion. Choose u to be a "severely congested" load factor. BMCC marks the packet with $(11)_2$ if $f \geq u$ or the packet already contains a mark $(11)_2$. Otherwise, ADPM computes a deterministic hash h of the packet contents, such as the 16-bit IPid field in the IPv4 header or the checksum of the payload in case of IPv6. This hash is compared to f, and the packet is marked with $(01)_2$ if $f > h$ or left unchanged otherwise. At the receiver, the ECN bits will reflect the state of the most congested router on the path.

B. *BMCC Receiver*

As part of ADPM, the receiver maintains the current estimate, $\hat{f}$ of the load factor at the bottleneck on the forward path. When a packet is received, this estimate is updated as,

u, if $b_{ecn} = (11)_2$

h, if ($b_{ecn} = (10)_2$ and $h < \hat{f}$)

or ($b_{ecn} = (01)_2$ and $h > \hat{f}$)

$\hat{f}$, otherwise

Where b_{ecn} refers to the two ECN bits in the IP header of the received packet. The estimate $\hat{f}$, is sent to the sender using TCP options. The receiver's estimate will lag behind the true value, except that values over u signaled immediately to indicate severe overload.

The resolution depends on the fraction of packets that hash to a particular range. For BMCC, values of f below a threshold η_0 are rounded up to η_0, and the hash is such that 1/4 of packets hash to values in (η_0,η) for some design parameter η , 1/4 of packets hash to $(\eta,1)$, and 1/2 hash to $(1,u)$.

C. *BMCC Sender*

Achieve high bottleneck utilization and fair bandwidth allocation with low fluctuation in rates, BMCC uses three modes of operation, based on whether the load is,

(1) Low load ($\eta_0 \leq \hat{f} < \eta$): To increase bottleneck utilization when the load is low, sources apply MI with factors proportional to $(1- \hat{f})/ \hat{f}$. In particular, flows with round-trip time (RTT) equal to t_p apply

$$w(t+T)= w(t)(1+k_2(1- \hat{f})/\hat{f} \qquad (1.2)$$

where T is the RTT of the flow and k_2 is the step size. BMCC use k_2 = 0.35 which is in the range $k_2 \in (0, 1/2)$. BMCC aims to give equal rate to flows with different RTTs.

Since flows with large RTTs update less often, the rule $w(t+T)= w(t)(1+k_2(1- \hat{f})/\hat{f})^{T/tp}$ is used so that windows grow at a rate independent of *T*. To mirror the benefits of traditional slow start, new flows remain in MI until $\hat{f}$ first reaches 1.

(2) Medium load ($\eta \leq \hat{f} <1$): When utilization is high, BMCC uses AIMD to achieve fairness. In medium load, sources apply AI,

$$w(t+T)= w(t)+\alpha \quad (1.3)$$

with $\alpha= (T/t_p)^2$ chosen to cause the equilibrium window to be proportional to the flow's RTT, giving RTT fairness.

(3) Overload($1 \leq \hat{f} < u$): When the load factor is greater than 1, sources use MD $w(t+T)= w(t)\beta(\hat{f})$ where the decrease factor $\beta(\hat{f}) = \beta_{max} - \Delta\beta(\hat{f}-1)/u-1$ varies linearly in $[\beta_{min},\beta_{max}] \in (0, 1)$, *u* is the maximum value of *f* that ADPM can signal, and $\Delta\beta =\beta_{max}-\beta_{min}$. Note that the MD factor is a function of $\hat{f}$. This implies that when $\hat{f}$ is high, sources back off by a greater amount, which improves responsiveness to congestion. Otherwise, sources back off by a smaller amount to reduce fluctuations in sending rates and maintain high utilization.

Parameter	Value	Purpose
t_p	200 ms	Load factor estimation interval
k_1	0.5	Control how fast to drain queue
k_2	0.35	Stability constant
α	$(T/t_p)^2$	AI parameter of flow with RTT T
β_{max}	0.875	Maximum back-off factor
β_{min}	0.65	Minimum back-off factor
η_0	0.15	Initial estimate of load factor
η	0.75	Mode threshold

Table 1:BMCC Parameter Setting

III. SIMULATION RESULTS

To evaluate the performance of the BMCC we use ns-2 simulator. The focus is on comparing the performance of BMCC with XCP [3] to overcome the limitation of TCP in high BDP networks. The bottleneck buffer size is set to the larger of BDP (i.e., the product of the bottleneck capacity and round-trip propagation delay) or two packets per flow. We use dumbbell topology.

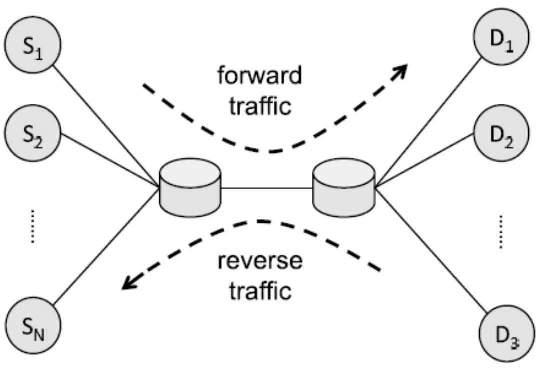

Fig. 1: Dumbbell Topology

Data packets are 1000 B and ACKs are 40 B and round-trip propagation delay is 80 ms. All simulation are run for 110 s except those in which the round-trip propagation delay is varied. In this case, the simulations are run for 300 s. The result represents the 10 simulation runs. We use single bottleneck topology and two way traffic with five 5 FTP flow on both the forward and reverse path. We vary different parameters and study their effect on performance.

A. Varying Bottleneck capacity

The bottleneck capacity is varied between 0.1 Mbps to 1000 Mbps, keeping everything else fixed.

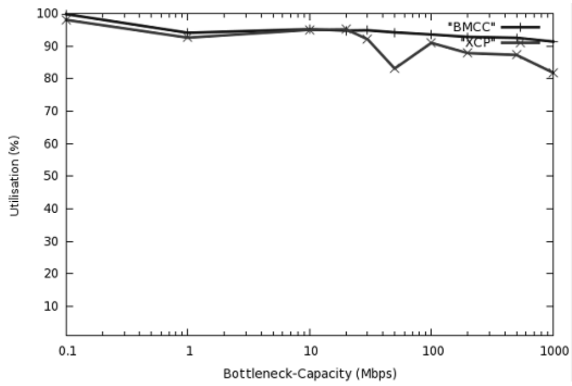

Fig. 2: Bottleneck utilization with varying capacity.

Fig. 2, 3 and 4 show that BMCC achieved high utilization(≥ 90%), negligible packet loss and high throughput while XCP achieve ≥ 85% utilization.

B. Varying Round-trip Propagation Delay

The round-trip propagation time is varied between 1 ms to 1 s. Bottleneck link capacity is set to 100 Mbps. Fig. 5, 6 and 7 show that BMCC again achieve high utilisation ≥ 80%, negligible packet loss and high throughput. While XCP utilization is less when link round-trip time is less than 10 ms.

C. Varying Number of Long Lived Flows

The number of long-lived flows is varied from 1 to 1000. Bottleneck link capacity is 10 Mbps.

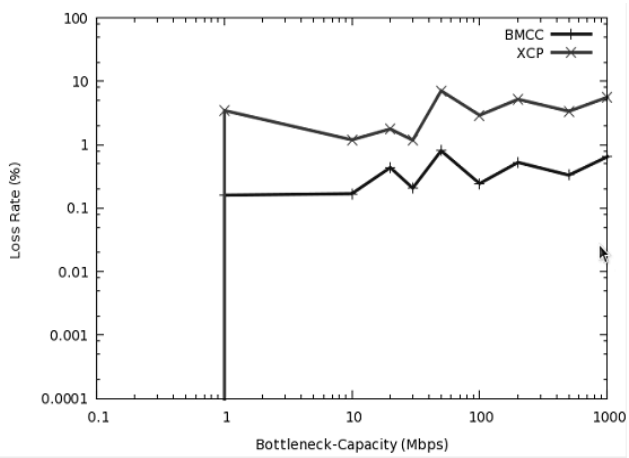

Fig. 3: Loss rate with varying capacity

Fig. 8 shows that BMCC again maintain high utilization (≥ 90%), while XCP also have high utilization when bandwidth is high but when bandwidth is low XCP utilization is below 50 %. Fig. 9 and 10 shows that BMCC have negligible packet loss and high throughput, while XCP slightly high packet loss compare to BMCC and less throughput when number of long lived flows is less.

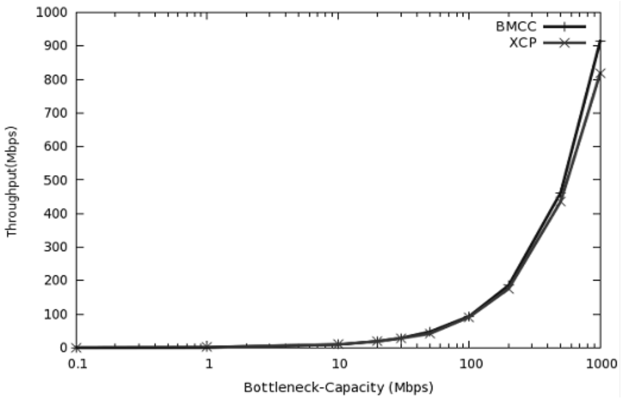

Fig. 4: Throughput with varying capacity

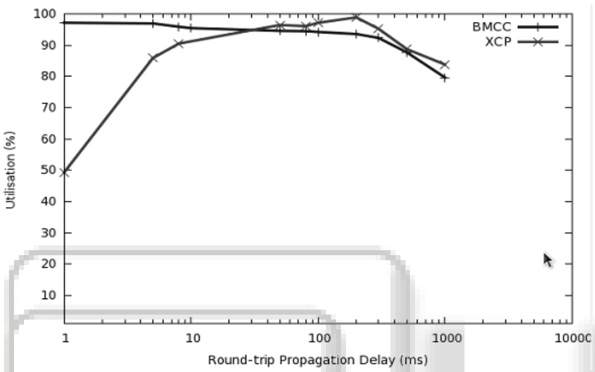

Fig. 5: Bottleneck utilization with varying round-trip propagation delay.

D. Varying Bottleneck Buffer Size

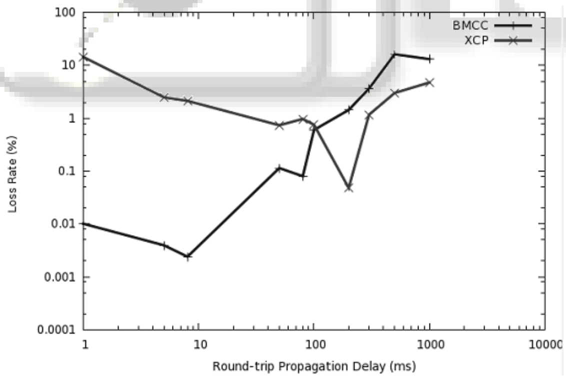

Fig. 6: Loss rate with varying round trip propagation delay.

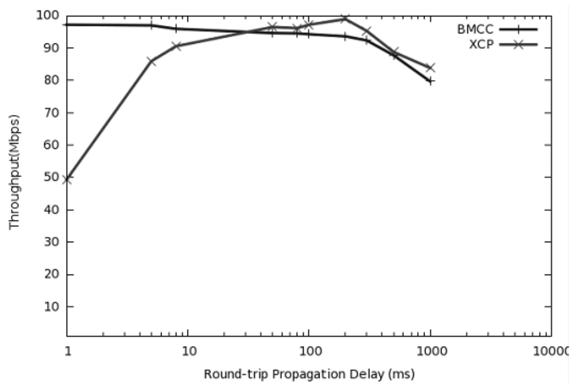

Fig. 7: Throughput with varying round trip propagation delay, with link capacity is 100 Mbps.

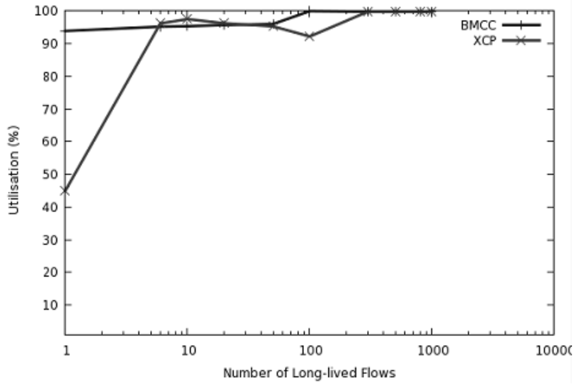

Fig. 8: Bottleneck utilization with varying number of long lived flows.

While varying buffer capacity of the bottleneck link. Fig.11, 12 shows that BMCC achieve high utilization and negligible packet loss, XCP also maintain high utilization and less packet loss. Fig. 13 shows that BMCC and XCP both achieve high throughput. Bottleneck link capacity is 1 Mbps.

CONCLUSION

This paper investigates the performance of the Binary Marking Congestion Control protocol and comparing with eXplicit control protocol. Result show that BMCC achieve high bottleneck utilisation, negligible packet loss and high throughput even on high bandwidth delay product networks and BMCC has better performance than XCP protocol.

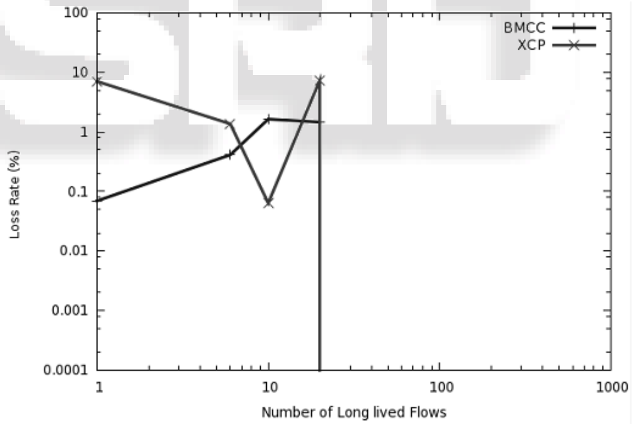

Fig. 9: Loss rate with varying number of long lived flows.

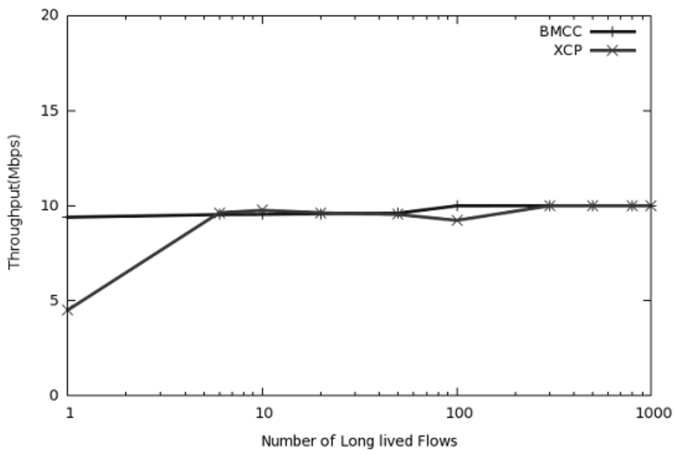

Fig. 10: Throughput with varying number of long lived flows, with link capacity is 10 Mbps.

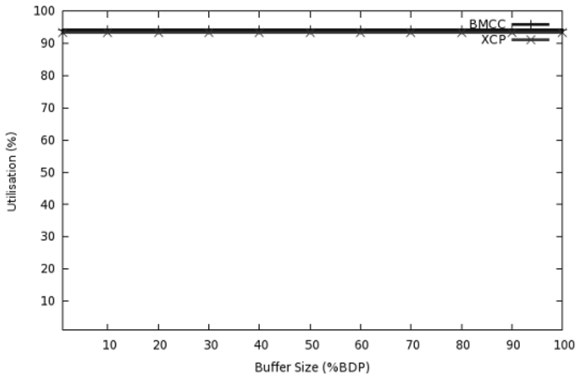

Fig. 11: Bottleneck utilization with varying buffer size.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my parents; I owe my deepest appreciation and admiration. Throughout my life, they have encouraged me in everything I ever wanted to do. Even at great hardship to themselves, they made sure I had every opportunity they could possibly give me. I hope that my accomplishments will make all of them proud.

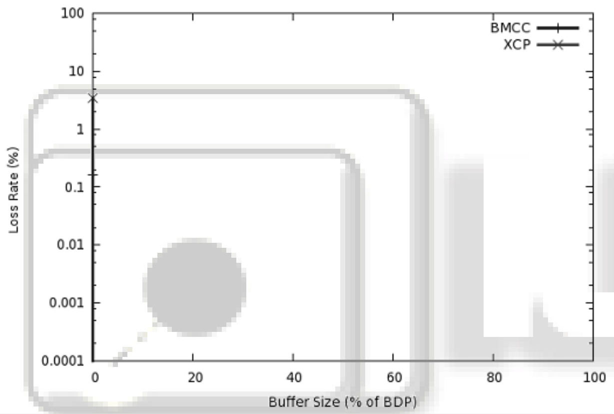

Fig. 12: Loss rate with varying buffer size.

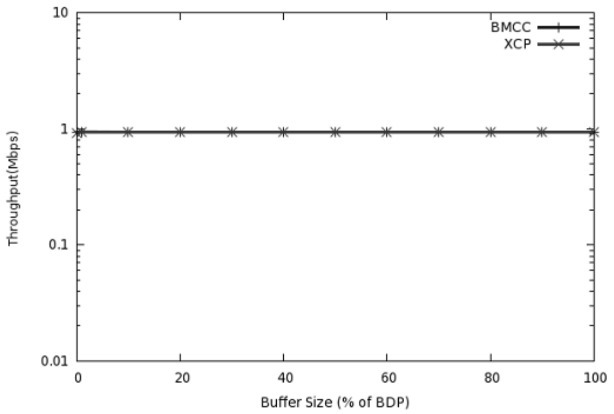

Fig. 13: Throughput with varying buffer size, with link capacity 1 Mbps.

REFERENCES

[1] Ihsan Ayyub Qazi, Lachlan L. H. Andrew and Taieb Znati, "Congestion Control With Multipacket Feedback" in proc IEEE INFOCOM, 2012, VOL.20, N0.6, DECEMBER 2012.
[2] L. L. H. Andrew, S. V. Hanly, S. Chan, and T. Cui, "Adaptive deterministic packet marking," IEEE Commun. Lett., vol. 10, no.11, pp. 790–792, Nov. 2006.
[3] D. Katabi, M. Handley, and C. Rohrs, "Internet congestion control for high bandwidth-delay product networks," in Proc. ACM SIGCOMM, 2002, pp.89-102
[4] K. K. Ramakrishnan and S. Floyd, "The addition of explicit congestion notification (ECN) to IP," IETF, RFC 3168, Sep. 2000.
[5] S. Athuraliya, V. Li, S. Low, and Q. Yin, "REM: Active queue management," IEEE Netw., vol. 15, no. 3, pp. 48–53, May 2001.
[6] H.-K.. Ryu and S. Chong, "Deterministic packet marking for max-min flow control,," IEEE Commun. Letters, vol. 9, no. 9, pp. 856–858, Sep. 2005