# Performance Evaluation of Controlling High Bandwidth Flows by RED-PD

Osama Ahmed Bashir
Sudan University of Science and Technology

Md Asri Ngadi
Universiti Teknology Malaysia (UTM)

Yahia Abdalla Mohamed
Sudan University of Science and Technology

Mohamed Awad
Sudan University of Science and Technology

## ABSTRACT

This paper proposed to investigate and evaluate the performance of one of the algorithm used to provide fair bandwidth allocation to the flows. First In First Out (FIFO) queuing is simple but does not protect responsive flows from unresponsive one, flows that are sending more than their fair share. One of FIFO queuing is Random Early Detection which it can effectively avoid congestion at routers, but it also cannot provide fair bandwidth for the flows. On the other hand, per-flow scheduling mechanisms provide max-min fairness but are more complex, it requires keeping state for all flows going through the router; it's proved that high bandwidth flows at the time of congestion consume most of the bandwidth of the link, so this algorithm (RED-PD) is most candidates to provide fairness to the flows. Simulations with networks demonstrate that there are chances for RED-PD to enhance its work, by means of incorporating the test of unresponsive flows actively in response to changes in the packet drop rate.

## Keywords

RED, High bandwidth flows, Transport Protocols (TCP,UDP), Drop Probability, max-min fairness.

## 1. INTRODUCTION

The wide usage congestion control in the Internet Network is one of FIFO queuing in integration with TCP congestion control. FIFO queuing is simple to realize, and because of that it is appropriate to the environment of the Internet.

So FIFO is simple but there is no fairness between flows, high bandwidth flows always consume most of the bandwidth and leave other flows starves. These high bandwidth connections may be connections with small RTT times, connections not implementing TCP congestion control, or using greedy TCP connections. Per-flow scheduling provides fairness, but unfortunately it is difficult to implement because of the size of CPU cycles and memory size that it wants to keep information about the connections in the buffer. The rest of the paper is organized as follows: Section 3 explains methods used to identify high bandwidth flows. The simulation results are presented in Section 4. In Section 5, we evaluate the performance of RED-PD. Our conclusions are presented in Section 6.

## 2. Related works

Floyd and fall [3] introduce the mechanisms for identifying high-bandwidth flows from RED history by partition misbehaving and conformant flows in classes, so Floyd and fall did not present a complete solution and leave a room for development. Retail and Floyd [1] implement one of the mechanisms for identifying high-bandwidth flows. Which is identifying flows that are TCP-Friendly by comparing any entered flows with standard TCP flow. This flow named a reference flow, with constant RTT time equal to 40ms and assume that the packet size is not changed from flow to flow, so assuming a constant packet size, this method will work properly when all the connections have round-trip time ranging from 30-70ms, and also all the flows have the same packet size, unfortunately there is roughly no two flow with same round-trip time, even they are from the same source. We suggest develop RED-PD by incorporating the second method introduced by Floyd and Fall [3]. Which is identifying unresponsive flows, this type of identifying high-bandwidth flows has two partitions, the first partition is identifying high-bandwidth flows passively. Which is done by applying restriction over any unresponsive flow after monitoring it's characteristic. The other is done actively by applying restriction over the flow and monitored it to see is it responsive or not. The second part of the second method for identifying high bandwidth flows is suggested by Floyd and Fall [3], but it does not implement by Ratul [1]. Because of that the RED-PD is very sensitive to round-trip time, and it does not work well under any scenario with different round-trip time except the scenario with specific round-trip time, and so the RED-PD cannot identify any flow with round-trip time less than 40ms.

FRED [4] is using the FIFO scheduler and construct per-flow table at the router buffer only for those flows that have packet in the buffer, the probability of dropping a flow depends on the number of the packets that the flow has saved at the router buffer, FRED is close to max-min fairness algorithms because its concentrate on giving any flow the same fair share bandwidth this concentration leave it complicated to implement. Choke [5] is another method for approximately fair bandwidth sharing. The incoming packet is compared to a randomly selected packet from the queue, if they both belong to the same flow, tow packets are dropped, if they are not same, the incoming packet is discarded. The reason for this is that high-bandwidth connections have more packets in the router buffer. This method cannot work well under heavy load when there are too many flows. Choke [9] which it based on Choke and inherent all its features, the incoming packet is compared to a randomly selected packet from the queue if they both belong to the same flow, tow packets are dropped, if they are not same, the incoming packet is discarded. It's different from Choke on that it does not stop comparing until the match is failing or reach the upper limit number of matching, this method will harm the responsive flows like TCP flows, because it drops many packets from the same TCP flow.

There are many methods a router may use it to identify which high-bandwidth flows to regulate. For a router with active queue management such as RED, the arrival rates of high-bandwidth flows can be estimated from the recent packet drop

history at the router. Because the packet drop history of active queue management constitutes a random sampling of the arrival rate of the packets, a flow with much fraction of the dropped packets is likely to have a correspondingly much fraction of the arriving packets. Thus, for higher-bandwidth flows, flow's fraction of the dropped packets can be used to estimate that flow's fraction of the arriving packets.

## 3. RED-PD algorithm

The RED-PD algorithm is a lightweight mechanism combining the simplicity of FIFO with some of the protection of full max-min fair techniques [8]. RED-PD achieves this by keeping state for the high-bandwidth flows only, and it based on the first type of identification: Identifying flows that are not TCP-friendly and inherent all the problems of this method for identifying high bandwidth flows one of this problem is that RED-PD could not identify the high bandwidth flows when these flows has small round-trip time.

RED-PD is designed to work in an environment prevail by an end-to-end congestion control. RED-PD would work if there is a congestion at the router links, and it punishes the high-bandwidth flows. These high bandwidth connections may be connections with small RTT times, connections not implementing TCP congestion control, or using greedy TCP connections. Thus, in times of congestion RED-PD provides a protection to the low bandwidth connections.

RED-PD used to distinguish misbehaving connections and controls them by using discriminatory dropping packets, identification in RED-PD mechanism depend on RED packet drop history. All the packets that are dropped from the queue are equally Distributed sample of the traffic. If the connection of a large number of drops, it means that it's the most widely used of the line.

### 3.1 Identifying flows that are not TCP-friendly

Please use a 9-point Times Roman font, or other Roman font with serifs, as close as possible in appearance to Times Roman in which these guidelines have been set. The goal is to have a 9-point text, as you see here. Please use sans-serif or non-proportional fonts only for special purposes, such as distinguishing source code text. If Times Roman is not available, try the font named Computer Modern Roman. On a Macintosh, use the font named Times. Right margins should be justified, not ragged. A flow is TCP-friendly if its arrival rate does not exceed the arrival of a conformant TCP connection, the test of a flow if it's TCP-friendly or not assumes that TCP connection can be characterized by a congestion response of reducing its congestion window at least by half when there is indications of congestion and of increasing its congestion window by a constant rate of at most one packet per round-trip time otherwise.

For routers with attached links with large propagation delays, the TCP-friendly test of equation (1) gives a useful tool for identifying flows which are not TCP-friendly. For routers with attached links of smaller propagation delay, the TCP-friendly test of equation (1) is less likely to identify any unfriendly flows. Sometimes conformant TCP flow could receive a disproportionate share of the link bandwidth because it has a significantly smaller round trip time than other competing TCP flows.

$$T <= \frac{1.5}{R * \sqrt{P}} \qquad (1)$$

T is the maximum sending rate, R the minimum round-trip time R and could be set to twice the one-way propagation delay of the attached link, p is the drop rate of the link; this would limit the appropriateness of this test to those routers where the propagation delay of the attached link is likely to be a significant fraction of the end-to-end delay of a connection's path.

### 3.2 Identifying unresponsive flows

A more general test would be simply to verify that a high bandwidth flow was responsive, so its arrival rate decreases appropriately in response to an increased packet drop rate, equation (1) shows that for a TCP flow with persistent demand, if the long-term packet drop rate of the connection increases by a factor of x, then the arrival rate from the source should decrease by a factor of $\sqrt{x}$.

## 4. Simulation

Please use a 9-point Times Roman font, or other Roman font with serifs, as close as possible in appearance to Times Roman in which these guidelines have been set. The goal is to have a 9-point text, as you see here. Please use sans-serif or non-proportional fonts only for special purposes, such as distinguishing source code text. If Times Roman is not available, try the font named Computer Modern Roman. On a Macintosh, use the font named Times. Right margins should be justified, not ragged. The simulation was used to investigate the performance of RED-PD, in subsection 4.1 we analyze the effect of round-trip time of the reference TCP flow on identifying high-bandwidth flows, and in subsection 4.2 we demonstrate the effects of reference time RTT R in window based protocol like TCP protocol, in subsection 4.3 we explain the effects of RTT R of reference flow on connections not based on TCP protocol, and in subsection 4.4 we try to investigate the reason for underestimate and overestimate.

### 4.1 Effect of RTT R of the reference flow

The title (Helvetica 18-point bold), authors' names (Helvetica 12-point) and affiliations (Helvetica 10-point) run across the full width of the page – one column wide. We also recommend e-mail address (Helvetica 12-point). See the top of this page for three addresses. If only one address is needed, center all address text. For two addresses, use two centered tabs, and so on. For three authors, you may have to improvise. The simulation illustrates how the choice of RED-PD's configured RTT parameter R affects both the identification of flows for monitoring and the bandwidth received by monitored flows. Each column in Figure 1 represents a different simulation, with a different value for R, ranging from 10 ms to 170ms. In each simulation 14 TCP connections were started, two with RTTs of 40 ms, 80 ms and 120 ms, and the remaining connections with RTTs of 160ms. The average bandwidth received by TCP flows with RTT times from 40-120ms presented in Figure 1 , the horizontal lines in Figure 1 show the bandwidth for each traffic type with RED.

For the simulations with R less than 40 ms, RED-PD rarely identifies any flows, However, for the simulations with an

RTT of 40 ms and more, the TCP flows with short RTT and 40-ms RTT starts to be identified and dropped preferentially.

## 4.2 Effects of reference round-trip time R in window based protocol

For pages other than the first page, start at the top of the page, and continue in double-column format. The two columns on the last page should be as close to equal length as possible. In this subsection we demonstrate the effect of round-trip time R of reference TCP flow on connections based on the TCP protocol. Equation (2) demonstrated below define the time required to keep history for high-bandwidth flows if the round- trip time r, for the flow is smaller than the reference time RTT R, this means that for this flow we underestimate the sending rate. Because the congestion epoch of this flow is less than the congestion epoch of reference flow (RTT R). So it has more than one congestion epoch in one congestion epoch of reference flow. As an example if the monitored flow has tow congestion epoch in one congestion epoch of reference flow, this means that for this flow to identify as high bandwidth flow, if it drop at least six packets, and for other ordinary flows to identify as high-bandwidth flows if it has just three drops. So RED-PD will miss the high-bandwidth flow if it has small RTT r, in general the RED-PD was very sensitive to round-trip time. This is explained by the simulation presented in figure (2). For the simulations with R less than 70 ms, at the left side of graph RED-PD rarely identifies any flows, we call this underestimate and for the connections with big RTT there is an overestimate.

$$CL(r,p) = \frac{R}{\sqrt{1.5p}} \quad \text{Second} \qquad (2)$$

## 4.3 The effects of reference RTT R on connections not based on the TCP protocol

All the applications based on TCP protocol is a window based protocol. In contrast to that is the application based on UDP protocol which is not window based protocol. And it's sending rate depend on rate because it's designed for real time application. So what are the effects of the reference RTT R on the reference TCP flow of this type of flows. By default there is no effect of RTT R on this type of connections because it's sending on constant rate (CBR), or sending on a variable rate (VBR) and it has no window at all. But it has a negative impact on CPU cycles of the router, because the frequency of identifying high-bandwidth flows depend on the time RTT R. And because the UDP application does not respond to any number of dropped packets from its sending rate under any circumstance, because of that the router spend a lot of time in defining high-bandwidth flows, and because these flows does not change its sending rate and the router does changing its policy for this connection and therefore the router waste its time in defining and redefining flows that does not respond. But we need the time RTT R for defining new high-bandwidth flows that start it's sending after the last time we define high-bandwidth flows, so for all applications that does not depend on window the time RTT R is just necessary for the algorithm to define new high-bandwidth flows, or to decide to remove it from the list of high-bandwidth flows.

## 4.4 The reason of underestimate and overestimate

The reason to underestimate the high-bandwidth flows with small RTT is that when the RTT r of the monitored flow is small the congestion epoch for this flow is also small so there is more than one congestion epoch in one congestion epoch of the reference TCP flow of RED-PD, so there is more than one round-trip time packets in one round-trip time of reference flow, and all the dropped packets of this one round-trip time packets will be treated by the drop-list of the RED-PD as dropped from one part of the list for the flow under check, so RED-PD will not identify this flow as high-bandwidth flow, because RED-PD count all those dropped packets as dropped from one round-trip time of the flow under check, so the flow will escape from monitoring and identifying as high-bandwidth flow, when this flow is not identified as high-bandwidth flow it will steal more bandwidth from other connections in the channel.

The other type of flows, the flow which has big round-trip time considered as high-bandwidth flows, because its RTT is high and therefore the congestion epoch is big so there is less than one congestion epoch in one congestion epoch of reference flow and therefore there is less than one round-trip time packets in one round-trip time of reference flow, therefore any number of dropped packets of one round trip-time of the flow under check will be treated by RED-PD drop-list as many drops from many round-trip time because it scattered in many congestion epoch of reference flow, so for this type of flows with high RTT r the RED-PD will overestimate the bandwidth, so it will be harm by more dropping, as example when there are three or more packets

dropped from the drop-list and this drop-list is contain less than one round-trip time packets from the flow under check this will cause the algorithm to define this flow as high-bandwidth flow.

If the round-trip time r for the flow is big than the reference RTT R, this means that for this connection we overestimate the sending rate, as example if it has one congestion epoch in three congestion epochs of reference flow this means that for this flow any number of close packets dropped will scattered in many part of the drop-list and so this flow identified by the RED-PD as flow with high-bandwidth, in contrast to the fact it's not high bandwidth flow.

To demonstrate also that RED-PD is very sensitive for round-trip time we repeat the same experiment done by original RED-PD by decreasing and increasing in figure (2) and figure (3) respectively the round-trip times of flows, and observe that there is underestimate for flows with small round-trip time figure(2)., and also there is overestimate for the flows with big round-trip time figure(3).

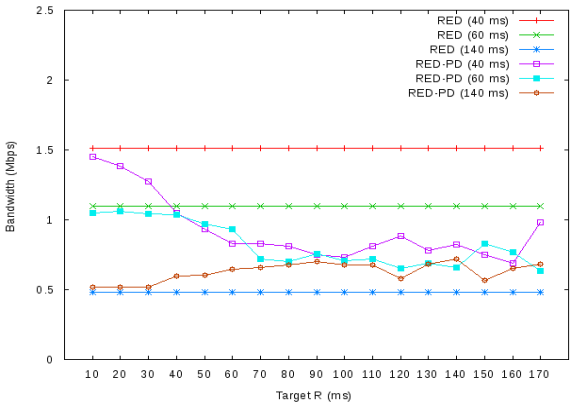


**Fig 1: demonstrate the overestimate and underestimate at the tow edges of the graph after decreasing the round-trip time for the flows**

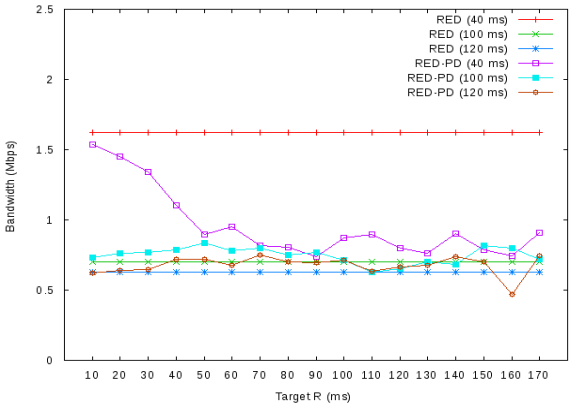


**Fig 2: demonstrate the overestimate and underestimate at the tow edges of the graph after increasing the round-trip time for the flows**

## 5. Evaluation

We use simulation to evaluate RED-PD, we study RED-PD's to demonstrate that RED-PD is sensitive to reference round-trip time R, and illustrate that the frequent time used to identify and regulate high-bandwidth flows is also a tradeoff value, we carried out the simulations using the NS network simulator [7]. The bandwidth of the congested link was 10 Mbps, RED-PD's target RTT R was 40 ms, the packet size was 1000 bytes, and RED was running in packet mode. The Selective Acknowledgement (SACK) [6] version of TCP was used, flows were started at a random time within the first 10 seconds, and the results, were not taken before 20 seconds into the simulation.

### 5.1 Reasons of fluctuation

We change the time of detecting or time in which we specify new high-bandwidth flows, or take one of high-bandwidth flows outside or inside the list of the high-bandwidth flows, those changes include the time that after it we modify the parameters for the flows that was changing its sending rate, we observed that there are fluctuation and degradation in the performance of the RED-ED algorithm, despite the assumption that high-bandwidth flows continue live for long periods of time.

High-bandwidth flows continue for long time connected and consuming bandwidth, the reason for fluctuate is the queue of the router itself, or the number of packets that buffered at each process of detecting, any detection of high-bandwidth flows are determined by the number of packets in the queue and number of
Recurrence time between detecting, in the drop-list of RED-PD the number of packets is small but it seems that the performance is better, but they have been compensated for the number of packets in the drop-list by a number of times for detecting high-bandwidth flows, so when we hold the same experiment by the same parameters, but after decreasing the time of detecting high-bandwidth flows, we observed that there is great fluctuation in the performance of the RED-PD algorithm.

But when we increase the number of packets contained in the drop-list, which we use to detect high-bandwidth from it at every process of detecting, so it will be recognized fully and effectively high-bandwidth at every process of detecting.

For figure (3) which it explained the simulation with many CBR connections, flow1 is sending at 0.1Mbps, flow2 at 0.5Mbps and every subsequent flow is sending at a rate 0.5Mbps more than the one previous to it, and the propagation delay for all the connections is 28ms and drop list size is 50 packets, separate lines in figure (3) show the bandwidth received by the 11 CBR flows with RED and with RED-PD while a third line shows each flow's max-min fair share, and for Figure (4): show a mix of TCP and CBR connections. Flows 1-9 are TCP connections with RTTs of 30-70ms. Flow 10, 11 and 12 are CBR flows with sending rates of 5, 3and 1 Mbps, and the propagation delay ranging between 30ms to 70ms and the drop-list of size 50 packets, we do not want to show the efficiency of RED-PD from this graph, but we want to show you that RED-PD is stable in controlling high-bandwidth flows but when we change the time frequency of detecting as we show in figure (3) (figure (4) for mix flows TCP and UDP) with same all the parameters of the network of figure (3) (and figure (4)) we see that there is fluctuation in the efficiency of RED-PD performance, the reason of this fluctuation is the number of packets in the drop-list which is used by RED-PD to determine which of the flows is high-bandwidth flows, so when this number of packets is small, and time frequency of detecting is also small, this will cause the fluctuation to happen, by other words this happen due to determine new high-bandwidth flow and after short time determine the same flow as normal flows and so on, this will lead at the end to fluctuation.

Figure (5) and figure (6) show network simulation scenario as in figure (3) and figure (4) respectively but we decrease the time frequency to detect high-bandwidth flows, and increase the number of packets in the drop-list which it used by RED-PD to determine high-bandwidth flows from the router queue, so as we explain there is stability in the RED-PD efficiency comparing to the efficiency of RED-PD in figure (3) and figure (4), although we increase the frequency time for detecting high bandwidth flows.

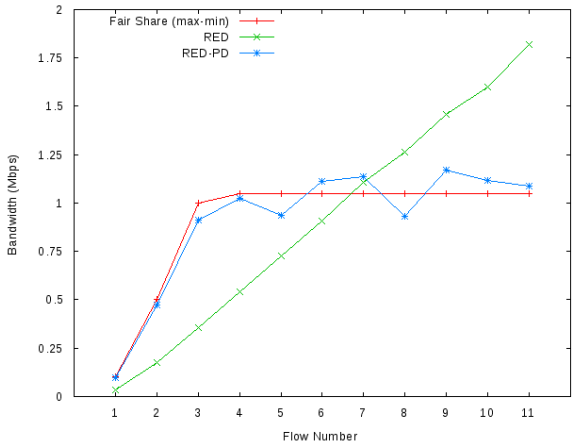


**Fig 3: Simulation show CBR flows. The sending rate are 0.1Mbps, 0.5 Mbps for the first and second flows and for the remaining flows the sending rate is 0.5 Mbps more than the former one, and the propagation delay for all the connections is 28ms and drop list of size 50 packets.**

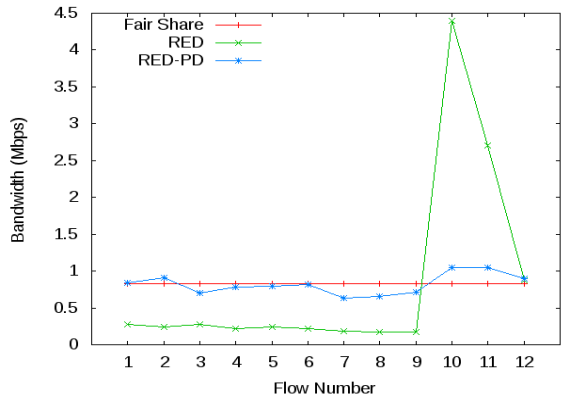


**Fig 4: In this simulation there are 12 TCP and CBR flows. The flow from 1 to 9 are TCP flows with round trip time of 30-70ms. Flows numbered 10, 11 and 12 are CBR flows with rates of 5, 3and 1 Mbps, and the propagation delay ranging between 30ms to 70ms and drop list of size 50 packets.**

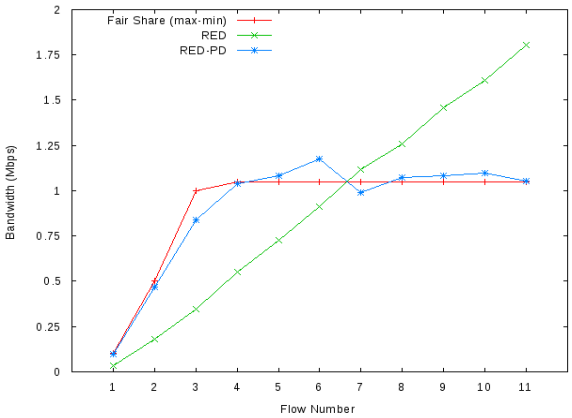


**Fig 5: Simulation show CBR flows. The sending rate are 0.1Mbps, 0.5 Mbps for the first and second flows and for the remaining flows the sending rate is 0.5 Mbps more than the former one, and the propagation delay for all the connections is 28ms and drop list of size 50 packets.**

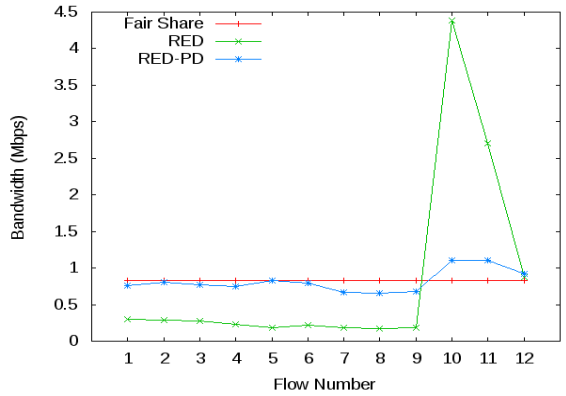


**Fig 6: In this simulation there are 12 TCP and CBR flows. The flow from 1 to 9 are TCP flows with round trip time of 30-70ms. Flows numbered 10, 11 and 12 are CBR flows with rates of 5, 3and 1 Mbps, and the propagation delay ranging between 30ms to 70ms and drop list of size 50 packets.**

## 6. Conclusion

Its proved that most of the bandwidth of the link consumed by a few connections, so it's better for routers to monitor those a few connections and control and regulate its bandwidth, in this paper we evaluate the performance of RED-PD for Controlling High Bandwidth Flows work and the capability of implementation by means of reducing the frequency of detecting and controlling high bandwidth flows which is proved that also a few number of connection not consume much of the bandwidth for the link but also continue for long time in consuming the bandwidth of that link and give chance to routers to regulate them perfectly, and we would like to mention that RED-PD is sensitive to round-trip time, because it use only one type of identifying (TCP Friendly test) which is depend on round-trip time of reference TCP flow.

## 7. REFERENCES

[1] Ratul Mahajan and Sally Floyd Controlling High Bandwidth Flows at the Congested Router in ACIRI, November 20, 2000.

[2] S. Floyd and V. Jacobson, Random early detection gateways for congestion avoidance, IEEE-ACM Transactions on Networking, pp. 397-413, August 1993.

[3] S. Floyd and K. Fall, Promoting the use of end-to-end congestion control in the internet, IEEE-ACM Transactions on Networking, pp. 458-472, August 1999.

[4] Dong Lin and Robert Morris. Dynamics of Random Early Detection. In ACM SIGCOMM, September 1997.

[5] Rong Pan, Balaji Prabhakar and Konstantinos Psounis. Choke, A Stateless Active Queue Management Scheme for Approximating Fair Bandwidth Allocation in IEEE INFOCOMM, March 2000.

[6] Matt Mathis, Jamshid Mahdavi, Sally Floyd, and Allyn Romanow. TCP Selective Acknowledgement Options. RFC 2018, April 1996.

[7] S. Floyd, NS network simulator, www.isi.edu/nsnam.

[8] Hahne E., and Gallager, R., Round Robin Scheduling for Fair Flow Control in Data Communications Networks, IEEE International Conference on Communications, June, 1986.

[9] Addisu Eshete and Yuming Jiang. Generalizing the CHOKe flow protection, Computer Networks, January 2013.