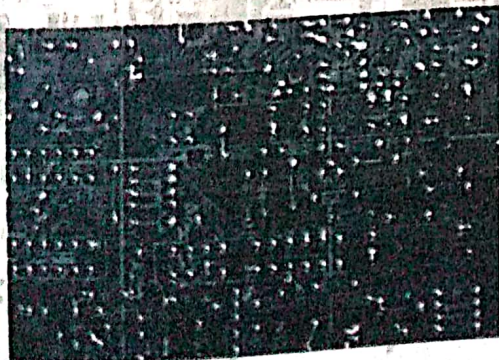
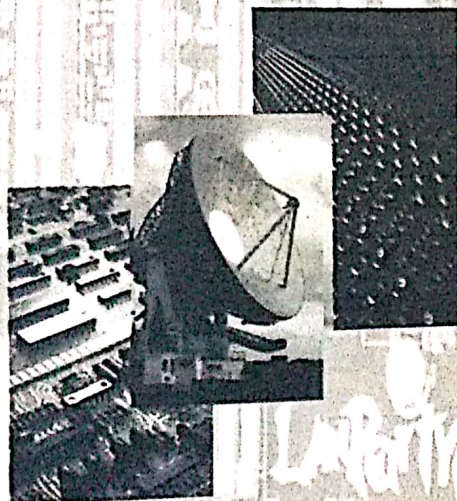
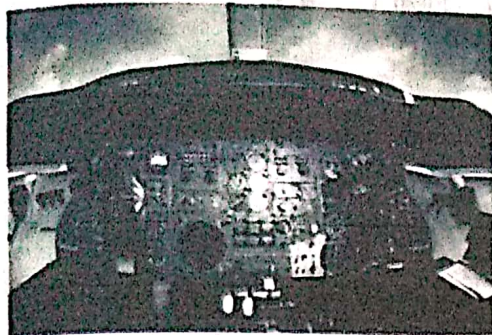




Sinusoida

Jurnal Penelitian dan Kajian Ilmu Teknik



Daftar Isi

Hal

- *Rancang Bangun Sistem Transportasi Booking Berbasis Sms Gateway Menggunakan Software GAMMU*
Edy Supriyadi dan Ayu Zuafah 1
- *Rancangan Antena Panel Mikrostrip 8 Larik Pada Aplikasi Wimax Frekuensi 2,4 Ghz*
Aditya Indra P dan Heru Abrianto..... 11
- *Evaluasi Kualitas Jaringan Serat Optik Dwdm 40λ Melalui Perhitungan Dan Pengukuran Untuk Perbaikan Kinerja*
Arfian Ahmad dan Zaqqy Kevin Laison 20
- *Perancangan Dan Implementasi Multicast Routing Pada Trafik Video Streaming Untuk Mengatasi Keterbatasan Bandwidth*
Mufti Gafar dan Yuli Yektiani 28
- *Perancangan Sistem Kendali Lampu Dan AC Pada NETWORK OPERATION CENTER ISTN Melalui Wifi Web Server Berbasis Arduino Uno*
Rachman Soleman, M. Febriansyah dan Mohammad Rizardi 37
- *Metode LOW LATENCY QUEUING untuk Mengurangi Terjadinya Kongesti Data pada Jaringan IP*
Febriansyah dan Irmayani 48
- *Implementasi GZIP COMPRESSION untuk Mempercepat Pengiriman Data di Jaringan WAN*
Eyasyer Hendra dan Mohammad Hamdani 58
- *Perencanaan Instalasi Listrik untuk Seksi Pengecatan di Industri Otomotif*
Sugianto, A. Muis dan Arief Priambudi 65
- *Pengaturan Kecepatan Motor Induksi Fasa Tiga untuk Menjaga Tekanan Tetap pada Pipa Outlet Motor Pompa Injektor CHLORINE*
Mohammad Amir dan Adis Suhur Nahar..... 78
- *Perbaikan Performansi Video Call Menggunakan Jaringan LTE*
H. Budihardjo Gozali dan Arnold Julyus.S 92

METODE *LOW LATENCY QUEUING* UNTUK MENGURANGI TERJADINYA KONGESTI DATA PADA JARINGAN IP

Febriansyah & Irmayani
Program Studi Teknik Elektro, Fakultas Teknologi Industri
Institut Sains Dan Teknologi Nasional
Email : ir.irmayani@istn.ac.id

Abstrak

Low Latency Queuing atau *LLQ* merupakan fitur *QoS* pada jaringan IP yang digunakan oleh vendor perangkat *Cisco* untuk memberikan layanan yang bersifat low-latency pada jenis data yang diinginkan. *LLQ* mengklasifikasikan data yang sensitif terhadap *delay* dan *packet loss* kemudian memberikan aturan agar data tersebut selalu diprioritaskan dalam antrian di jaringan dibanding data yang lain.

Pada tulisan ini, metode *LLQ* diujicoba pada jaringan sebesar 128 Kbps yang mengalami kongesti dengan prioritas data *VoIP*. Percobaan dilakukan dengan melakukan panggilan telepon *VoIP*, *FTP* transfer, test ping dan *browsing* secara bersamaan. Dari hasil pengujian secara *default*, *delay VoIP* didapat hingga 583 ms, dengan *throughput* rata-rata 25,191 kbps. Hal tersebut dikarenakan paket *VoIP* harus menunggu proses antrian data besar lain yang membutuhkan waktu hingga 94,942 ms tiap paketnya dengan utilitas yang didominasi oleh *FTP* transfer hingga 88.8 Kbps. Setelah diterapkan metode *LLQ* didapat *delay VoIP* 11 mili detik dengan *throughput* 31,94 Kbps.

Optimasi yang dihasilkan metode *LLQ* tersebut karena *LLQ* memprioritaskan paket *VoIP* dan mensegmentasi paket data lainnya menjadi fragmen-fragmen maksimal sebesar 160 Bytes. Dengan hasil percobaan tersebut, dapat disimpulkan bahwa metode *LLQ* cocok untuk diterapkan pada aplikasi interaktif seperti *VoIP* pada jaringan yang padat sehingga diharapkan dapat menjadi solusi alternatif bagi pengguna jaringan yang memiliki bandwidth terbatas.

Kata kunci : *bandwidth, network congestion, delay, throughput, FIFO, LLQ.*

Abstract

Low Latency Queuing or *LLQ* is a feature of *QoS* over IP network used by *Cisco* to provide low-latency service on the desired data type. *LLQ* classify delay and drop sensitive data and then policed them to always be prioritized among the others. In this paper, *LLQ* method is being tested to prioritize *VoIP* traffic on congested network that has speed of 128 Kbps. The trials run by doing *VoIP* call, *FTP* transfer, ping test and *browsing* at the same time. At the default configuration method, the result of *VoIP* delay is up to 583 ms, with the *throughput* of 25,191 Kbps. These result is because of *VoIP* packet have to wait others queued data that takes time up to 94,942 ms per packet with the utilization is dominated by the *FTP* transfer up to 88.8 Kbps. After *LLQ* method applied, the result of *VoIP* delay become 11 ms with the *throughput* of 31,94 Kbps. The optimization by *LLQ* method obtained by prioritize *VoIP* packet and segmentize others data packet to be 160 bytes maximum of fragments. With these result, the conclusion can be taken that *LLQ* method is suitable to be applied on congested network has *VoIP* application with the limited bandwidth.

Keywords : Bandwidth, Network Congestion, Delay, Throughput, FIFO, LLQ.

1. Pendahuluan

Setiap pengguna telekomunikasi memiliki kebutuhan untuk berkomunikasi seefisien dan semaksimal mungkin dengan kualitas maksimal yang didapatkan dan biaya minimum yang diperlukan.

Dalam penerapannya di lapangan, seringkali pengguna merasakan layanan komunikasi yang lambat terutama pada aplikasi yang sensitif terhadap *delay* dan *packet loss* seperti *VoIP*. Hal ini umumnya disebabkan karena bandwidth tidak mencukupi sehingga terjadi kongesti pada jaringan. Solusi utama dari permasalahan tersebut adalah dengan menambah kapasitas lebar jalur, yang tentunya tidak semua pengguna dapat memenuhinya.

Pada tulisan ini, dibahas mengenai solusi lain yang bersifat alternatif untuk mengatur lalu lintas data berbasis IP semaksimal dan seefisien mungkin terhadap kualitas yang didapatkan saat terjadi kongesti data pada jaringan. Teknologi antrian dan pengaturan kongesti data yang diterapkan yaitu *Low Latency Queuing* (*Cisco Proprietary*) yang mengenali, mengklasifikasikan dan

memberikan pengaturan dan prioritas terhadap jenis layanan jaringan diinginkan. Diharapkan dengan metode ini dapat meminimalkan terjadinya kongesti data sehingga didapat performa jaringan yang maksimal terutama pada aplikasi yang diprioritaskan.

2. Tinjauan Pustaka

2.1 Quality of Service Pada Jaringan IP

QoS (*Quality of Service*) secara bahasa adalah nilai layanan secara keseluruhan yang dirasakan oleh pengguna. Secara istilah dalam jaringan komputer, *QoS over IP Network* secara singkat adalah "kemampuan dari suatu Jaringan IP untuk memberikan layanan yang baik untuk suatu tipe data atau *traffic* tertentu dengan memberikan suatu aturan prioritas".

QoS dibutuhkan pada suatu jaringan yang mengalami masalah seperti *slow data rate transfer*, gagal proses aplikasi dan suara putus-putus pada *VoIP*. Solusi utama dari masalah jaringan tersebut umumnya akan terselesaikan dengan menambah kapasitas *bandwidth*,

tetapi dengan menerapkan *QoS* dapat menjadi suatu pilihan untuk meminimalkan *packet loss*, *delay* dan *delay variation (jitter)* pada jaringan yang padat dengan *bandwidth* yang terbatas.

Fitur *QoS* dalam memberikan kualitas jaringan yang lebih baik yaitu dengan menyediakan layanan :

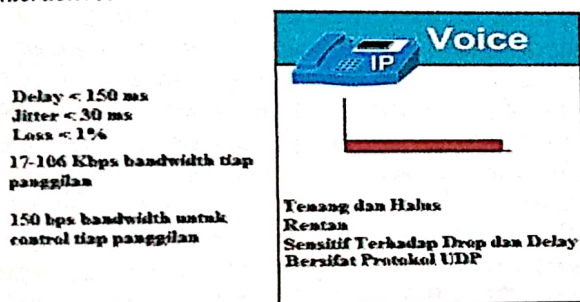
- Memberikan *bandwidth* yang terjamin
- Mengurangi hal-hal yang menyebabkan rugi-rugi dalam jaringan
- Menghindari dan mengatur kongesti data pada jaringan
- Membentuk dan menentukan lalu lintas data
- Memberikan jalur prioritas sepanjang lintasan data

2.2 Jenis Data Dalam Lalu-Lintas Jaringan

Jenis data yang umum ada dalam jaringan adalah data, suara, dan video atau (*triple-play*). Setiap jenis data tersebut memiliki karakteristik dan kebutuhan yang berbeda. Nilai data dinyatakan dalam satuan bit (b) atau Byte (B).

a. Voice

Voice adalah tipe data yang sangat sensitif terhadap *delay* dan *packet drops* yang terjadi. *Voice* memiliki karakteristik *connection-oriented* dengan protokol *UDP*, yang berarti tidak mengirim ulang paket walaupun ada *packet drops* yang terjadi karena bersifat *realtime interactive*.



Gambar 2.1. Karakteristik *Traffic Voice*

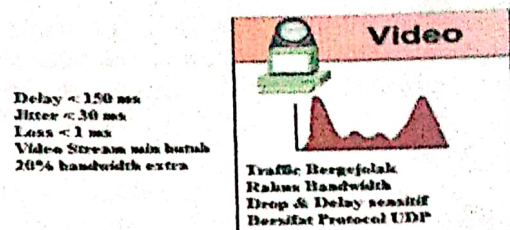
Jenis data *voice* bersifat *real-time* yang pada jaringan ditangani oleh protokol *RTP* atau *real-time protocol*. ITU dan Cisco memiliki nilai rekomendasi untuk *one-way delay* per paket *voice* yaitu kurang dari 150 mili detik (ITU) dan 200 mili detik (Cisco)

Tabel 2.1. Rekomendasi *Delay Traffic Voice*

Delay Search	Deskripsi
0-150 ms	Rekomendasi ITU
0-200 ms	Rekomendasi Cisco
150-400 ms	Degradasi layanan
400 ms +	Nilai tidak layak

b. Video

Traffic video pada jaringan juga memiliki karakteristik dan sensitifitas yang serupa dengan *traffic voice*, bersifat *interactive realtime* dan bersifat protokol *UDP*, bahkan membutuhkan *bandwidth* yang lebih besar dengan jaminan kebutuhan 20% ekstra *bandwidth* yg dibutuhkan.



Gambar 2.2. Karakteristik *Traffic Video*

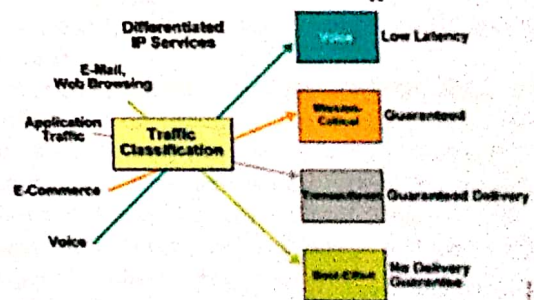
Seperti halnya *VoIP*, *Video* dikirimkan dalam sebuah jaringan juga menggunakan kompresi video, yang umum digunakan adalah teknologi *MPEG*. ITU juga memiliki rekomendasi nilai minimum *bandwidth* yang diperlukan untuk setiap teknologi kompresi video yang umum digunakan pada tabel 2.1. Seperti pada jenis data *voice*, data video juga ditangani oleh protokol *RTP* dalam jaringan

c. Data

Traffic data bersifat *connection less*, yang memiliki sifat *best-effort* sampai tujuan. *Traffic data* memiliki protokol transport *TCP* yang artinya akan mengirimkan ulang data yang mengalami *drop* selama transmisi. *TCP* tidak terlalu sensitif terhadap *delay* dan *drop* karena memiliki sifat *retransmission*. Pada dasarnya tipe data sendiri umumnya dibedakan menjadi 4 yang digambarkan pada gambar 2.3 dengan karakteristik tersendiri yang dimilikinya.



Gambar 2.5. Karakteristik *traffic Data*



Gambar 2.3. Tipe Data Dan Kebutuhannya

2.3. Proses dan Struktur Paket IP

Pada jaringan IP, data diproses dalam bentuk paket yang berisi *payload* atau data dan *network header*. *QoS* bekerja dengan mengenali struktur paket-paket tersebut dan memprosesnya. Paket IP diatur dalam layer *Network* atau *Internet* pada protokol *OSI* dan *TCP/IP*.

Display		By data	
Display filter:		0 (0.000%)	
Ignored packets:			
Traffic	Captured	Displayed	Displayed
Packets	46424	3468	5.216%
Between first and last packet	541.877 sec	491.866 sec	
Avg. packets/sec	122.808	7.501	
Avg. packet size	218 bytes	1513 bytes	
Bytes	14479111	5243298	36.213%
Avg. bytes/sec	26768.703	11243.892	
Avg. MB/s/sec	0.254	0.081	

Gambar 4.10. Throughput FTP

Setelah kelas data FTP diberikan *bandwidth* 64 Kbps dan diatur oleh metode LLQ dilakukan percobaan lagi dalam jaringan yang *congest* dan didapat *throughput* 70 Kbps seperti gambar 30. Dari hasil pengukuran dengan wireshark, didapat penurunan *throughput* pada aktifitas FTP transfer dikarenakan metode LLQ memberikan maksimal *bandwidth* kepada kelas data FTP transfer maksimal 64 Kb saat jaringan kongesti.

Display		By data	
Display filter:		0 (0.000%)	
Ignored packets:			
Traffic	Captured	Displayed	Displayed
Packets	8862	4107	45.17%
Between first and last packet	463.264 sec	455.147 sec	
Avg. packets/sec	19.345	8.823	
Avg. packet size	475 bytes	967 bytes	
Bytes	4254143	3973248	93.87%
Avg. bytes/sec	9182.581	8728.576	
Avg. MB/s/sec	0.073	0.070	

Gambar 4.11. Throughput FTP (LLQ)

Selanjutnya untuk paket test-ping yang umum dijadikan acuan seorang user menilai sebuah jaringan juga akan diukur berapa *success-rate* dari 100 kali pengiriman paket ICMP ping, dan hasil yang didapat dari pengujian saat *network congestion* adalah 85%. Dengan hasil tersebut, pengguna akan menilai jaringan yang ada mengalami degradasi performansi.

Display		By data	
Display filter:		0 (0.000%)	
Ignored packets:			
Traffic	Captured	Displayed	Displayed
Packets	100	85	85.00%
Between first and last packet	0.000 sec	0.000 sec	
Avg. packets/sec	100	85	
Avg. packet size	60 bytes	60 bytes	
Bytes	6000	5100	85.00%
Avg. bytes/sec	6000	5100	
Avg. MB/s/sec	0.006	0.005	

Gambar 4.12. Hasil Test Ping

Setelah paket ICMP dari test-ping diberikan *bandwidth* 24 Kbps oleh LLQ dilakukan test-ping lagi selama 100 kali dan didapatkan *success rate* 97% terlihat pada gambar 4.12. Terlihat bahwa metode LLQ mengoptimalkan aktifitas test ping dengan memberikan *bandwidth* kepada aktifitas tersebut sebesar 24 Kb kedalam reserved queue. Dengan hasil ini, user akan lebih percaya dengan kualitas jaringan dibanding yang sebelumnya karena hasil test ping yang dilakukan mengalami optimasi dibanding tanpa metode LLQ.

Gambar 4.13. Hasil Ping Test (LLQ)

4.2. Perhitungan Parameter Jaringan

Hal Pertama yang dilakukan setelah didapat hasil dari pengujian selama *network congestion* adalah mengenali parameter jaringan 128 Kbps saat dalam kondisi jaringan normal dengan cara menghitung kebutuhan *bandwidth* dan *delay* yang dibutuhkan oleh panggilan telepon VoIP, data transfer dan test-ping saat kondisi normal atau saat *bandwidth* mencukupi dan dibandingkan hasilnya dengan percobaan saat jaringan kongesti.

4.2.1. Perhitungan Delay Propagasi

Nilai *delay* propagasi pada jaringan bernilai tetap karena dipengaruhi oleh propagasi jalur transmisi berupa fiber optik sepanjang ± 20 Km yaitu :

$$d_{prop} = \frac{D}{p}$$

$$d_{prop} = \frac{20000 \text{ meter}}{3 \times 10^8 \frac{\text{m}}{\text{s}}} = 0.067 \text{ ms}$$

Nilai *delay* propagasi tersebut sangat kecil karena jarak yang tidak begitu jauh dan menggunakan media fiber optik yang merupakan teknologi akses yang paling handal sampai saat ini, sehingga nilai *delay* propagasi pada Jaringan hampir dapat diabaikan dalam perhitungan.

4.2.2. Perhitungan Paket VoIP

Pada pengujian yang telah dilakukan, *codec* VoIP yang digunakan adalah G.729 yang memiliki parameter tiap paketnya :

- Payload = 20 Bytes
- Sampling = 20 ms
- IP Header = 20 Bytes
- UDP + RTP Header = 20 Bytes
- Layer 2 Ethernet Frame Header = 18 Bytes

Sehingga *bandwidth* yang dibutuhkan setiap detik atau data rate yang terjadi dapat dihitung dengan rumus:

$$DR = \frac{b}{T} = \frac{(20 \text{ Bytes} + 20 \text{ Bytes} + 20 \text{ Bytes} + 18 \text{ Bytes}) \times 8}{20 \text{ ms}}$$

$$DR = \frac{624 \text{ bits}}{20 \text{ ms}} = 31,2 \text{ kbps}$$

Sedangkan untuk *delay* pengiriman setiap paketnya pada WAN sebesar 128 Kbps dapat dihitung dengan rumus:

$$(d_{end-to-end}) = d_{sz} + d_{prop} + d_q$$

Dimana nilai *delay* serialization untuk setiap paket suara dapat dihitung dengan rumus 2.4 yaitu :

$$d_{sz} = \frac{b}{DR}$$

$$d_{sz} = \frac{624 \text{ bit}}{128000 \text{ bps}} = 4,875 \text{ ms}$$

Maka nilai *delay* yang terjadi untuk satu paket VoIP dengan *codec* G.729 pada jaringan jika jaringan dalam keadaan normal adalah :

$$(d_{\text{VoIP}}) = d_{sz} + d_{prop} + d_q$$

$$(d_{\text{VoIP}}) = 4,875 + 0,067 + 0 = 4,942 \text{ ms}$$

4.2.3. Perhitungan Delay Per Paket FTP Transfer

Data transfer yang dilakukan dalam pengujian ini menggunakan *protocol* FTP transfer dengan data file sebesar 5 MB atau sekitar 40 Mbit. Setiap satu paket data sebesar 1518 Bytes membutuhkan waktu delay pada jaringan yaitu :

$$d_{sz} = \frac{b}{DR}$$

$$d_{sz} = \frac{1518 \text{ Bytes} \times 8}{128000 \text{ bps}} = 94,875 \text{ ms}$$

$$(d_{\text{ftp}}) = d_{sz} + d_{prop} + d_q$$

$$(d_{\text{ftp}}) = 94,875 + 0,067 + 0 = 94,942 \text{ ms}$$

Karena teknologi media yang digunakan pada pengujian jaringan ini memiliki *protocol standard Ethernet* 802.3 yang mempunyai *MTU (Maximum Transfer Unit)* 1518 Bytes, maka data file file sebesar 40 Mbit akan difragmentasi menjadi paket data yang maksimal berukuran 1518 Bytes atau sekitar 12 Kb (sudah termasuk data header).

4.2.4. Perhitungan Delay Per Paket Ping

Ping test merupakan parameter yang biasa digunakan oleh user untuk menilai kualitas suatu jaringan berdasarkan nilai *success rate*-nya. Pada pengujian ini dilakukan ping-test dari PC ke PC sebanyak 100 paket. Setiap paket ping yang dikirimkan memiliki *protocol ICMP* dengan data sebesar 32 Bytes dan 8 Bytes header ICMP, sehingga perhitungannya *delay* per paketnya :

$$d_{sz} = \frac{b}{DR}$$

$$d_{sz} = \frac{(32 \text{ Bytes} + 8 \text{ Bytes} + 20 \text{ bytes} + 18 \text{ bytes}) \times 8}{128000 \text{ bps}} = 4,8 \text{ ms}$$

$$(d_{\text{ping}}) = d_{sz} + d_{prop} + d_q$$

$$(d_{\text{ping}}) = 4,875 + 0,067 + 0 = 4,942 \text{ ms}$$

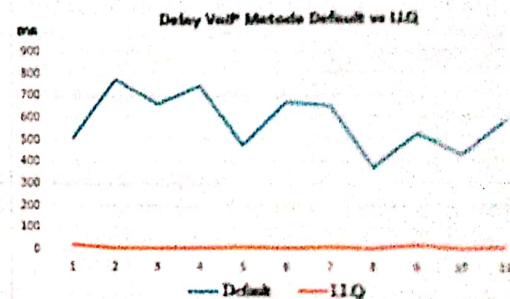
4.3. Analisa Hasil Pengujian VoIP Priority Queue

Hasil pengujian implementasi *congestion mangement Low latency Queuing* difokuskan pada paket data suara atau panggilan VoIP karena paket data tersebutlah yang paling sensitif terhadap *delay* dan *packet loss* sehingga menjadi paket data yang diprioritaskan dalam pengujian dibanding yang lainnya.

Dari hasil pengujian didapat bahwa metode *default* atau secara *FIFO* tidak sesuai untuk paket suara. Minimum *delay* yang dihasilkan oleh panggilan telepon adalah sebesar rata-rata 583 mili detik. Kemudian setelah digunakan *Low Latency Queuing (LLQ) congestion mangement*, hasil pengujian menunjukkan bahwa minimum *delay* yang dihasilkan saat panggilan telepon hanya 6 mili detik dan maksimal 25 mili detik. Dari hasil pengujian tersebut didapat bahwa *queuing delay* saat terjadi kongesti data pada jaringan dapat diminimalkan dengan metode *LLQ*.

Tabel 4.1. *Queuing Delay* dan Optimasi (mili detik)

NO	Delay Total (Default)	Queuing Delay (Default)	Delay Total (LLQ)	Queuing Delay (LLQ)
1	511	506.058	22	17.058
2	776	771.058	6	1.058
3	664	659.058	6	1.058
4	744	739.058	8	3.058
5	477	472.058	11	6.058
6	668	663.058	6	1.058
7	655	650.058	12	7.058
8	374	369.058	8	3.058
9	528	523.058	25	20.058
10	432	427.058	8	3.058
Avg	583	577.958	11	6.258



Gambar 4.14. Grafik *Delay Default VS LLQ*

Optimasi yang dihasilkan oleh *LLQ* dikarenakan metode tersebut selalu memprioritaskan paket VoIP dengan bandwidth terjamin 32 Kbps untuk diproses pertama dalam antrian dan paket lainnya disegmentasi maksimal 160 Bytes dan tidak menimbulkan *delay* yang lama dalam antrian data. Oleh metode *LLQ* Paket data lainnya dengan besaran maksimal 1518 Bytes dan memerlukan 94,942 mili detik untuk memprosesnya disegmentasi menjadi fragmen-fragmen dengan besaran maksimal 160 Bytes sehingga hanya membutuhkan waktu sebesar 10 mili detik tiap paketnya.

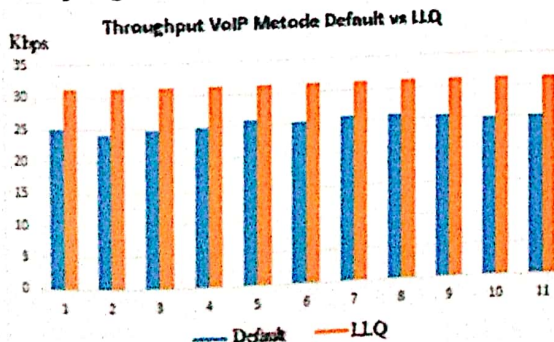
Dengan melihat hasil pengujian implementasi manajemen kongesti *Default* atau secara *FIFO* dibandingkan dengan *LLQ*, dapat disimpulkan bahwa metode *LLQ* dapat memprioritaskan suatu jenis lalu lintas data dengan handal dan stabil. Pada implementasi ini diarahkan untuk memprioritaskan lalu lintas suara sebagaimana paket suara lebih sensitif terhadap *delay* dan *packet loss* dibanding paket transfer data biasa. Paket suara yang diprioritaskan oleh *LLQ* terlihat lebih stabil, dengan nilai *packet loss* rata-rata hanya 0.02% yang artinya efisiensi *throughput* yang didapat pada paket VoIP mencapai 99.98%. Pengguna akan merasakan adanya dampak peningkatan kualitas panggilan suara bahkan saat jaringan dalam keadaan penuh atau sibuk.

Selanjutnya bisa dibandingkan hasil pengujian packet loss pada penerapan metode LLQ dengan perhitungan seharusnya dari data rate traffic VoIP yang melintasi jaringan saat kondisi normal yaitu 31,2 Kbps, sehingga bisa didapatkan nilai throughput dari traffic VoIP dengan mengalikan jumlah packet loss terhadap perhitungan data rate paket VoIP dalam keadaan normal. Berikut nilai dan performansi yang dihasilkan oleh LLQ untuk meminimalkan kongesti data yaitu sebagai berikut :

Tabel 4.2. Throughput Paket VoIP

NO	VoIP Success Packet (Default)	VoIP Success Packet (LLQ)	Throughput (Default) Kbps	Throughput (LLQ) Kbps
1	80.25%	100.00%	25.038	31.200
2	77.13%	100.00%	24.065	31.200
3	79.47%	100.00%	24.795	31.200
4	79.86%	99.97%	24.916	31.191
5	82.85%	99.99%	25.849	31.197
6	80.97%	100.00%	25.263	31.200
7	82.80%	99.90%	25.834	31.169
8	82.84%	100.00%	25.846	31.200
9	81.54%	99.96%	25.440	31.188
10	79.67%	99.99%	24.857	31.197
Average	80.74%	99.98%	25.191	31.194

Dengan hasil pengujian tersebut, terbukti paket suara yang lebih kritikal dan rentan terhadap packet loss dan delay dapat diprioritaskan. Sedangkan kelas lainnya yang dibutuhkan dalam jaringan yang bersisi paket transfer data yang bersifat TCP dan test ping dinomor duakan tetapi tetap dijamin bandwidth-nya karena tidak begitu sensitif terhadap delay dan packet loss selama ada pengecekan dan pengiriman ulang data yang loss. Sedangkan untuk jenis kelas lalu lintas data lainnya atau class-default tidak dijamin bandwidth dan layanannya dalam jaringan atau bersifat best-effort.



Gambar 4.15. Grafik Throughput Default VS LLQ
 Dari hasil dari pengujian yang didapat ini bahwa metode Low Latency Queuing dapat memberikan layanan low latency terhadap aktifitas yang sensitif seperti VoIP, diharapkan dapat menjadi solusi alternatif

bagi pengguna IP-based network yang mengalami kendala layanan jaringan karena network congestion atau bandwidth yang tidak mencukupi

5. SIMPULAN

Setelah melakukan analisa dari implementasi metode Low Latency Queuing pada jaringan, dapat diambil kesimpulan sebagai berikut.

1. Metode LLQ meminimalkan terjadinya kongesti pada jaringan dengan memberikan performansi pada queuing delay hingga hanya 6,258 mili detik sehingga nilai delay total dari paket suara mencapai 11 mili detik sesuai dengan rekomendasi ITU dan Cisco. Hal ini dikarenakan LLQ telah memprioritaskan paket VoIP dan mensegmentasi paket-paket data transfer yang besar menjadi frame-frame lebih kecil yang memiliki waktu proses antrian lebih singkat.
2. Metode LLQ pada jaringan mampu meningkatkan throughput traffic VoIP menjadi 31.194 Kbps dan efisiensi hingga 99.98% dengan meminimalkan nilai packet loss yang terjadi selama transmisi.
3. Jaringan secara default tanpa metode LLQ pada jaringan VoIP menghasilkan Queuing Delay saat kongesti data sangat besar hingga 577,958 mili detik dan packet loss hingga 19,26%.

DAFTAR PUSTAKA

- [1] Tim Szigeti, Christina Hattingh. *Quality of Service Design Overview*. Cisco Press. Dec 2004
- [2] Vegesna, Srinivas Raju. *IP Quality of Service*. Cisco Press. Jan 2001
- [3] Jonathan Davidson, James F. Peters, Manoj Bhatia, Satish Kalidindi, Sudipto Mukherjee. *Voice over IP Fundamentals, 2nd Edition*. Cisco Press. Jul 2006
- [4] Vinton G. Cerf, Robert E. Kahn. "A Protocol for Packet Network Intercommunication". *IEEE Transactions on Communications*, Vol. 22, No. 5. May 1974
- [5] Tim Szigeti, Christina Hattingh. *End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs*. Cisco Press. Dec 2004
- [6] Wallace, Kevin. *CCVP QOS Quick Reference*. Cisco Press. Sep 2006
- [7] ITU-T Series G: Transmission Systems and Media, Digital Systems and Networks, G.114. 2003
- [8] Fischer, Martin J, Masi, Denise M. Bevilacqua, Shortle, John F. *Approximating Low Latency Queuing Buffer Latency*. IEEE Computer Society, Athens. 2008.
- [9] Purbo, Onno W. *TCP/IP: Standar, Desain dan Implementasi*. Jakarta: PT Elex Media Komputindo. 2002
- [10] 802.3-2012 - IEEE Standard for Ethernet. iee.org. IEEE Standards Association. Dec 2012