# Evaluate the Performance of RED and DropTail Algorithm in NS2

**Huy-Khoi Do[1], Thi-Thu-Hang Nguyen[2], Anh-Tuan Nguyen[3], Thanh-Nam Pham[4]**

[1]University of Information and Communication Technology (ICTU), Quyet Thang, Thai Nguyen, Vietnam
* To whom correspondence shoud be addressed: dhkhoi[at]ictu.edu.vn

**Abstract:***Nowadays, with the increasing number of computers participating in the Internet, network congestion can occur when the packet traffic reaching the network line is beyond the processing capacity of the network. This leads to a waste network resources and causes many negative effects, affecting the service quality of the network. Therefore, the congestion control issue was put in place to control the operation of network components appropriately. The paper uses NS2 to evaluate congestion control mechanism by managing queues in buffers such as RED, DropTail, ... based on the evaluation of parameters of throughput, queue size, transmission window size.*

**Keywords:** RED, DropTail, NS2, congestion control, queue management

## 1. Introduction

In packet-switched networks, different packet flows often have to share the path along the way to the destination station. In order to ensure the most efficient distribution of bandwidth for the balanced and efficient flows, it is necessary to have appropriate service mechanisms at the network nodes, especially at gateways or routers, where there are often a lot of various data flows passing.

In normal condition, when no congestion occurs, information packets will be sent as soon as they arrive. In case of congestion, if the service quality assurance method is not applied, extended congestion time may result in packet loss, affecting service quality. In some cases, extended and widespread congestion in the network can result in the loss of many packets that severely affect service quality.

Buffer management methods are one of the quality service delivery mechanisms. Buffer management decides buffer allocation and eliminates incoming packets according to a predetermined policy. Therefore, there are many algorithms given in queue management techniques such as Random Early Detection (RED), Blue, Flow Random Early Detection (FRED), Tail Removal Method (DropTail), ...

## 2. Queue Management Algorithms

### 2.1. DropTail

DropTail is a traditional technique for managing the network node's queue length by setting the maximum queue length for each queue, accepting incoming packets until the maximum length is reached. Incoming packets will be discarded until the queue size decreases [1].

When the queue occupancy is less than the maximum value, the packets coming from the traffic flow are still accepted and are queued for processing. When the queue occupancy exceeds the allowed maximum, all subsequent packets are discarded until the queue occupancy decreases.

* When the queue is full, there are two ways to eliminate incoming packets:

- Eliminate in the queue: if the queue is full and there are packets coming to the queue, the discard will happen randomly inside the queue. A new package will arrive in the queue.

- Remove queue headers: if the queue is full but there are still packets coming to the queue, the router will discard the packet at the beginning of the queue.

Both of these methods solve the lock out problem but still do not solve the problem of full queue. In the current Internet, packet removal is like a congestion notification mechanism to destination nodes. The solution to the problem of full queue is that the router eliminates packets before the queue begins to fill up, so routers can respond to congestion before the buffer overflows.

### 2.2. RED

The idea is not to wait until the buffer is full to detect congestion, but start to look for congestion before the buffer overflows. Signs of congestion can still be through packet discard, but also be through packet marking without having to discard them. RED buffer management algorithm has following objectives:

- Priority: reserved for short bursts of data belonging to sensitive delay type, but not allowing a big increase in the average queue size. By using some "low past" filtering for queue size, the purpose is to detect code bottlenecks that are long enough.
- Ports of DropTail and Random Drop have priority for cluster traffic. Indeed, in such buffers, the more traffic that a cluster type connection has, the more likely the queue will overflow at the time of its connection.
- Avoid synchronization: In the DropTail type buffer, many connections may receive congestion signs at the same time leading to unwanted fluctuations in throughput. These oscillations can be for lower average flux and high jitter. In order to avoid synchronization (which is to prevent the connection session from receiving congested signals at the same time), the congestion signals are randomly selected.
- Control the average queue size.

To achieve these goals, RED monitors the average queue size avg, and checks if it is between a minth and maxth or not. If so, an incoming packet will be discarded or marked with probability p=p(avg), this probability is a row increasing with the average packet size. All packets until avg exceeds maxth will be marked / discarded.

Probability p (avg) is selected as follows. When the average queue size varies between minth and maxth, a probability of pb varies linearly between 0 and certain maxp values [2].

$$p_b(avg) = max_p \frac{avg - min_{th}}{max_{th} - min_{th}}$$

This probability is used as p (avg) if at the time of the arrival of the previous packet avg>= $min_{th}$. Otherwise p(avg)=p(avg) /(1+ p(avg)).
This average queue size is monitored as follows. Initially, the avg parameter is set to be zero (=0). Then, for each incoming packet, avg sets a new value [2]:

$$(1 - w_q)avg + w_q q$$

*Note: q is the actual queue size and wq is a constant.*

## 3. Simulation and Comparision

### 3.1. Simulation Settings

NS2 which is an open source software available on Ubuntu, version ns-allinone 2.34 and some tools supporting the analysis and display of simulation results such as perl, gnuplot, ...are used to evaluate and compare evaluation results with the performance of RED and DropTail buffer management mechanisms.

To analyze the simulation results and evaluate the performance of RED and DropTail, we need to use the same network configuration to compare the two mechanisms. The simulation parameters as described in Table 1:

**Table 1:** Network configuration parameters

| Simulation parameters | Value |
|---|---|
| Version NS | Ns-allinone-2.34 |
| Evaluation methods | RED, DropTail |
| Number of nodes joining simulation process | 11 |
| Queue size | 100 |
| Form of transmission | TCP/Reno |
| Time of simulation | 50s |

This paper will simulate an 11-node network simulation and conduct the performance evaluation analysis of the mechanisms by comparing results based on a number of performance metrics such as queue size, throughput, and size of transmission window.

Figure 1 shows the topology of the network simulation:

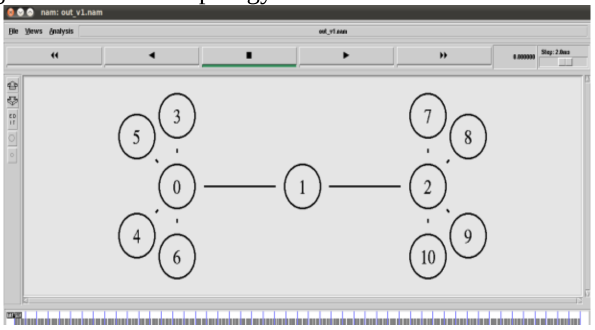

**Figure 1:** Network Topology

### 3.2. Evaluation Results ang Analysis

**The analysis of Drop Tail**
*** Queue Size *of DropTail***

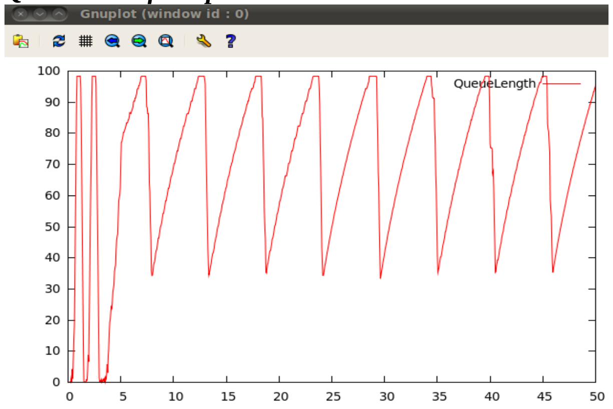

**Figure 2:** Queue size

As shown in figure 3 above, the largest queue size is 100; and simulation using TCP / Reno object should follow the principle of congestion control of Reno. When the throughput exceeds the queue size, it goes into the congestion control process to reduce ½ of the queue window size. After that, it goes into the congestion control process, increasing the size of the transmission window to 1 after each transmission. The process repeats until the simulation time is over.

*** Average throughput of TCP flows - according to DropTail**

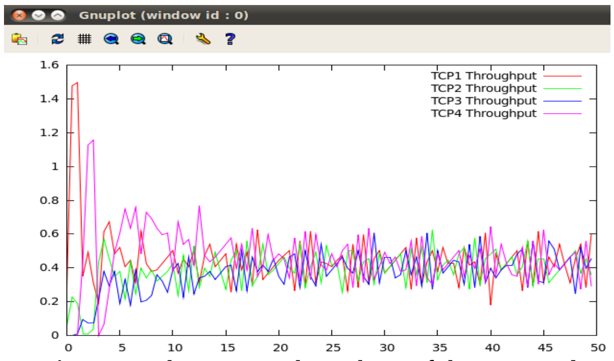

**Figure 3:** The average throughput of the DropTail

It can be seen in the figure 4 that on the first stage, there is only TCP1 flow in the transmission line, so tcp1 performs slow-start phase of the number of packets and transmission line in exponential, the amount of data increases very quickly in the transmission environment. When TCP2 joins in the transmission process, the number of packets of the TCP stream is still very high in the queue and in the transmission line, so the number of successfully-transmitted tcp2 is very low. However, because it works under the tcp mechanism, the number of tcp2 packets reaches a stable level. Then tcp3 and tcp4 start to transmit leading to a decrease in the throughput of tcp1 and tcp2. At the point when t is 5 seconds or more, since 4 flows work with the same mechanism, it reaches Load balancing, and 3 flows achieve a stable throughput of about 0.4 Mbytes / sec.
Average achieved throughput of flows:
Flow 1: Avg throughput = 0.4512 MBytes / sec
Flow 2: Avg throughput = 0.3643 MByte / sec
Flow 3: Avg throughput = 0.3893 MByte / sec

Flow 4: Avg throughput = 0.4638 MBytes / sec

With the results of the average throughput, it can be said that due to flowing first, tcp1 reaches higher average throughput, while tcp2, tcp3 and tcp4 streams go later and reach lower average throughput.

*** Congestion control –DropTail**

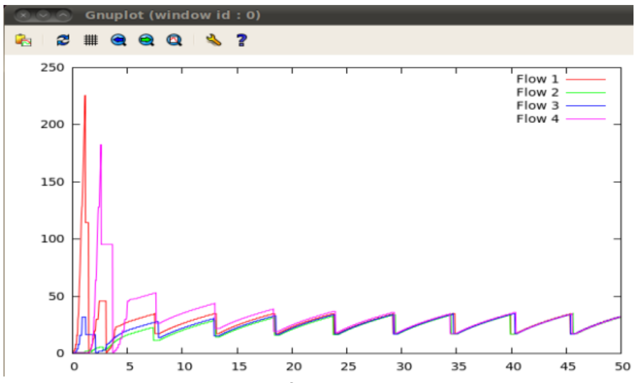

**Figure 4:** Size of the DropTail window

It can be interpreted from figure 5 that TCP flows congestion control mechanism by Reno, so the first stage size of tcp1 flow increases rapidly. However, when entering the congestion control stage, the window size of all 4 flows decreases. After about 10 seconds, all 4 flows reach the equal state, and the transmission window size stays unchanged for the remaining time.

**The analysis of RED**

The average queue size and the actual queue size of the RED queue as shown in the following figure 5:

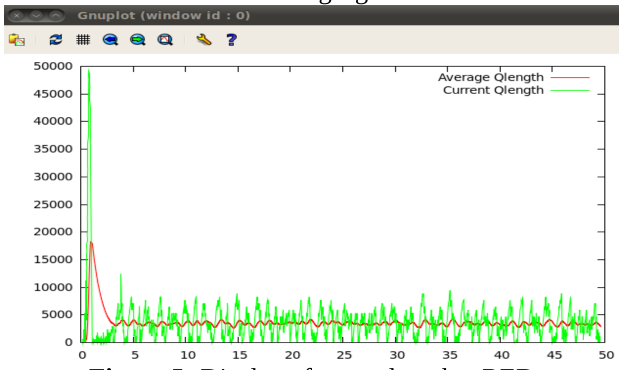

**Figure 5:** Display of queue length – RED

With the results showing the actual flux and the average flux achieved of the Red protocol, with simulation setting minth = 0; maxth = 7; The control of the number of packets in the traffic to avoid congestion done by processing incoming packets, red calculates the number of packets in the network compared to the transport capacity of the network, and it will cancel a the number of packets randomly follows a probability p (0 <= p <= 1). On the first stage, because the information flow transmitted under TCP protocol performs the slow-star phase, the network throughput reaches a high level, and then stabilizes within the established threshold [3].

*** Average throughput of TCP flows - according to RED**

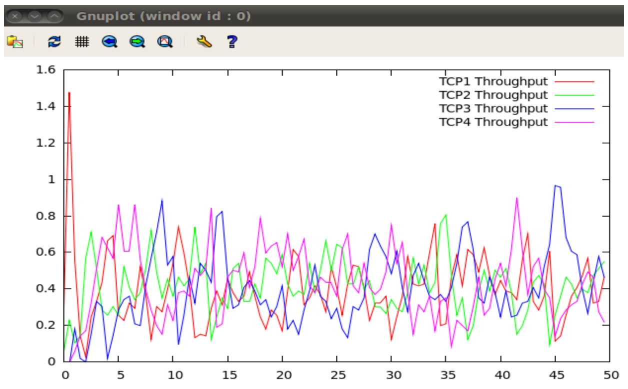

**Figure 6:** Average throughput – RED

According to the result in the graph, when the tcp flows perform with the RED queue, the data is estimated to be calculated with actual throughput and some packets of the stream are randomly removed, so the flows reach an unstable throughput [2] deepending on network calculation. However, the average throughput of the networks still balances among the flows.

**Average achieved throughput of flows:**

Flow 1: Avg throughput = 0.3966 MBytes / sec
Flow 2: Avg throughput = 0.4 MByte / sec
Flow 3: Avg throughput = 0.4118 MByte / sec
Flow 4: Avg throughput = 0.4265 MBytes / sec

*** Congestion control – RED**

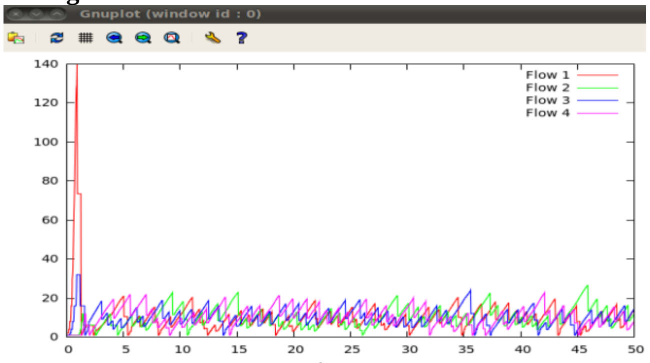

**Figure 7:** Size of the RED window

As the simulation results shown above, on the first stage tcp1 flow is predominant thanks to its earliest transmission, but then when packet loss occurs - if there is a possibility of congestion control, Red will turn into congestion control mode, and it randomly estimates a number of packets before entering the queue, and after about 2s, all 4 tcp streams of the transmission window size or the number of packets transmitted by the 4 streams are equal.

## 3.3. Comparison

We can see that, in the first seconds when there are not many streams, the entire data of RED and DropTail are successfully transmitted and because the first TCP1 stream goes first, the maximum throughput [3] is achieved.

With the DropTail method, from 5s to 15s due to the participation of continuous flow of data into the network, the number of packets going into the queue is big; if the maximum size of the queue is exceeded, the incoming packet will be lost, and the number of packets is greater than the network node's processing capacity, congestion occurs. But since all 4 threads operate under the TCP mechanism, from 15s to 50s the streams reach load balancing with the number of packets entering the queue ranging from 15 to 35 packets and reaching an average throughput of 0.4 Mbytes / sec. [4].

With the RED method, due to the early detection mechanism, when the number of packets in the path increases, the congestion control mechanism is implemented, resulting in the loss of packets in the queue leading to the traffic flow. Therefore, the number of packets in the queue decreases and only fluctuates around 5 to 20 packets [5].

## 4. Conclusion

With the above results, the two management mechanisms have the same throughput but the DropTail method is better since the size of the queue window in the network is higher and more stable, which help restore the transmission path better. The packet transmission window size of DropTail is also higher than that of RED because RED's mechanism is early congestion detection, which makes the congestion control mechanism to be implemented earlier resulting in a bigger number of lost packets.

## References

[1] Shubhangi Rastogi and Samir Srivastava, “Comparison Analysis of Different Queuing Mechanisms DropTail, RED and NLRED in Dumb-bell Topology”, *International Journal of Advanced Research in Computer and Communication Engineering*, 3, (4), pp. 6286 – 6288, 2014.

[2] Eitan Altman and Tania Jiménez, *NS Simulator for beginners*, Univ. de Los Andes, Mérida, Venezuela and ESSI, Sophia-Antipolis, France, 2003.

[3] Vũ Xuân Bảo, "Assessing the effectiveness of QoS for multimedia communication by WRED queue management strategy", University of Technology - Vietnam National University, Hanoi, 2011.

[4] Babek Abbasov and Serdar Korukoğlu, (2009), “An Active Queue Management Algorithm for Reducing packet loss rate”, *Mathematical and computational Application*, 14, (1), 65-72.

[5] Ganesh Patil, Sally McClean, Gaurav Raina, “Drop Tail and RED queue management with small buffers: stability and hopf bifurcation”, *Ictact journal on communication technology*, 2, (2), pp. 339-344, 2011.

## Author Profile

**Huy-Khoi Do** was born in Vietnam. He obtained his master degreein University of Engineering and Technology, Vietnam National University. His research interests include networks and data communication, communication sytems, images and speech processing, telecommunication.

**Thi-Thu-Hang Nguyen**was born in Vietnam. She obtained his masterdegree in Hanoi University of Engineering and Technology. Her research interests include networks and data communication, and wireless access network.

**Anh-Tuan Nguyen**was born in Vietnam. He obtained his master degreein Le Quy Don Technical University, Vietnam National University. His research interests include networks and data communication, communication sytems.

**Thanh-Nam Pham** received the Ph.D. degree in Information Engineering and Computer Science from Feng Chia University, Taiwan, in 2018. He is currently with the Department of Electronics and Communications at Thai Nguyen University of Information and Communications Technology, Vietnam. His research interests include Internet of Things Technologies and applications.