Denial of Service Attacks in Networks with Tiny Buffers

Veria Havary-Nassab, Agop Koulakezian,
Department of Electrical and Computer Engineering
University of Toronto
{*veria, agop*}*@comm.toronto.edu*

Yashar Ganjali
Department of Computer Science
University of Toronto
yganjali@cs.toronto.edu

***Abstract*- Recently, several papers have studied the possibility of shrinking buffer sizes in Internet core routers to just a few dozen packets under certain constraints. If proven right, these results can open doors to building all-optical routers, since a major bottleneck in building such routers is the lack of large optical memories. However, reducing buffer sizes might pose new security risks: it is much easier to fill up tiny buffers, and thus organizing Denial of Service (DoS) attacks seems easier in a network with tiny buffers. To the best of our knowledge, such risks have not been studied before; all the focus has been on performance issues such as throughput, drop rate, and flow completion times.**

In this paper, we study DoS attacks in the context of networks with tiny buffers. We show that even though it is easier to fill up tiny buffers, synchronizing flows is more difficult. Therefore to reduce the network throughput, the attacker needs to utilize attacks with high packet injection rates. Since such attacks are easily detected, we conclude that DoS attacks are in fact more difficult in networks with tiny buffers.

I. Introduction

Denial of service (DoS) attacks pose many threats to the networking infrastructure. They consume network resources such as network bandwidth and router CPU cycles with the malicious objective of preventing or severely degrading service to legitimate users. A major class of DoS attacks prevent legitimate flows from going through the system by filling up router buffers, and consequently causing significant packet drops. Such attacks usually need to inject high-rate flows in order to keep buffers full at all times, and starve legitimate flows. These high-rate attacks, though harmful to the network, are easily detected by the DoS traffic pattern monitors and neutralized [1], [2]. Any DoS attack that manages to degrade the performance of the system through low-rate flow injections is more dangerous as it remains undetected. Recently, Kuzmanovic and Knightly introduced and analyzed a class of low-rate DoS attacks [3] using a vulnerability in the congestion handling and timeout mechanism of TCP [4]. These attacks cause the network to deny bandwidth to TCP flows while operating at a sufficiently low average rate to elude detection by counter-DoS mechanisms.

During the past few years, a series of papers have studied the possibility of reducing buffer sizes in Internet core routers from millions of packets to only a few dozen packets [5]–[11]. If proven right, these results can open doors to building all-optical routers, since a major bottleneck in building such routers is lack of large optical memories. All these results focus on the performance issues such as throughput, drop rate, and flow completion times, and to the best of our knowledge, security risks associated with reducing buffer sizes have not been studied before.

Decreasing the buffer size is a double edged sword in networks under DoS attacks. On the one hand, smaller buffer sizes seem to make the attacker's job easier in filling up the buffers. On the other hand, a DoS attack on a router with a tiny buffer might affect a smaller portion of the total flows, hence may fail in synchronizing them, which results in a less effective attack. The question would be then, which one is the case in a real network. Are DoS attacks simpler to create in a network with tiny buffers? Or, does tiny buffers force the attacker to increase their rate, and thus make it easier to detect such an attack?

In this paper, we answer these questions by investigating the performance of a network under a low-rate DoS attack for different buffer sizes. We perform simulations in a variety of settings to evaluate the possibility of organizing low-rate DoS attacks against a network with tiny buffers. We show that even though forcing packet drops in a network with tiny buffers is easier, unless this is kept up for a long time, the DoS attack cannot synchronize the flows and degrade the performance of the network. Forcing drops for long periods is also not a good idea, as it will result in a high volume of traffic which can be detected using network monitoring systems. We also show that

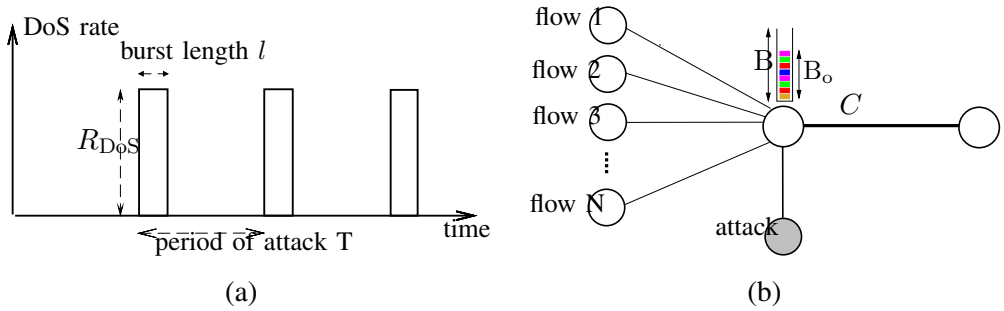


Fig. 1. (a) The DoS attack flow signal, (b) Topology of the sample network.

- Internet core routers need a minimum buffer size for normal operation of TCP flows. Reducing the buffer size below this level will result in a continuous drop of packets and thus causes instability in the network.
- A network with higher number of flows will perform better under low-rate DoS attacks. To impact a high number of flows, the attacker needs to increase the packet injection rate, which makes it easier to be detected. Since core routers carry much more traffic than edge routers, this means DoS attacks against core routers are less effective than edge routers.
- While increasing the duration of the attack degrades the performance of flows, for equal attack durations, tiny buffer networks perform much better than networks with typical buffer sizes.
- The Random Early Detection (RED) queue management scheme makes networks more resilient to the studied low-rate DoS attacks.

The rest of this paper is organized as follows. Section II discusses the TCP congestion handling mechanism and specifications of the low-rate DoS attack. Simulation setup, parameters and softwares used are listed in section III. Section IV describes the results of the simulations for scenarios with different parameters. Section V summarizes the results and concludes the paper.

II. Description of the Attack

Here, we start with a brief description of how TCP Reno works. We use TCP Reno and its successor, New Reno, as they are the most commonly used congestion control schemes today, and most other variants of TCP today have a similar behavior. We will study other variants of TCP later in the paper. TCP Reno detects packet loss through either a timeout due to not receiving ACKs or through receiving three duplicate ACKs. In the first case, the source reduces the congestion window to one packet and retransmits the packets after waiting for a period of Retransmission Time Out (RTO). Upon further loss, RTO doubles with every subsequent timeout. In the second case, it decreases the congestion window by half, performs a fast retransmit, and enters a phase called fast recovery. The TCP Congestion Control RFC [4] shows that TCP achieves the greatest throughput if RTO is at least one second [3].

The above algorithm, while necessary for the robustness of TCP, raises the opportunity of a low rate denial of service (DoS) attack. To understand these low-rate DoS attacks, let us consider a periodic on-off "square-wave" (Fig. 1-(a)) attack consisting of short duration bursts (l) that repeat with a fixed period (T) close to the RTO of the TCP flows. This choice of the DoS attack period makes the TCP flow continually incur packet drops as it tries to exit the timeout state, and consequently, fails to exit timeout and gets forced to near zero throughput. To see this, consider a simple network consisting of N nodes all transmitting packets using TCP towards a common destination through a shared bottleneck link as illustrated in Fig. 1-(b). This is a classic topology considered in congestion control studies. As it is unlikely to have more than one point of congestion on a single path, such topology can be adopted where the congested link generally represents a path encountering congestion.

We call the intermediate node the *router*, the destination node the *server* and the link between the router and the server *the bottleneck link*. The buffer is shared among the flows as shown in the figure. The attack flow now bursts and fills up the buffer causing every new packet from the flows to drop (when the buffer is using drop tail strategy). These drops will consequently put the flows into a timeout and slow start phase if the attack's rate and duration are chosen properly [3].

In order for the attack to be successful, the traffic shape of the attack should be selected based on the following criteria. First, the rate of the attacker flow(R_{DoS})

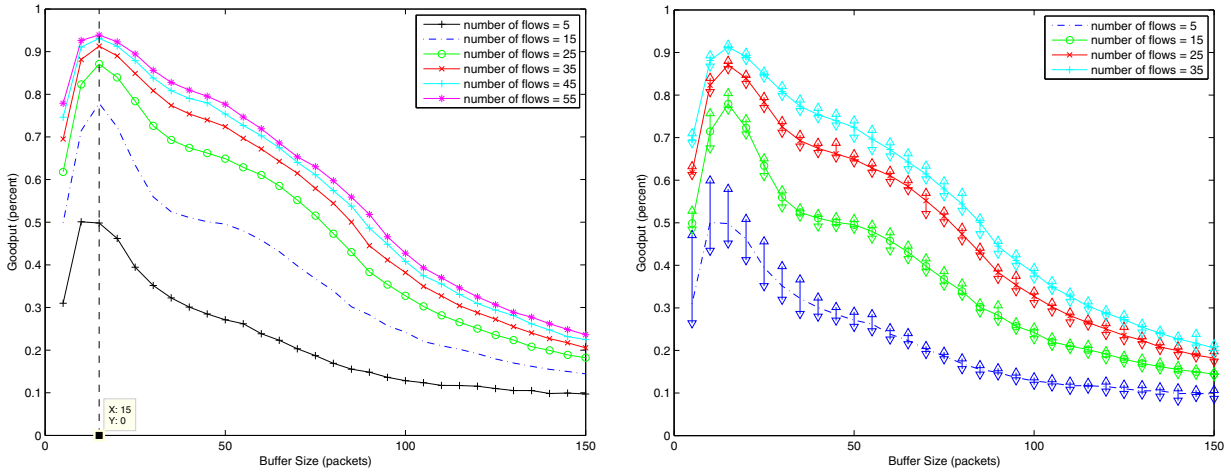


Fig. 2. (a) Goodput vs. buffer size for different number of flows $N = 5 - 55$. Note that the best goodput is achieved at a buffer size of 15 packets (b) Average and variations of the goodput vs. buffer size for different number of flows $N = 5 - 35$

and the burst length (l) are to be selected high enough to cause packet drops in the buffer. On the other hand, in order to conceal this malicious activity from the network monitors, each of the rate and the burst length cannot go arbitrarily high.

Fig. 1-(a) illustrates the rate of the attack versus time. Mathematically, the burst time l and the attack rate R_{DoS} depend on the bottleneck capacity and buffer size. Let B denote the buffer size at the bottleneck link and C denote the capacity of the bottleneck link (Fig. 1-(a)). In order for the attack to cause the buffer to overflow and hence lead to packet drops from the other flows, l should satisfy the following equation:

$$l = \frac{B - B_{\text{o}}}{R_{\text{DoS}} + R_{\text{TCP}} - C} \qquad (1)$$

Where B_{o} and R_{TCP} denote the buffer occupancy just prior to the attack and the aggregate rate of the TCP flows on the bottleneck link respectively.

As RTO doubles with each subsequent timeout, with a single flow, the time out period will be 3, 7, 15, ... seconds. However, higher number of flows can be forced into continual timeouts if an attacker creates outages with period minimum RTO, provided that all have an identical minimum RTO. Thus, the period of the attack, T (Fig. 1-(a)) is chosen to be equal to the proposed minimum RTO, one second, as recommended by Allman and Paxton [4]. Based on the results presented in [3], the DoS attack characterized with these parameters, can dramatically degrade the throughput of the network. (80% degradation on a 1.5Mbps link using FTP traffic and a 100ms burst length).

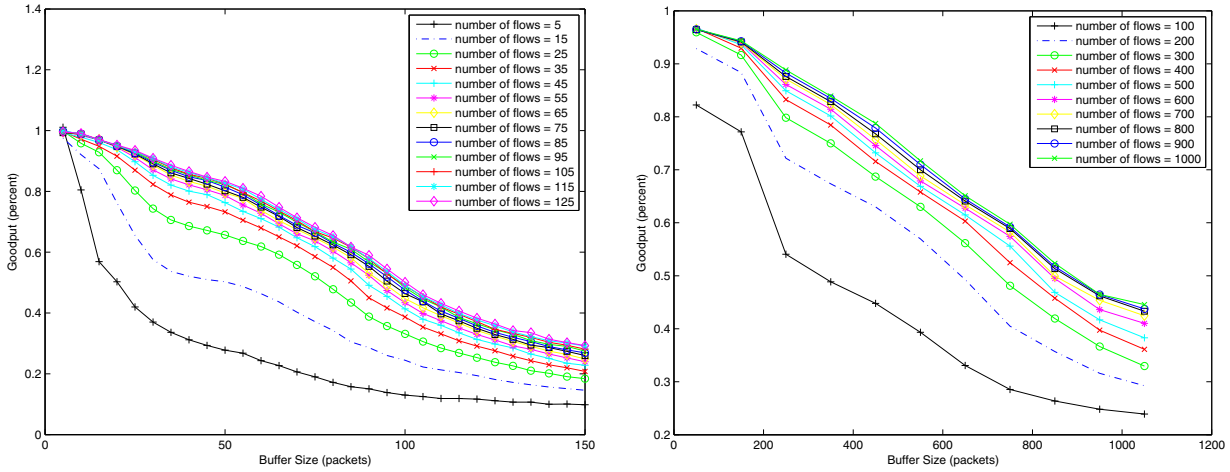
III. DoS Attacks in Networks of Tiny Buffers

As mentioned earlier, it is not intuitively easy to guess the impact of reducing buffer sizes on the class of DoS attacks that we just described. On the one hand, smaller buffer sizes can make the attacker's job easier in filling up the buffers. On the other hand, a DoS attack on a router with a tiny buffer might affect a smaller portion of the total flows, hence may fail in synchronizing them, specially while the aggregate number of flows is relatively high. The question would be then, which one is the case in a real network?

We study the effect of changing the buffer size on the severity of a DoS attack using *ns-2* simulations. We use the topology illustrated in Fig. 1-(b) along with the following parameters.

$$\begin{aligned} C &= 10Mbps, RTT \in [90ms\ 110ms] \\ R_{\text{DoS}} &= 20Mbps, l = \frac{B}{R_{\text{DoS}} - C}, \\ T &= 1s, \text{Runs} = 50 \end{aligned}$$

In calculating l, the effect of B_o and R_{TCP} is ignored, which means that the burst length can still be smaller. With the above parameters, we use different buffer sizes from a few packets to the nominal value which is $RTT \times C = 125$ packets. Each simulation is repeated 50 times and the results are averaged. In order to see the effect of flow desynchronization [6], different number of flows are used in the simulations, namely $5 - 100$ flows. The simulations are done by *ns-2* and the results are analyzed by MATLAB.



(a) (b)

Fig. 3. (a) Relative (Dos over no Dos) Goodput vs. buffer size for different number of flows $N = 5 - 125$ (b) Goodput vs. buffer size for $1Gbps$ bottleneck link capacity

IV. Simulation Results

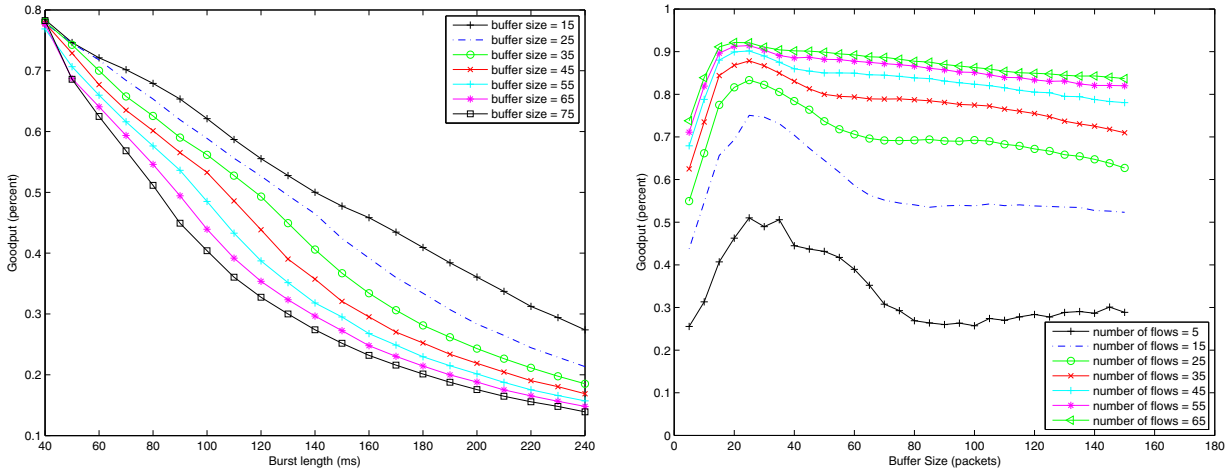
In this section, we present the results of our simulations for different sets of parameters. We try to justify the phenomena observed in the simulations based on how a TCP network operates.

A. Reno with a 100Mbps link

Fig. 2-(a) depicts the goodput of a network versus the buffer size for different number of flows. Here, we use TCP/Reno for congestion control. We can see that smaller buffers will result in better goodput and this is true for different number of flows. This shows that decreasing the buffer size prevents the attack from affecting all flows. That is, right after the attack, still some of the flows are not forced into time-out, hence keeping the goodput relatively high. The effect of decreasing the buffer size is valid up to a lower bound, below which the goodput again degrades regardless of the presence of the attack. As specified in Fig. 2-(a), the optimal buffer size is 15 packets here.

Fig. 2-(a) also shows that the goodput is higher when more flows are sharing the bottleneck. This is due to the fact that increasing the number of flows makes the traffic smoother and more difficult to synchronize, thus only a small portion of the flows are affected during the burst length of the attack.

In Fig. 2-(b), the averaged goodput versus buffer size curve has been plotted along with the variations of the goodput in the 50 runs. The vertical line in each point of the curve corresponds to the maximum and minimum of the goodput for that particular number of flows and buffer size. As seen in this figure, the variance is typically higher for $N = 5$ flows and for the *too small* buffer sizes (5 and 10 packets), where the goodput degrades. The variance shows that the goodput in these points (specified by the underlying parameters) has a higher randomness while in other points each run of the simulation is generating almost similar results. We have observed that when the buffer is below a certain threshold, packet drops happen continuously throughout the simulation. Here, flows will not have the chance to increase their congestion window significantly. This is inline with findings of papers studying tiny buffers [5], [6], and we believe the phenomenon happening here has nothing to do with the presence of the attack. To take a closer look at this, we repeat the simulations using the same parameters but without the DoS attack. Fig. 3-(a) illustrates the relative performance of the network in two cases: first, with the attack being active though the simulation and second with the attack deactivated. In other words, the attack is once removed and the performance for different number of flows and buffer sizes is calculated. Next, the attack is enabled and the performance is remeasured and the ratio of both is plotted. As seen in Fig. 3-(a), The ratio of the two performances approaches one even for the buffer sizes of 5 and 10 packets. In other words, although the performance of the attacked network degrades for buffer sizes less than 15, the un-attacked network also undergoes the same performance degradation.



(a) (b)

Fig. 4. (a) Goodput vs. the burst length l, for different buffer sizes (b) Goodput vs. buffer size, for different number of flows using TCP/RED

B. Bottleneck link Capacity of 1Gbps

Fig. 3-(b) depicts the goodput versus the buffer size for different number of flows for a bottleneck link of $1Gbps$ capacity. Here, the nominal buffer size, $RTT \times C$ is 1250 packets (we have kept $RTT = 100ms$). The same result of higher performance for lower buffer sizes and higher number of flows is also observable. We note that the enhancement from $N = 100$ flows to $N = 200$ flows is significant. This shows how the attack loses its severity with the increasing number of flows, which makes synchronization of flows harder.

C. Effect of changing the link capacities of the flows

In the previous experiments, the links from the TCP nodes to the router were all of equal capacity, namely $C = 10Mbps$. In the next simulation, the capacity of each node is randomly chosen from the interval $[0.5Mbps..99.5Mbps]$. The averaged performance is observed to be identical to that of the fixed link capacity (hence not shown). This is due to the fact that the bottleneck link (router - server link) is the dominant parameter in the performance of the system and as the aggregate goodput on the bottleneck link is being measured, the individual capacities through which the nodes access the router do not change the total performance.

D. Effect of changing the burst length (l)

Based on the specifications of the attack, a decrease in the buffer size also decreases the burst length with the same factor, as $l = \frac{B}{R_{\mathrm{DoS}-C}}$. This way, less number of flows drop packets. A question that arises here is what if the burst length is kept constant while decreasing the buffer size. For example, while the burst length for a configuration with a buffer size of 300 packets is $200ms$, for a tiny buffer of 30 packets, it should be $20ms$. The question is what would happen if the burst length is kept at $200ms$ and this attack is applied to a buffer of 20 packets. Fig. 4-(a) illustrates the effect of changing the burst length. In this figure, the goodput of the bottleneck link is plotted for different burst lengths and different buffer sizes. The maximum burst length is $250ms$ so that with an attack rate of $2 \times C$, the aggregate attack rate does not exceed $\frac{C}{2}Mbps$, where C is the average capacity of other TCP links. As seen in Fig. 4-(a), the network with small or tiny buffer size will undergo significant performance degradation with increasing the burst length l. This is expected because even with a small buffer, increasing the burst length is equivalent to increasing the total time during which the buffer is full, thus, causing more packet drops. This will in turn increase the number of flows that get affected by the attack and hence degrades the performance of the network, regardless of the buffer size. Interestingly, even for equal burst lengths, the network with a smaller buffer size still has a better performance. In other words, although the attack used here is not exactly the low-rate attack introduced in [3], with fixed burst length, a network with a 15 packet buffer shows a performance of 20% higher than a network with a buffer size of 75 packets. (Here, 50 flows are using the link).

E. Other TCP variants

Another interesting experiment is to see how different TCP variants will change the severity of the attack.

The performances of the network using TCP/Tahoe and TCP/SACK are almost identical to that of TCP/Reno (and hence not shown). On the other hand, randomly early detection (RED) [12] is expected to be able to significantly enhance the performance of the system. RED is a queue management algorithm. While in *drop-tail* strategy, when the queue becomes full, every arriving packet are dropped, in RED, arriving packets drop probabilistically with a probability that increases with the queue occupancy. So when the attack starts to fill the buffer up, simultaneously, with increasing the occupancy of the buffer, the malicious packets will drop with higher probability. As mentioned in [12] "RED's use of randomness breaks up synchronized processes that lead to lock-out phenomena".

Fig. 4-(b) confirms our expectations by showing how using RED will decrease the performance degradation caused by the attack in a network with large buffer sizes. Comparing Fig. 4-(b) and Fig. 2-(a) obviously clarifies the difference. Specially for higher number of flows, the attack can *not* degrade the performance of the network regardless of the buffer size. As the number of flows increases, the sensitivity of the performance to buffer size decreases and for example for $N = 65$ flows, the network almost keeps a goodput of 90% regardless of the buffer size. For lower number of flows, still, the pattern of goodput vs. buffer size is there and smaller buffers will result in higher performances. The too small buffer effect is also present here except for the fact that the minimum buffer size needed is observed to be higher than that of Fig. 2-(a) (15 packets).

V. Conclusion

This paper studies the security risk posed by DoS attacks on networks consisting of routers with tiny buffers. We show that although forcing packet drops (which is intended by the DoS attack) in routers with tiny buffers is easier, synchronizing the network flows, which deteriorates the overall performance of the system, is not as easy as it would be for routers with typical buffer sizes. This means that networks with tiny buffers are more robust to this class of attacks.

Furthermore, our simulations show that the performance also grows with number of flows, which is again due to the fact that synchronization will be more difficult. We also investigate the effect of increasing the burst length on the severity of the attack. Interestingly, although longer attack durations degrade the performance of networks with any buffer size, in equal situations, tiny buffer networks still have better performances than networks with large buffers. We conclude that, at least for the class of low-rate DoS attacks studied in this paper, reducing buffer sizes in Internet core routers makes the attack more difficult, and easier to detect.

VI. Acknowledgement

V. Havary-Nassab and A. Koulakezian wish to express their gratitude towards their supervisors, professor Shahrokh Valaee and professor Alberto Leon-Garcia, for supporting their studies and research at the University of Toronto.

References

[1] Y. Bouzida, F. Cuppens, and S. Gombault, "Detecting and reacting against distributed denial of service attacks," in *IEEE International Conference on Communications, (ICC)06*, vol. 5, Istanbul, June 2006, pp. 2394–2400.

[2] C. H. Lin, J. C. Liu, H. C. Huang, and T. C. Yang, "Using adaptive bandwidth allocation approach to defend DDos attacks," in *International Conference on Multimedia and Ubiquitous Engineering, (MUE'08).*, Busan, Apr. 2008, pp. 176–181.

[3] A. Kuzmanovic and E. W. Knightly, "Low-rate TCP-targeted denial of service attacks and counter strategies," *IEEE/ACM Transactions on Networking*, vol. 14, no. 4, pp. 683–696, Aug. 2006.

[4] M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control," RFC 2581 (Proposed Standard), Apr. 1999, updated by RFC 3390. [Online]. Available: http://www.ietf.org/rfc/rfc2581.txt

[5] N. Beheshti, Y. Ganjali, R. Rajaduray, D. Blumenthal, and N. McKeown, "Buffer sizing in all-optical packet switches," in *Optical Fiber Communication Conference, 2006 and the 2006 National Fiber Optic Engineers Conference. (OFC'06)* ., Mar. 2006.

[6] Y. Ganjali and N. McKeown, "Update on buffer sizing in internet routers," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 5, pp. 67–70, 2006.

[7] N. Beheshti, Y. Ganjali, A. Goel, and N. McKeown, "Obtaining high throughput in networks with tiny buffers," in *16th International Workshop on Quality of Service, (IWQoS'08)* ., Enschede, June 2008, pp. 65–69.

[8] D. Wischik and N. McKeown, "Part i: buffer sizes for core routers," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 3, pp. 75–78, 2005.

[9] G. Raina, D. Towsley, and D. Wischik, "Part ii: control theory for buffer sizing," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 3, pp. 79–82, 2005.

[10] G. Raina and D. Wischik, "Buffer sizes for large multiplexers: TCP queueing theory and instability analysis," in *Next Generation Internet Networks,*, Apr. 2005, pp. 173–180.

[11] N. Beheshti, Y. Ganjali, M. Ghobadi, N. McKeown, and G. Salmon, "Experimental study of router buffer sizing," in *8th ACM SIGCOMM conference on Internet measurement (IMC'08)* . New York, NY, USA: ACM, 2008, pp. 197–210.

[12] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet," RFC 2309 (Informational), Apr. 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2309.txt