BULLETIN OF ENGINEERING SCIENCE, TECHNOLOGY AND INDUSTRY
https://bestijournal.org | PT. Radja Intercontinental Publishing
E-ISSN: 3025-5821

ANALYSIS OF RANDOM EARLY DETECTION AND HIERARCHICAL TOKEN BUCKET METHOD ON LOCAL AREA NETWORK WIRELESS

Shaidul Maula[1], Anwar[2], Jamilah[3], Mark G. Maffett[4]
Department of Information and Computer Technology, Politeknik Negeri Lhokseumawe[1.2.3]
University of Chicago[4]
*Correspondence Email: shaidulmaula@live.com

Abstract

One of the problems that internet users often face is slow internet access. This problem can occur because there is no bandwidth management system in use. The Random Early Detection and Hierarchical Token Bucket methods are good methods. However, it is not yet known which is the best between the two. Research needs to be carried out to determine the comparison of network quality using these two methods. This research aims to determine the quality of networks that apply the Random Early Detection and Hierarchical Token Bucket methods whether they are given noise injection or not. This research data was collected by testing the wireless local area network. The collected data is analyzed using QoS. The results show that the implementation of the Hierarchical Token Bucket method is of better quality, it can be seen with a throughtput value of 879128 bps, a delay value of 8 ms, a jitter value of 11,041 ms and a packet loss value of 0%. On the other hand, the Hierarchical Token Bucket method was given Noise Injection with a throughput of 582716 bps, a delay value of 15,427 ms, a jitter value of 21,924 ms and a packet loss value of 2.8%. Meanwhile, the application of the Random Early Detection method obtains quality with a throughput value of 816,287 bps, a delay value of 8,673 ms, a jitter value of 12,591 ms and a packet loss value of 0%. On the other hand, the Random Early Detection method is given Noise Injection with a throughput value of 704778 bps, a delay value of 11,154 ms, a jitter value of 15,565 ms and a packet loss value of 2.14%.

Keywords: QoS, Random Early Detection, Hierarchical Token Bucket.

INTRODUCTION

The development of computer technology is very rapid and the development of computers is also accompanied by the development of networks. In this case the network in question is the internet network. As the number of internet users increases, network traffic becomes increasingly complex and requires bandwidth management [1]. However, the high price of Bandwidth causes restrictions on the amount of Bandwidth provided by operators. With the high level of internet demand, of course this is a problem for users. Information becomes difficult to obtain due to limited bandwidth which makes it difficult for users to access internet sites [2]. Several packet queuing methods have been implemented to improve network quality such as the PCQ, SFQ, HTB, RED and FIFO methods. Regarding the HTB method, research has been carried out on the application of the Hierarchical Token Bucket method on internet networks by carrying out a scenario, namely limiting bandwidth for each user, 512Kbit for user 1 and 256Kbit for user 2. The results of this research show that the bandwidth capacity is small which causes a decrease in network performance. for users, even though the number of packets to be sent decreases or delivery times experience a slowdown, there is a large increase in data validity as evidenced by the decrease in the number of lost packets which reaches 80% or more [1]. Apart from that, there is research that aims to improve the quality of the network in the Cibuntu-Kuningan Tourism Village using the RED and PCQ methods. The results of this research show that the RED method is better used in signal conditions in Cibuntu Village which are characterized by an average ping time <163.65, while the PCQ method only gets an average ping time <163.85 [3].

A lot of information has been obtained from various references, the Random Early Detection and Hierarchical Token Bucket methods are good methods. However, it is not yet known which is the best between the two. Research needs to be carried out to determine the comparison of network quality using these two methods. The Random Early Detection method is used as a packet queue which tries to avoid network congestion by controlling the average value of the queue [4]. On the other hand, the Hierarchical Token Bucket method provides traffic restriction facilities at each level and classification so that bandwidth that is not used by high levels can be used by lower levels [5]. Based on this research, research will be carried out to find out the comparison of network quality using the RED method with HTB.

METHOD

The research was carried out using system design and system testing methods. The steps taken are: first, determine the needs analysis, second, determine the overall network architecture, third, determine the system block diagram and finally determine the system block diagram to carry out system testing.

A. Needs Analysis

1. Mikrotik

   MikroTik Router is a Linux based operating system which is intended as a router. Mikrotik has 2 products which can be seen as follows.

   a. Mikrotik RouterOS is an operating system and software that can be used to turn a computer into a reliable network router, including various features created for IP networks and wireless networks, suitable for use by ISPs and hotspot providers.

   b. Mikrotik routerboard is an embedded router product from Mikrotik. A routerboard is like an integrated mini PC because the processor, RAM, ROM and flash memory are embedded in one board. Routerboard uses RouterOS OS which functions as a network router, bandwidth management, proxy server, DHCP, DNS server and can also function as a hotspot server [6].

B. Wireshark

Wireshark is one of the many Network Analyzer tools that is widely used by network administrators to analyze network performance, including the protocols in it. Wireshark is used for network troubleshooting, analysis, software and communications protocol development, and education. Wireshark is widely used by network admins to analyze network performance. Wireshark is able to capture data or information that passes through a network that we observe. In other words, Wireshark is used to find out events that occur when we interact with the internet [7].

C. System planning

In system design, we will discuss the application of the RED and HTB methods using the RB951G-2HND routerboard proxy. The architecture for implementing the RED and HTB methods can be seen in Figure 1 below.

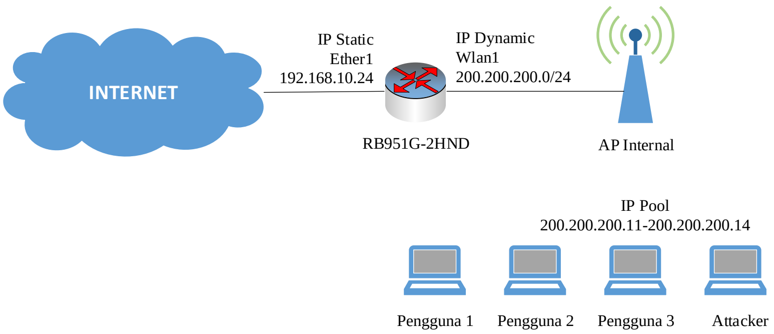

Figure 1. Architecture for implementing the RED and HTB methods

Based on Figure 1, the RED and HTB methods are implemented on a WLAN network using the Mikrotik RB951G-2HND Routerboard. This network working system, after applying the RED and HTB queuing methods on the WLAN network, then downloads, uploads and streams as test samples. In addition, Noise Injection is given to the test sample. After that, recording is carried out according to the QoS parameters via the Wireshark application which is taken from the user's side

D. System Block Diagram

1. Design of Random Early Detection Method

To design the RED Method, the bandwidth received from the ISP is then channeled to the Mikrotik Routerboard to configure the RED method on the IP firewall mangle then create connection marks and packet marks to filter packets such as upload, download and streaming packets. After determining the upper limit value, lower limit value and average queue value, the final step is to send the packet to the user's computer based on the packet request. The stages carried out in implementing the RED method can be seen in Figure 2 below.

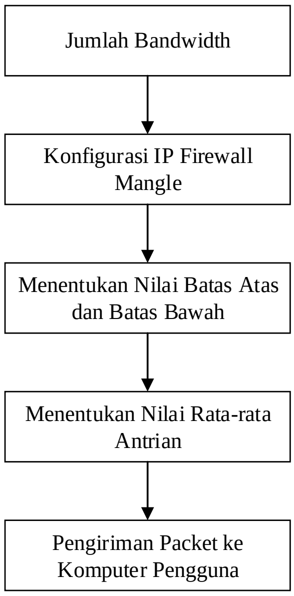

Figure 2. Design stages of the RED method

2. Hierarchical Token Bucket Method Design

To design the HTB method, the bandwidth received from the ISP is then channeled to the Mikrotik Routerboard to configure the RED method on the IP firewall mangle then create connection marks and packet marks to filter packets such as upload, download and streaming packets. After determining the max-limit value on the parent, the limit-at and max-limit values on the child, the final step is to send the packet to the user's computer based on the packet request. The stages carried out in implementing the HTB method can be seen in Figure 3 below.

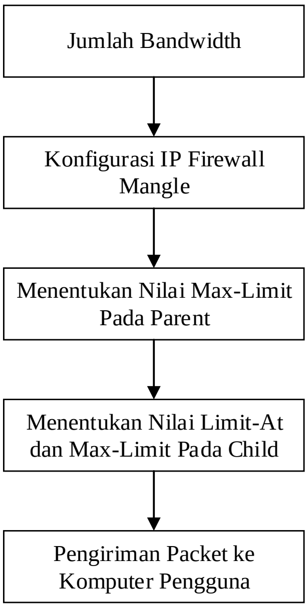

Figure 3. Stages of HTB Method Design

E. System Testing Block Diagram

System testing aims to determine whether the RED and HTB methods have been implemented. As for overall system testing, it can be seen in Figure 4 and Figure 5 below.

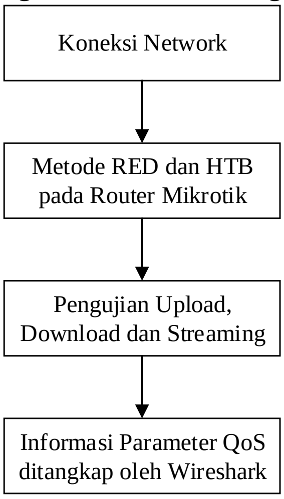

Figure 4. System Testing

Based on figure 4. The first stage of testing the system must be connected to the ISP network then configure the IP Address, Gateway, DNS, NAT and Hotspot. Next, in the RED and HTB methods, a mangle is created as a marker that marks the packet during processing using special marks in the Router. The RED method must create a queue type then continue with a queue tree. On the other hand, the HTB method only creates a queue tree. After completing the configuration, it continues with testing of various aspects of usability, namely uploading, downloading and streaming based on test scenarios and the provisions of the QoS parameters captured by Wireshark.

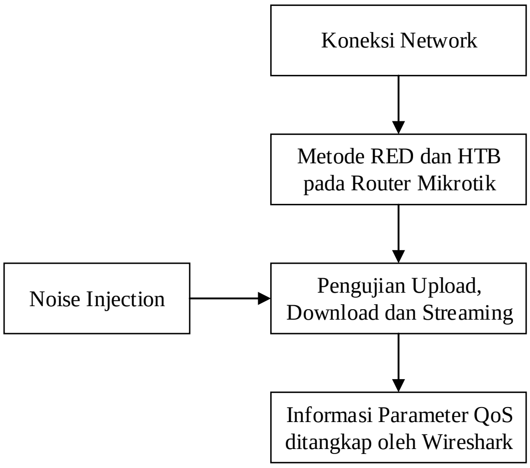

Figure 5. System Testing with Noise Injection

Based on Figure 5. The final stage of testing the system provided by Noise Injection must be connected to the ISP network then configure the IP Address, Gateway, DNS, NAT and Hotspot. Next, in the RED and HTB methods, a mangle is created as a marker that marks packets during processing using special marks in the router. The RED method must create a queue type followed by a queue tree. On the other hand, the HTB method creates a queue tree. After completing the configuration, it continues with testing of various aspects of usability, namely uploading, downloading and streaming based on test scenarios and the provisions of the QoS parameters captured by Wireshark. Then noise injection is given to disrupt or intercept packets from the communicating party.

RESULTS AND DISCUSSION

The network quality test results of the Random Early Detection and Hierarchical Token Bucket methods on the Wireless Local Area Network network are in accordance with QoS standards in the form of Throughtput, Delay, Jitter and Packet Loss values from each method based on the test scenario in table 1, for various usability aspects, namely upload , downloading and streaming by 3 users is carried out at the same time and is given Noise Injection so that the quality on the network is known. Data collection was carried out for 100 seconds using Wireshark software from the user's side. The test scenarios can be seen in table 1 below.

TABLE I
TESTING SCENARIOS

N0	Test	User	Activities performed
1	1st try	User 1 Users 2 to 3	*Upload* Shut up
2	2nd try	User 1 Users 2 to 3	*Downloads* Shut up
3	3rd try	User 1 Users 2 to 3	*Streaming* Shut up
4	4th try	Users 1 to 2 User 3	*Upload* Shut up
5	5th try	Users 1 to 2 User 3	*Downloads* Shut up
6	6th try	Users 1 to 2 User 3	*Streaming* Shut up
7	7th try	Users 1 to 2 User 3	*Upload* *Downloads*
8	8th attempt	Users 1 to 2	*Upload*

		User 3	*Streaming*
9	9th attempt	Users 1 to 2	*Downloads*
		User 3	*Upload*
10	10th try	Users 1 to 2	*Downloads*
		User 3	*Streaming*
11	11th attempt	Users 1 to 2	*Streaming*
		User 3	*Upload*
12	12th attempt	Users 1 to 2	*Streaming*
		User 3	*Downloads*
		User 1	*Upload*
13	13th attempt	User 2	*Downloads*
		User 3	*Streaming*
		User 1	*Upload*
14	14th attempt	User 2	*Upload*
		User 3	*Upload*
		User 1	*Downloads*
15	15th try	User 2	*Downloads*
		User 3	*Downloads*
		User 1	*Streaming*
16	16th attempt	User 2	*Streaming*
		User 3	*Streaming*

Throughput Testing

Throughput testing is carried out to determine the effective data transfer speed (rate). Throughput describes the actual amount of data received in each second. So the greater the throughput value, the better the network. Each result from the testing process is added up to obtain an average throughput value which can be seen in table 2 and figure 6 below.

TABLE II
THROUGHPUT TEST RESULTS

RED Method				HTB method			
No Noise Injection		*Noise Injection*		No Noise Injection		*Noise Injection*	
Throughput(bps)	Quality	*Throughput*(bps)	Quality	*Throughput*(bps)	Quality	*Throughput*(bps)	Quality
816287	Very good	704778	Very good	879128	Very good	582716	Very good

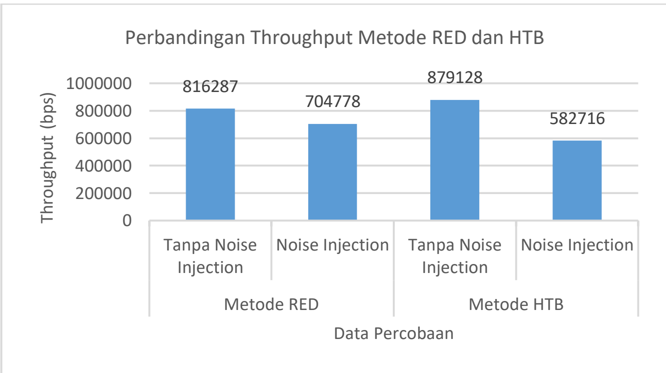

Figure 6. Throughput Test Results

From testing, there are differences in throughput in the RED and HTB methods. The throughput of the RED and HTB methods in table 2 is shown in figure 6 showing that each method has different results and characteristics. The RED method by controlling the average queue with a throughput value of 816287 bps is included in the very good category. Meanwhile, the RED method which was given Noise Injection with a Throughput value of 704778 bps is included in the very good category. On the other hand, the HTB method has a throughput value of 879128 bps, which is in the very good category. Meanwhile, the HTB method given Noise Injection with a Throughput value of 582716 bps is included in the very good category. Of the two methods, HTB without Noise Injection has better performance compared to the RED method, seen from the throughput value which is greater than the RED method.

1. Delay Testing

Delay testing is carried out to determine the time required for data to travel the distance from origin to destination. So the smaller the delay value in a network, the better the network. Each result from the testing process is added up to obtain an average delay value which can be seen in table 3 and figure 7 below.

TABLE III
DELAY TEST RESULTS

RED Method				HTB method			
No Noise Injection		*Noise Injection*		No Noise Injection		*Noise Injection*	
Delay(ms)	Quality	*Delay(ms)*	Quality	*Delay(ms)*	Quality	*Delay(ms)*	Quality
8,673	Very good	11,154	Very good	8	Very good	15,427	Very good

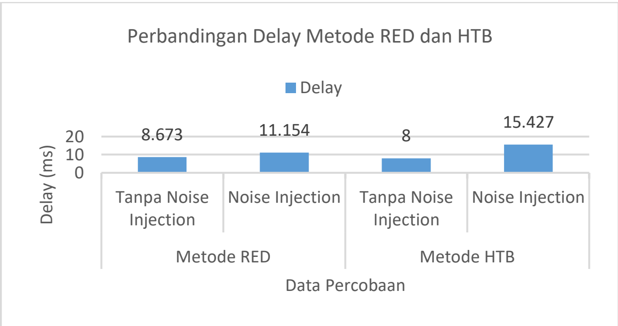

Figure 7. Delay Test Results

The comparison results of the RED and HTB Delay Methods in table 3 are shown in figure 7 showing that each method has different results and characteristics. The RED method by controlling the average queue with a Delay value of 8,673 ms is included in the very good category. On the other hand, the RED method which was given Noise Injection with a Delay value of 11,154 ms is included in the very good category. Meanwhile, the HTB method which has a Delay value of 8 ms is included in the very good category. On the other hand, the HTB method which was given Noise Injection with a Delay value of 15,427 ms is included in the very good category. Of the two methods, HTB without Noise Injection has better performance compared to the RED method, seen from the smaller Delay value compared to the RED method.

2. Jitter Testing

Jitter testing is carried out to determine variations in packet arrival. So the smaller the jitter value, the better the network. Each result from the testing process is added up to obtain an average jitter value which can be seen in table 4 and figure 8 below.

TABLE IV
JITTER TEST RESULTS

RED Method				HTB method			
No Noise Injection		*Noise Injection*		No Noise Injection		*Noise Injection*	
Delay(ms)	Quality	*Delay(ms)*	Quality	*Delay(ms)*	Quality	*Delay(ms)*	Quality
12,591	Very good	15,565	Very good	11,041	Very good	21,924	Very good

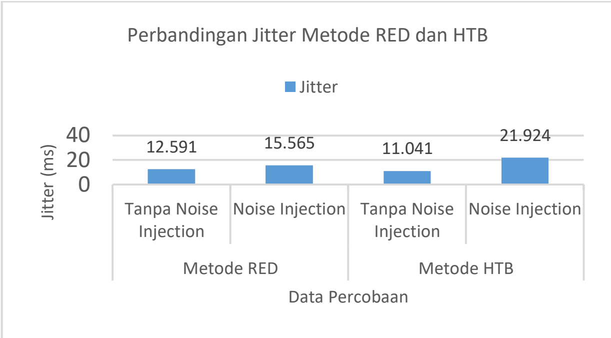

Figure 8. Jitter Test Results

The Jitter comparison results of the RED and HTB methods in table 4 are shown in figure 8 showing that each method has different results and characteristics. The RED method by controlling the average queue with a Jitter value of 12,591 ms is included in the good category. On the other hand, the RED method which was given Noise Injection with a Jitter value of 15,565 ms is included in the good category. Meanwhile, the HTB method which has a Jitter value of 11,041 ms is included in the good category. On the other hand, the HTB method which was given Noise Injection with a Jitter value of 21,924 ms is included in the good category. Of the two methods, HTB without Noise Injection has better performance compared to the RED method as seen from the smaller Jitter value compared to the RED method.

3. Packet Loss Testing

Packet loss testing is carried out to determine the percentage of packet loss when sending data. Each result from the testing process is added up to obtain an average packet loss value which can be seen in table 5 and figure 9 below.

TABLE V
PACKET LOSS TEST RESULTS

RED Method				HTB method			
No Noise Injection		*Noise Injection*		No Noise Injection		*Noise Injection*	
Packet Loss(%)	Quality	*Packet Loss* ((%))	Quality	*Packet Loss* ((%))	Quality	*Packet Loss* ((%))	Quality
0	Very good	2.14	Very good	0	Very good	2.8	Very good

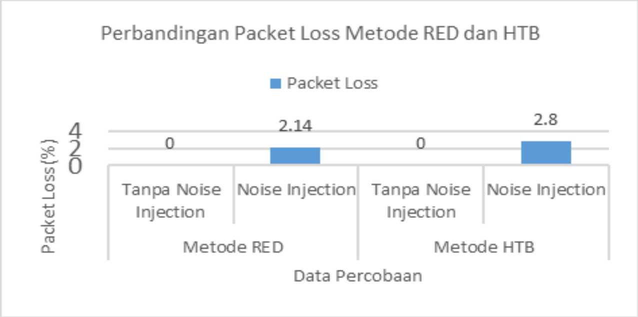

Figure 9. Packet Loss Test Results

The results of the Packet Loss comparison of the RED and HTB methods in table 5 are shown in figure 9, showing that each method has different results and characteristics. The RED method by controlling the average queue with a Packet Loss value of 0% is included in the very good category. On the other hand, the RED method which was given Noise Injection with a Packet Loss value of 2.14% is included in the good category. Meanwhile, the HTB method which has a Packet Loss value of 0% is included in the very good category. On the other hand, the HTB method which was given Noise Injection with a Packet Loss value of 2.8% is included in the good category. Of the two methods, HTB and RED without Noise Injection have the same performance as seen from the Packet Loss value of 0% in both methods.

CLOSING

Based on the results and discussion of the research that has been carried out, it can be concluded that:

1. Network quality using the Random Early Detection method without noise injection, with values for QoS parameters that have been measured, namely: Throughput value 816287 bps with very good quality, Delay value 8,673 ms with very good quality, jitter value 12,591 ms with good quality and packet value 0% loss with very good quality.
2. The quality of the network using the Random Early Detection method is given noise injection, with values for the QoS parameters that have been measured, namely: Throughput value 704778 bps with very good quality, Delay value 11,154 ms with very good quality, jitter value 15,565 ms with good quality and packet value loss 2.14% with good quality.
3. Network quality using the Hierarchical Token Bucket method without noise injection, with values for QoS parameters that have been measured, namely: Throughput value 879128 bps with very good quality, Delay value 8 ms with very good quality, jitter value 11,041 ms with good quality and packet value 0% loss with very good quality.
4. The quality of the network using the Hierarchical Token Bucket method is given noise injection, with the QoS parameter values that have been measured, namely: Throughput value of 582716 bps with very good quality, Delay value of 15,427 ms with very good quality, jitter value of 21,924 ms with good quality and packet value loss 2.8% with good quality.
5. The Hierarchical Token Bucket method gets better network quality as seen from the throughput rate which is greater, delay and jitter is smaller than the Random Early Detection method

.

REFERENCES

CP Antodi, AB Prasetijo and ED Widianto, "Implementation of Quality of Service on Internet Networks Using the Hierarchical Token Bucket Method," Computer Technology and Systems, vol. 5, no. 1, pp. 23-28, 2017.

A. Akmal, F. Susanti and MI Iskandar, "Configuration and Analysis of Bandwidth Management on PC Routers Using the HTB (Hierarchy Token Bucket) and CBQ (Class Based Queue) Methods. Case Study of the Bukittinggi National Land Agency Office," pp. 1-8, 2011.

H. Agung, "Analysis of the RED and PCQ Methods in the Cibuntu-Kuningan Tourism Village Mikrotik," DASI, vol. 18, no. 2, pp. 13-18, 2017.

T. Pratama, M. Irwansyah and Yulianti, "Comparison of PCQ, SFQ, RED and FIFO Methods on Mikrotik as an Effort to Optimize Network Services at the Faculty of Engineering, Tanjungpura University," Informatics Engineering, pp. 1-6, 2015.

AI Wijaya and LB Handoko, "Bandwidth Management Using the HTB (Hierarchical Token Bucket) Method at Semarang State 5 Junior High School," UDINUS Informatics Engineering, pp. 1-3, 2016.

KA Wilmadi, M. Kusban and FY Irsyadi, "Analysis of Bandwidth Management Using PCQ (Per Connection Queue) and HTB (Hierarchical Token Bucket) Methods Using a Mikrotik Router," Unpublished, Surakarta, 2013.

Sukri and Jumiati, "Bandwidth Analysis Using the Queue Per Connection Queue Method," Univrab Information Technology and Systems, vol. 2, no. 2, pp. 244-257, 2017.

AN Wicaksono, "Qos (Quality Of Service) Analysis of Wireless Local Area Network Networks at the Faculty of Engineering, Yogyakarta State University," Unpublished, Yogyakarta, 2016.

NR Wiharto, "Explanation of the IEEE 802.11 a/b/g/n/ac Code on Wireless LAN (Wi-fi) Devices," [Online]. Available: http://www.pintarkomputer.com/penjualan-tangan-kode-ieee-80211-abgn-ac-pada-perangkat-wireless-lan-wifi. [Accessed April 2, 2019].

SK P, GA Mutiara and MI Iskandar, "Implementation of the Per Connection Queue (PCQ) Algorithm in the Hierarchical Token Bucket (HTB) Algorithm for Bandwidth division in the Khelambiqunet internet cafe," pp. 1-7, 2012.