

ANALISIS QOS *DIFFERENTIATED SERVICE* PADA JARINGAN MPLS MENGUNAKAN ALGORITMA *THRESHOLD*

Laufi Dian Deodo Saputra¹, Wiwin Sulisty²

^{1,2}Universitas Kristen Satya Wacana

Email : ¹laufisaputra@gmail.com, ²wiwinsulisty@staff.uksw.edu

(Naskah masuk: 15 Agustus 2017, diterima untuk diterbitkan: 30 Desember 2017)

Abstrak

Seiring perkembangan layanan komunikasi data seperti *voice* (VoIP) dan *video streaming* pada jaringan yang memiliki *buffer space* dan *bandwidth* terbatas menyebabkan terjadinya beban *traffic*. Hal tersebut membuat pengguna VoIP dan *video streaming* membutuhkan suatu jaringan yang dapat memberikan *Quality of Service* (QoS) dalam memenuhi kebutuhan pengguna. IETF (*Internet Engineering Task Force*) mempunyai standar mekanisme layanan untuk memenuhi permintaan QoS diantaranya adalah penggabungan teknologi MPLS Diffserv yang mampu mengklasifikasi paket sesuai kebutuhan, tetapi ketika penumpukan terjadi akibat proses QoS ini, paket yang menumpuk tersebut akan di-drop, maka solusi untuk mengantisipasi *dropping* digunakanlah algoritma *threshold* pada WRED. Penambahan WRED sebagai algoritma *threshold* pada jaringan MPLS Diffserv memberikan pengaruh yang signifikan, dari hasil parameter QoS untuk layanan VoIP mampu mengurangi *packet loss* 43,1%, *delay* 0,005%, memaksimalkan *throughput* 1,26% dan mengurangi *jitter* 48,56%, untuk layanan *video streaming* mengurangi *packet loss* 15,93% dan memaksimalkan *throughput* 1,6% dibandingkan sebelum menerapkan algoritma *threshold*.

Kata kunci: *QoS, Diffserv, MPLS, threshold, VoIP, video streaming*

Abstract

The development of data communication services like *voice* (VoIP) and *video streaming*, causing traffic load on networks which has limited *buffer space* and *bandwidth*. This condition makes VoIP and *video streaming* users need a network which can provide *Quality of Service* (QoS) to fill user needs. The IETF (*Internet Engineering Task Force*) has a standard service mechanism to fill QoS requests such as Incorporation of MPLS Diffserv technology which able to classify the package as needed but when the buildup occurs due to this QoS process, the packet will be dropped, Then the solution to anticipate dropping is used *threshold* algorithm on WRED. Added WRED as *threshold* algorithm on the MPLS-Diffserv network give a significant effect, from the results of QoS parameters for VoIP service is able to reduce *packet loss* 43.1%, *delay* 0.005%, maximize of *throughput* 1.26% and reduce *jitter* 48.56%, for streaming video services reduce *packet loss* 15.93% and maximize the *throughput* 1,6% than before applying the *threshold* algorithm.

Keywords: *QoS, Diffserv, MPLS, threshold, VoIP, video streaming*

1. PENDAHULUAN

Perkembangan teknologi informasi pada era globalisasi khususnya komunikasi data terus berkembang pesat. Hal ini dikarenakan layanan ini memberi kemudahan pada penggunaanya dalam berkomunikasi maupun bertukar informasi baik itu dalam bentuk data teks, suara maupun *video*.

QoS (*Quality of Service*) memegang peranan penting dalam implementasi berbagai layanan komunikasi, dikarenakan QoS merupakan parameter tingkat kualitas dari suatu layanan, akan tetapi pada jaringan IP tradisional saat ini masih berdasarkan pada *best effort services*. Dalam layanan *best effort services*, semua paket akan diperlakukan tidak sesuai kebutuhan dan diperlakukan *forwarding* yang sama. Seiring dengan perkembangan aplikasi-aplikasi seperti *voice* (VoIP) dan *video streaming* pada jaringan yang memiliki *buffer space* dan *bandwidth* yang terbatas menyebabkan adanya beban yang signifikan. Hal ini lah memicu

terjadinya *congestion* tidak bisa dihindari dalam jaringan IP *best effort* dan akan menghasilkan *packet loss*, *delay*, *throughput* dan *jitter* yang tidak baik. Hal ini akan berdampak pada kualitas dari *voice* (VoIP) dan *video* itu sendiri, karena *voice* (VoIP) dan *video streaming* sangat sensitive terhadap *delay* dan *packet loss*. Jadi, dalam arsitektur jaringan harus dilakukan semacam jaminan QoS yang akan menjamin kualitas aplikasi *voice* (VoIP) dan *video streaming* (Dhani dan Hafidudin, 2007).

Terdapat beberapa mekanisme QoS yang bisa dipetakan dalam klasifikasi paket *buffer management* dan *scheduling* dapat mengurangi dampak dari *congestion*. IETF (*Internet Engineering Task Force*) mempunyai standar model dan klasifikasi QoS diantaranya yaitu metode MPLS-Diffserv. Metode ini sudah banyak digunakan pada era sekarang dikarenakan memiliki kinerja yang lebih bagus dibanding MPLS-Intserv, karena MPLS-Diffserv merupakan suatu arsitektur jaringan IP QoS

yang mampu mengklasifikasi paket sesuai kebutuhan serta dapat melakukan proses *Dropping* saat terjadi penumpukan paket akibat proses QoS ini, dimana saat keadaan tertentu, paket-paket yang telah menumpuk akan dibuang atau di-drop (Putri, 2010). Mekanisme *dropping* pada umumnya diimplementasikan dengan mengatur fungsi paket *forwarding*. Dengan adanya *Differentiated Service Code Point* (DSCP) pada Diffserv, semua paket yang telah diprioritaskan biasanya di-drop pada saat antrian penuh, bisa diatasi dengan menggunakan algoritma *threshold* yang terdapat pada WRED (*Weighted Random Early Detection*).

Melihat dari permasalahan yang terjadi, maka pada penelitian ini akan diimplementasikan dan diuji, dimana jaringan IP MPLS-Diffserv dan mekanisme *dropping* menggunakan algoritma *threshold* akan dikombinasikan untuk melihat QoS dari *voice* (VoIP) dan *video streaming*. Dengan penambahan mekanisme *dropping* pada penggabungan *Differentiated Service* (Diffserv) dengan *Multi Protocol Label Switching* (MPLS) maka dapat memberikan performansi QoS yang terjamin dan memberikan solusi komunikasi yang lebih efektif dalam memenuhi kebutuhan pengguna.

Penelitian terdahulu yang dilakukan oleh Firman Nurdiansyah dkk. (2010) membahas mengenai solusi mekanisme layanan untuk memenuhi permintaan QoS dengan kombinasi dari kedua teknologi Diffserv dan MPLS menjamin layanan jaringan tulang punggung dengan teknologi *fast packet switching*. Dari hasil penelitian dalam algoritma penjadwalan, CBQ-WRR menunjukkan performansi lebih baik dibanding WF2Q dikarenakan untuk algoritma penjadwalan WF2Q lebih handal dalam menangani paket VoIP saja dan untuk algoritma penjadwalan CBQ-WRR lebih handal dalam menangani paket *video* serta dalam menangani paket data.

Penelitian kedua yang membahas mengenai algoritma *threshold*, berdasarkan penelitian yang dilakukan oleh Vivie Wulandari Sugiarto dkk. (2011) membahas mengenai simulasi dan implementasi algoritma *threshold*. Algoritma ini diimplementasikan ke dalam RED (*Random Early Detection*) dimana rata-rata ukuran paket yang berada dalam antrian dibandingkan dengan minimum *threshold* dan maximum *threshold*. Sebagai pembandingnya akan digunakan algoritma WRED (*Weighted Random Early Detection*). Dari hasil penelitian bahwa algoritma WRED menghasilkan nilai QoS yang lebih baik pada layanan *video*, karena algoritma WRED mempunyai keunggulan pada layanan dengan bobot paket yang besar seperti *voice* dan *video*, sedangkan algoritma penjadwalan RED menghasilkan nilai QoS yang lebih baik pada layanan dengan panjang paket berubah-ubah seperti HTTP.

Acuan penelitian terakhir oleh Misbahul Fajri (2016) yang membahas mengenai penanganan

kepadatan trafik data pada jaringan dengan mekanisme *Congestion Avoidance and Control* yang akan diimplementasikan metode AQM klasik FIFO (*Drop Tail*). Dengan mekanisme *Drop Tail* adanya pengiriman ulang pada paket yang telah di-drop karena terbatasnya *resource buffer*.

Berdasarkan acuan pustaka penelitian terdahulu, penulis memilih metode algoritma *Threshold* dengan menggunakan WRED menggantikan mekanisme *Drop Tail* sebagai *default Cisco* yang berada pada *Differentiated Service Code Point* (DSCP). Hal ini untuk menangani *congestion* pada *dropping* paket yang diimplementasikan pada trafik jaringan MPLS-Diffserv.

2. DASAR TEORI

2.1 Quality of Service

Quality of Service (QoS) digunakan untuk mengukur performansi dan tingkat kualitas pada jaringan IP untuk menyediakan tingkat jaminan performansi pada layanan yang berbeda-beda. Parameter-parameter QoS yang biasa digunakan untuk pengukuran performansi suatu jaringan antara lain *packet loss*, *delay*, *throughput* dan *jitter* [6].

Packet Loss merupakan suatu parameter QoS yang menunjukkan suatu jumlah total keseluruhan paket hilang atau tidak sampai ke *destinaton*, dikarenakan adanya *overload* atau *congestion* pada jaringan. Dalam suatu jaringan, *packet loss* diwajibkan mempunyai persentase yang kecil sesuai dengan standar (Ningsih, 2004).

Adapun standar *packet loss* dalam suatu implementasi jaringan versi TIPHON (*Telecommunications and Internet Protocol Harmonization Over Networks*) ditunjukkan pada Tabel 1.

Tabel 1. *Packet Loss*

Kategori	Packet Loss
Sangat Bagus	0 %
Bagus	3 %
Sedang	15 %
Jelek	25 %

(Sumber: TIPHON)

Persamaan perhitungan *packet loss* dengan persamaan :

$$Packet Loss = \frac{(\text{paket dikirim} - \text{paket diterima})}{\text{paket dikirim}} \times 100\% \dots\dots\dots (i)$$

Delay merupakan suatu parameter QoS yang menunjukkan jumlah waktu yang diperlukan paket untuk mencapai jarak dari *source* ke *destination*. Berberapa hal yang mempengaruhi *delay* adalah jarak, perangkat keras dan *congestion* (Ningsih,

2004). Standar *delay* menurut versi TIPHON (*Telecommunications and Internet Protocol Harmonization Over Networks*) ditunjukkan pada Tabel 2.

Tabel 2. Delay

Kategori	Delay
Sangat Bagus	< 150 ms
Bagus	150 - 300 ms
Sedang	300 - 450 ms
Jelek	> 450 ms

(Sumber: TIPHON)

Persamaan perhitungan *delay* dengan persamaan :

$$\text{Delay rata-rata} = \frac{\text{Total Delay}}{\text{Total Paket yang diterima}} \dots\dots\dots (ii)$$

Throughput merupakan parameter QoS yang menunjukkan suatu kecepatan rata-rata *bandwidth* yang sebenarnya, diukur dengan satuan waktu tertentu pada kondisi jaringan tertentu untuk melakukan pengiriman paket dengan ukuran tertentu juga (Ningsih, 2004). Hasil *throughput* diambil dari jumlah paket data yang dikirim dibagi dengan jumlah waktu yang diperlukan saat pengiriman paket data.

Tabel 3. Throughput

Kategori	Throughput
Sangat Bagus	100 %
Bagus	75 %
Sedang	50 %
Jelek	25 %

(Sumber: TIPHON)

Persamaan perhitungan *throughput* dengan persamaan :

$$\text{Throughput} = \frac{\text{Jumlah paket data yang dikirim}}{\text{Waktu pengiriman data}} \dots\dots\dots (iii)$$

Jitter merupakan suatu parameter QoS yang menunjukkan jumlah dari variasi-variasi *delay* pada transmisi paket pada jaringan. Hal ini disebabkan banyaknya variasi panjang antrian paket dalam waktu proses paket dan waktu penghimpunan ulang paket-paket (Ningsih, 2004).

Terdapat 4 kategori penurunan performansi *jitter* menurut TIPHON (*Telecommunications and Internet Protocol Harmonization Over Network*) ditunjukkan pada Tabel 4.

Tabel 4. Jitter

Kategori	Jitter
Sangat Bagus	0 ms
Bagus	0 - 75 ms
Sedang	75 - 125 ms
Jelek	125 - 225 ms

(Sumber: TIPHON)

Persamaan perhitungan *jitter* dengan persamaan :

$$\text{Jitter} = \frac{(\text{Total variasi delay})}{\text{Total paket yang diterima} - 1} \dots\dots\dots (iv)$$

2.2 Differented Service

Differented Service (Diffserv) merupakan metode dalam implementasi kebutuhan QoS yang menyediakan multi layanan dengan membagi *traffic* dan memperlakukan setiap kelas secara berbeda. Diffserv menentukan IP *precedence field* maupun *Differentiated Service Code Point field* (DSCP) pada awal ketika paket akan masuk atau melewati ke dalam jaringan untuk menentukan jalan mana yang harus dilalui dengan tahapan *packet classification*, *traffic shaping*, *traffic policing*, dan *queuing* berdasarkan aturan yang dibutuhkan (Mirah, 2016).

2.3 Multiprotocol Label Switching

Multiprotocol Label Switching (MPLS) merupakan metode paket *forwarding* meneruskan paket pada jaringan dengan menyisipkan suatu label pada IP, yang memungkinkan suatu *router* dapat meneruskan packet tanpa perlu melihat IP alamat tujuan namun hanya melihat label dari paket tersebut (Nurdiansyah dkk., 2010).

MPLS pernggabungan metode *switching* pada *layer 2* dan metode *routing* pada *layer 3* sehingga dapat menyelesaikan masalah dalam *scalability*, kecepatan dan *Quality of Service* (QoS). *Label switching* yang didapat dari *layer routing network*, semua paket akan dianalisa didalam *router* saat pertama kali masuk dalam jaringan, setelah itu mekanisme *label switching* melakukan *forwading* paket data yang diterima. Tujuan utama MPLS adalah untuk mempercepat pencarian rute paket dalam setiap *router* yang dilewati, sehingga membuat proses jaringan menjadi efisien dan proses transmisi paket lebih cepat (Petersson, 2005).

2.4 WRED

Weighted Random Early Detection (WRED) merupakan mekanisme probabilitas *dropping* berdasarkan pada panjang antrian dan mengadopsi proses *dropping* RED pada model *traffic* yang berbeda. WRED dapat menentukan profil *dropping* yang tidak sama pada suatu panjang antrian maupun

traffic namun dalam antrian yang sama (Floyd and Jacobson, 1993).

WRED dikonfigurasi dengan dua *policy-map*, pada packet saat akan masuk (*in profile*) dan pada packet saat keluar (*out profile*), sehingga router akan mengklasifikasi paket tersebut sebagai *in packet* maupun *out packet* saat kedatangan paket. Jika diketahui sebagai *in packet*, router akan melakukan perhitungan rata-rata panjang antrian *in packet* (*avg_in*) dan apabila diketahui sebagai *out packet*, router akan melakukan perhitungan rata-rata total dari antrian paket yang datang untuk semua *in packet* dan *out packet* (*avg_total*). Peluang *dropping* untuk *in packet* berdasarkan pada *avg_in* dan untuk *out packet* berdasarkan pada *avg_total* (Dhani dan Hafidudin, 2007).

Perhitungan probabilitas *dropping* pada algoritma *threshold* dijelaskan pada gambar 1.

```

For each packet arrival
  if it is an In packet
    calculate the average In queue size avg_in;
    calculate the average queue size avg_total;

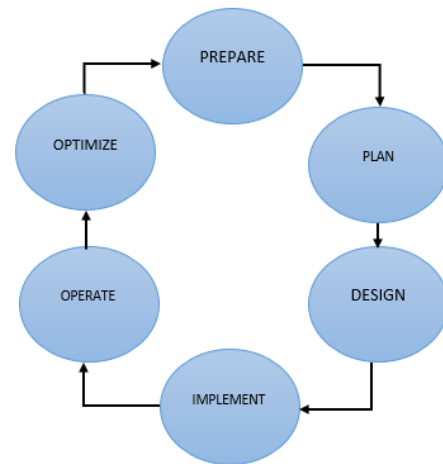
  If it is an In packet
    if min_in < avg_in < max_in
      calculate probability Pin;
      with probability Pin, drop this packet;
    else if max_in < avg_in
      drop this packet.
  if this is an Out packet
    if min_out < avg_total < max_out
      calculate probability Pout;
      with probability Pout, drop this packet;
    else if max_out < avg_total
      drop this packet.
  
```

Gambar 1 Algoritma *threshold*
(Sally Floyd and Van Jacobson, 1993)

Router akan melakukan *dropping* paket berdasarkan jenis paket yang datang, apabila *avg_in* (rata-rata panjang antrian masuk) berada diantara minimal *threshold in* dan maksimal *threshold in* maka paket akan di *dropping* berdasarkan besarnya MPD (*Mark Probability Denominator*) dan router akan melakukan *dropping* paket berdasarkan besarnya MPD yang akan keluar router apabila rata-rata panjang antrian keluar (*avg_out*) berada diantara minimal dan maksimal *threshold out*. Jika rata-rata total paket *in* dan *out* lebih besar dari maksimal *threshold* maka paket otomatis di-*dropping* secara acak.

3. METODE DAN PERANCANGAN

Metode desain jaringan yang digunakan pada penelitian ini adalah PPDIOO (*Prepare, Plan, Design, Implement, Operate and Optimize*) yang juga dikenal dengan *network lifecycle* merupakan sebuah metode penelitian yang dikembangkan oleh Cisco System.Inc (Wilkins, 2011). Berikut ini adalah tahap-tahapannya :



Gambar 2 Metode PPDIOO (Cisco Sysem.Inc)

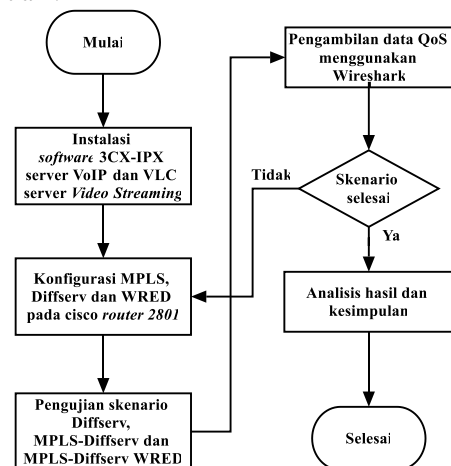
3.1 Prepare

Prepare adalah tahap awal pada proses penelitian untuk menyusun rencana kerja agar penelitian dapat terorganisir dengan baik. Tahap ini menyangkut hal-hal yang berhubungan dengan analisa pokok pembahasan seperti permasalahan yang muncul, analisa kebutuhan penelitian dalam

VoIP dan *video streaming*, baik itu dari segi *hardware* ataupun *software* dan analisa topologi jaringan yang akan dibangun.

3.2 Plan

Plan adalah tahap perencanaan jaringan yang dibuat baik segi *hardware* atau segi *software* yang dibutuhkan beserta skenario pengujian yang dilakukan pada penelitian. Berikut flowchart diagram yang menjelaskan tahapan perancangan penelitian :



Gambar 3 Flowchart perancangan penelitian

Gambar 3 menunjukkan proses perancangan penelitian dimulai dengan instalasi perangkat lunak yang digunakan yaitu 3CX-PBX sebagai server VoIP, 3CX Softphone, VLC Media Player sebagai server dan *client*, Wireshark untuk *capture* pengambilan data parameter QoS VoIP dan *video streaming* serta Putty untuk telnet router dan konfigurasi perangkat keras yaitu 2 laptop sebagai

server dan *client*, 5 *router* cisco seri 2801 dimana konfigurasi Diffserv, MPLS dan algoritma *threshold* diterapkan, 1 kabel Console serta 6 kabel LAN Cross-Over sebagai penghubung antar router dan antar host.

Skenario pengujian yang dilakukan pada penelitian ini sebagai berikut :

a. Skenario 1 Diffserv :

Melakukan VoIP dan *video streaming* pada jaringan MPLS dari PC *client* ke PC server dengan melewati 5 *router* dan hanya 2 *router* CE yang dikonfigurasi *Differentiated Service* (Diffserv), untuk ukuran file *video streaming* sebesar 15,40 MB. Proses pengambilan data VoIP dilakukan selama 90 detik dan pengambilan data *video streaming* dilakukan selama 60 detik. Masing-masing pengambilan dilakukan sebanyak 30 kali untuk mendapatkan nilai rata-rata parameter QoS.

b. Skenario 2 MPLS-Diffserv

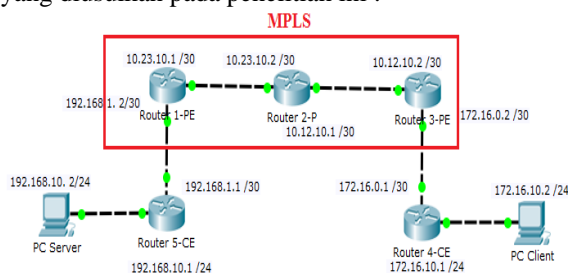
Melakukan VoIP dan *video streaming* pada jaringan MPLS dari PC *client* ke PC server dengan melewati 5 *router* yang dikonfigurasi *Differentiated Service* (Diffserv), untuk ukuran file *video streaming* sebesar 15,40 MB. Proses pengambilan data VoIP dilakukan selama 90 detik dan pengambilan data *video streaming* dilakukan selama 60 detik. Masing-masing pengambilan dilakukan sebanyak 30 kali untuk mendapatkan nilai rata-rata parameter QoS.

c. Skenario 3 MPLS-Diffserv WRED

Melakukan VoIP dan *video streaming* pada jaringan MPLS dari PC *client* ke PC server dengan melewati 5 *router* yang dikonfigurasi *Differentiated Service* (Diffserv) dan penambahan algoritma *threshold* pada *queue* WRED di *router* R4-CE dan R5-CE, untuk ukuran file *video streaming* sebesar 15,40 MB dengan bandwidth 1024 kbps dan VoIP dengan bandwidth 128 kbps. Proses pengambilan data VoIP dilakukan selama 90 detik dan pengambilan data *video streaming* dilakukan selama 60 detik. Masing-masing pengambilan dilakukan sebanyak 30 kali untuk mendapatkan nilai rata-rata parameter QoS.

3.3 Design

Design merupakan tahapan membuat suatu topologi jaringan yang akan dilakukan pada penelitian. Berikut ini gambar rancangan topologi yang diusulkan pada penelitian ini :



Gambar 4 Design topologi jaringan MPLS-Diffserv

Gambar 4 merupakan topologi yang digunakan untuk implementasi jaringan MPLS-Diffserv pada penelitian ini. Sistem MPLS dengan memiliki arsitektur sendiri walaupun mempunyai proses yang begitu kompleks. Perangkat yang dibutuhkan pada jaringan MPLS antara lain adalah: *router* P (*Provider*), *router* PE (*Provider Edge*) dan *router* CE (*Customer Edge*) (Petersson, 2005).

Pada topologi jaringan MPLS domain terdapat 3 *router* yaitu *router* R2-P sebagai *router* provider (*core*) hanya untuk *swapping* label informasi, *router* R1-PE dan *router* R3-PE sebagai *router* *Provider Edge* perangkat distribusi yang berfungsi untuk menghubungkan jaringan MPLS (*routing* OSPF) dengan jaringan luar (*routing* BGP) yaitu *router* R4-CE dan R5-CE untuk menghubungkan *host* dengan jaringan MPLS.

3.4 Implement

Implement merupakan tahap yang akan dilakukan sesuai pada topologi tahap *design*. Mulai dari konfigurasi laptop 1 sebagai server VoIP menggunakan perangkat lunak 3CX-PBX dan server *video streaming* menggunakan perangkat lunak VLC Media Player, konfigurasi laptop 2 sebagai *client* VoIP menggunakan perangkat lunak 3CX Softphone dan *client* *video streaming* menggunakan perangkat lunak VLC Media Player. Konfigurasi *router* cisco 2801 diimplementasikan sebagai jaringan MPLS-Diffserv dan mekanisme *dropping* Algoritma *Threshlod*. Pada R1-PE dan R3-PE *router* dikonfigurasi sebagai MPLS domain dan Diffserv dengan *routing* menggunakan OSPF *routing*. Pada *router* R2-P digunakan sebagai *core* MPLS dan untuk konfigurasi *router* R4-CE serta R5-CE hanya sebagai Diffserv serta mekanisme algoritma *threshold* yang berada pada WRED diimplementasikan pada DSCP Diffserv.

3.5 Operate

Operate merupakan tahap melakukan percobaan skenario yang telah disiapkan. Percobaan yang dilakukan dengan mengalirkan *traffic* VoIP dan *video streaming* pada jaringan MPLS-Diffserv dengan menggunakan protokol RTP dari *source* ke *destination* menggunakan *software* 3CX untuk VoIP dan *software* VLC Media Player untuk *video streaming*. Pada host laptop 2 melakukan *capture* wireshark untuk menguji QoS dari skenario yang sudah ditetapkan dan pemecahan masalah yang timbul selama proses pengambilan data.

3.6 Optimize

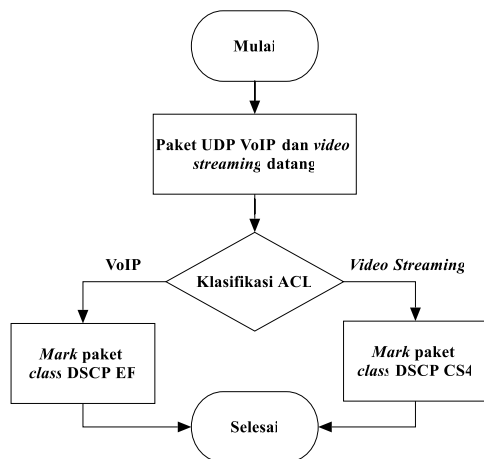
Optimize merupakan tahapan optimalisasi setelah melakukan analisa pada tahap *operate*, tahap ini dapat menyebabkan perlunya desain ulang jaringan apabila terlalu banyak masalah jaringan yang timbul atau kinerja tidak sesuai dengan yang diharapkan.

4. HASIL PEMBAHASAN

Pada Penelitian ini digunakan perangkat lunak 3CX Softphone untuk melakukan VoIP selama 90 detik dan satu *file video* dengan format MPEG-4 dan *codec* H.264 berukuran 15,40 MB berdurasi 60 detik untuk *video streaming*. Kemudian dilakukan pengujian sebanyak 30 kali VoIP dan *video streaming* pada jaringan Diffserv, jaringan MPLS-Diffserv dan jaringan MPLS-Diffserv dengan mekanisme *dropping* algoritma *threshold* WRED. Pengujian pertama dilakukan panggilan VoIP dan *streaming video* pada jaringan Diffserv. Kode konfigurasi penentuan *marking* paket untuk *Differentiated Service Code Point* (DSCP) pada router R4-CE atau R5-CE dapat dilihat pada Kode Konfigurasi 1.

Kode Konfigurasi 1 Kode konfigurasi Diffserv

```
R4-CE(config)#access-list 101 permit
udp any any range 40000 45000
R4-CE(config)#access-list 102 permit
udp any any range 537 7000
R4-CE(config)#class-map VOIP
R4-CE(config-cmap)# match access-group
101
R4-CE(config)#class-map VIDEO
R4-CE(config-cmap)# match access-group
102
R4-CE(config)#policy-map DSCP_DIFFSERV
R4-CE(config-pmap)#class VOIP
R4-CE(config-pmap-c)#bandwidth 128
R4-CE(config-pmap-c)#set dscp ef
R4-CE(config-pmap-c)#class VIDEO
R4-CE(config-pmap-c)#bandwidth 1024
R4-CE(config-pmap-c)#set dscp cs4
```



Gambar 5. Flowchart proses paket Diffserv

Pada router cisco, *access-list* (ACL) merupakan metode selektivitas terhadap paket data yang akan dikirimkan ke tujuan. Hanya paket yang memiliki kriteria sesuai aturan ACL yang diperbolehkan (*permit*) melewati gerbang keamanan. *Class-map* digunakan untuk mengklasifikasikan *traffic* jaringan yang masuk atau melewati router berdasarkan aturan *match access-list* yang telah ditentukan dan *policy-map* merupakan serangkaian

tindakan yang akan diproses pada *traffic* yang telah diklasifikasi melalui *class-map* [9]. Untuk protokol UDP dengan port *range* 40000 hingga 45000 akan diklasifikasi sebagai ACL untuk VoIP dan untuk protokol UDP dengan port *range* 537 hingga 7000 akan diklasifikasi sebagai ACL untuk *video streaming*. *Class-map* berdasarkan *match* yang sesuai dengan ACL yang telah diklasifikasi. Kemudian *policy-map* digunakan mengatur *action* terhadap *class-map* dan untuk mengatur *bandwidth* minimal yang diprioritaskan pada paket yang sesuai dengan *class-map*. Ada 2 *class-map* dalam konfigurasi Diffserv, *class voip* untuk VoIP *marking* diatur sebagai DSCP EF (*Expedited Forwarding*) mendapatkan *bandwidth* minimal 128 kbps dan *class video* untuk *video streaming marking* diatur sebagai DSCP CS4 (*Expedited Forwarding*) mendapatkan *bandwidth* minimal 1024 kbps. Pengujian kedua dilakukan panggilan VoIP dan *streaming video* pada jaringan MPLS-Diffserv. Pengujian kedua dilakukan panggilan VoIP dan *streaming video* pada jaringan MPLS-Diffserv. Kode konfigurasi MPLS pada router R2-P dapat dilihat pada Kode Konfigurasi 2.

Kode Konfigurasi 2 Kode konfigurasi MPLS

```
R2-P(config)#ip cef
R2-P(config)#mpls ip
R2-P(config)#mpls label protocol ldp
R2-P(config)#interface FastEthernet0/0
R2-P(config-if)#mpls label protocol ldp
R2-P(config-if)#mpls ip
R2-P(config-if)#exit
R2-P(config)#interface FastEthernet0/1
R2-P(config-if)#mpls label protocol ldp
R2-P(config-if)#mpls ip
```

Ip cef pada konfigurasi MPLS berfungsi untuk menghidupkan fitur *ip cisco express forwarding* sebagai fitur untuk mempercepat *switching*, MPLS IP untuk mengaktifkan protokol MPLS pada sebuah ip dan *mpls label protocol ldp* digunakan untuk mengaktifkan label LDP (*Label Distribution Protocol*) mendistribusikan informasi pada label ke setiap LSR (*Label Swicthing Router*) [10]. Konfigurasi MPLS diterapkan pada *interfaces* yang terkoneksi antar router jaringan MPLS. Pengujian ketiga dilakukan panggilan VoIP dan *streaming video* pada jaringan MPLS-Diffserv WRED.

Kode Konfigurasi 3 Kode konfigurasi algoritma *threshold* WRED

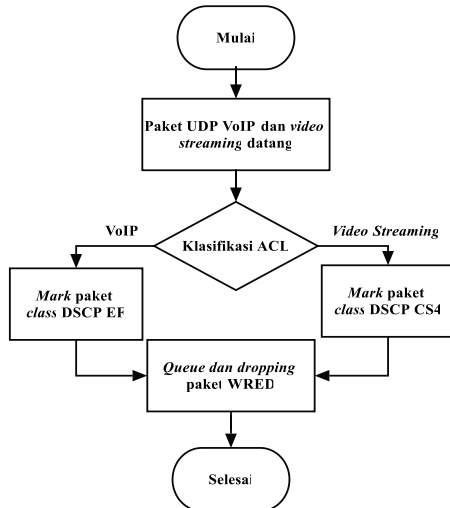
```
R5-CE(config)#class-map m_cs4
R5-CE(config-cmap)#match dscp cs4
R5-CE(config)#class-map m_ef
R5-CE(config-cmap)#match dscp ef
R5-CE(config)#policy-map WRED
R5-CE(config-pmap)#class m_ef
R5-CE(config-pmap-c)#bandwidth 128
```

```

R5-CE(config-pmap-c)#random-detect
dscp-based
R5-CE(config-pmap-c)#random-detect dscp
46 30 50 10
R5-CE(config-pmap)#class m_cs4
R5-CE(config-pmap-c)#bandwidth 1024
R5-CE(config-pmap-c)#random-detect
dscp-based
R5-CE(config-pmap-c)#random-detect dscp
32 20 60 10

```

Mengaktifkan WRED pada *base DSCP field* di *policy-map* digunakan perintah *random-detect dscp-based*, untuk *class-map* VoIP *dropping marking* diatur sebagai DSCP 46 atau EF (*Expedited Forwarding*) prioritas *bandwidth* minimal sebesar 128 kbps dan dengan nilai minimal *threshold* 30 batas minimal antrian paket masuk dalam *drop flag*, nilai maksimal *threshold* 50 batas maksimal antrian paket akan di-drop secara acak dan nilai MPD (*Mark Probability Denominator*) 10 ketika nilai maksimal *threshold* terpenuhi maka 1 dari 10 paket akan di-drop. *Class-map* video *Streaming* diatur sebagai DSCP 32 atau CS4 prioritas *bandwidth* minimal sebesar 1024 kbps dengan nilai minimal *threshold* 20 batas minimal antrian paket masuk dalam *drop flag*, nilai maksimal *threshold* 60 batas maksimal antrian paket akan di-drop secara acak dan nilai MPD (*Mark Probability Denominator*) 10. *Flowchart* proses paket melewati *router* konfigurasi Diffserv dan WRED sebagai berikut :



Gambar 5. Flowchart proses paket

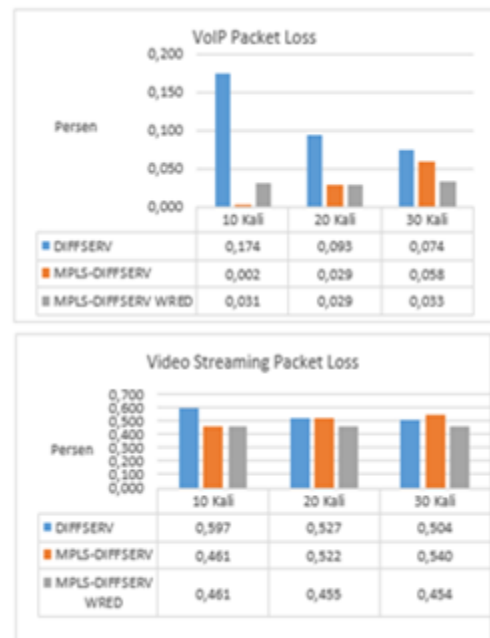
Pada gambar 5 menunjukkan bahwa saat paket VoIP atau *video streaming* datang ke *router*, tahap pertama yang diproses *router* adalah mengklasifikasi paket sesuai dengan aturan *access-list* apabila paket VoIP maka akan diperbolehkan masuk ke dalam *class-map* EF dan apabila paket *video streaming* maka akan diperbolehkan masuk ke dalam *class-map* CS4 untuk dioprioritaskan. Setelah itu paket yang sudah terklasifikasi sebagai paket VoIP dan *video streaming*, akan diproses sesuai kebijakan pada *policy-map* EF akan diberikan

bandwidth minimal sebesar 128 kbps dan CS4 diberikan *bandwidth* minimal 1024 kbps. Tahap terakhir apabila terjadi penumpukan paket pada antrian (*queue*) maka proses *dropping* paket akan dilakukan sesuai dengan algoritma *threshold* dikonfigurasi pada WRED sebelum paket keluar dari *router*.

Setelah konfigurasi *router* sesuai skenario pengujian selesai, dilakukan proses pengambilan data. Proses pengambilan data dengan *me-capture* menggunakan *wireshark* yang ada pada laptop 2, kemudian dilakukan analisis untuk parameter *Quality of Service* (QoS) yaitu *packet loss*, *delay*, *throughput* dan *jitter*.

4.1 Packet Loss

Berikut ini merupakan hasil pengukuran parameter QoS *packet loss* berdasarkan skenario pengujian yang telah ditentukan sebelumnya.



Gambar 6. Hasil pengukuran *Packet Loss*

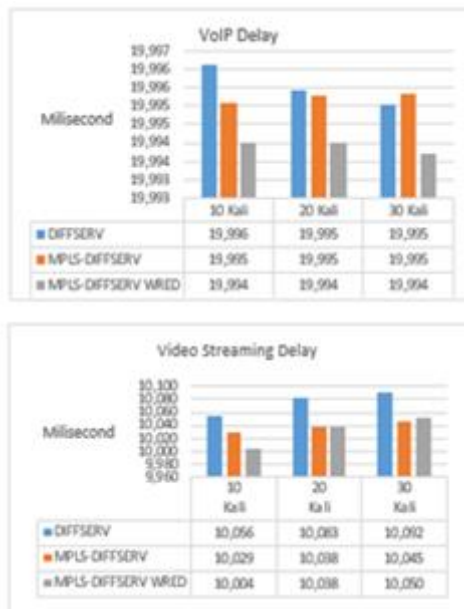
Dilihat dari data pada Gambar 7 *packet loss* layanan VoIP dengan semua skenario tergolong kategori sangat bagus menurut standar TIPHON yaitu dibawah 1% namun untuk *packet loss* dengan skenario Diffserv mempunyai nilai tertinggi. Hal ini disebabkan karena proses *marking* paket VoIP sebagai DSCP *field* EF (*Expedited Forwarding*) hanya terjadi pada *router* CE yang dikonfigurasi sebagai Diffserv dan saat paket melewati jaringan MPLS *router* P dan PE yang tidak dikonfigurasi sebagai Diffserv, maka tidak terjadi *marking* paket yang menyebabkan paket tidak diprioritaskan sehingga tidak menjamin keutuhan paket sampai pada tujuan.

Packet loss pada layanan *video streaming* metode MPLS-Diffserv WRED mempunyai *packet loss* yang paling rendah. Hal ini disebabkan adanya

congestion avoidance menggunakan algoritma *threshold WRED* yang menangani *dropping* paket saat penumpukan paket pada antrian.

4.2 Delay

Berikut ini merupakan hasil pengukuran parameter QoS *delay* berdasarkan skenario pengujian yang telah ditentukan sebelumnya.



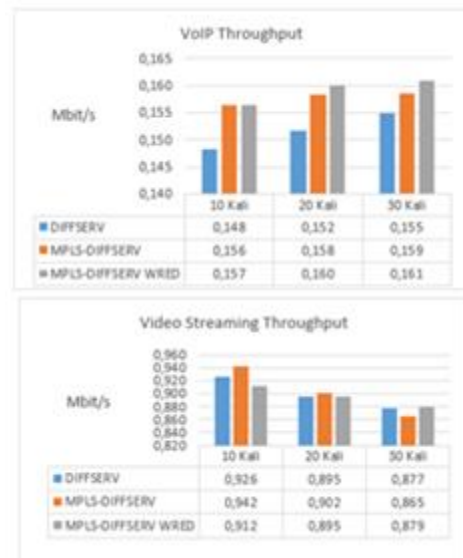
Gambar 7 Hasil pengukuran *delay*

Dilihat dari data grafik pada Gambar 8 *delay* layanan VoIP dengan semua skenario tergolong indeks sangat bagus menurut standar TIPHON yaitu dibawah 150 ms dan semakin banyak percobaan dilakukan maka *delay* pada semua metode semakin turun nilainya namun untuk rata-rata *delay* dengan metode Diffserv mempunyai nilai tertinggi. Sama halnya saat terjadi pada *packet loss*, pada saat paket VoIP melewati jaringan MPLS router P dan PE tanpa konfigurasi Diffserv, maka tidak terjadi proses *marking* paket yang menyebabkan paket tidak diprioritaskan sehingga membuat nilai *delay* menjadi tinggi.

Delay pada layanan *video streaming* baik metode Diffserv, MPLS-Diffserv maupun MPLS-Diffserv WRED semakin banyak dilakukan percobaan maka nilai *delay* semakin naik. Hal ini disebabkan adanya antrian paket *video streaming*. Skenario MPLS-Diffserv dan MPLS-Diffserv WRED mempunyai rata-rata nilai *delay* yang lebih rendah dikarenakan adanya *marking* paket pada DSCP field Diffserv pada semua router sehingga paket *video streaming* lebih diprioritaskan untuk sampai ke *destination*.

4.3 Throughput

Berikut ini merupakan hasil pengukuran parameter QoS *throughput* berdasarkan skenario pengujian yang telah ditentukan sebelumnya.

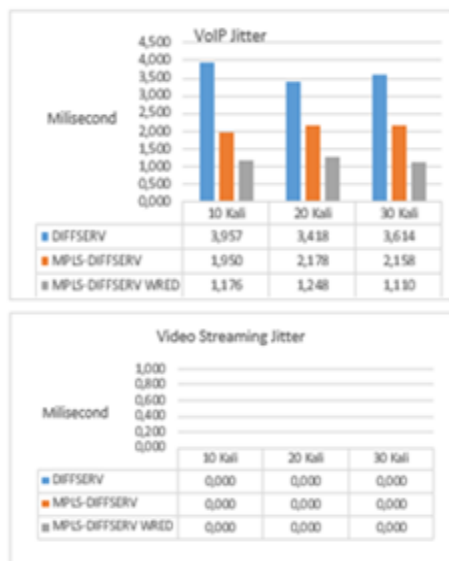


Gambar 8 Hasil pengukuran *throughput*

Dilihat dari data grafik pada Gambar 9 bisa dilihat bahwa nilai *throughput* layanan VoIP ketika semakin banyak percobaan dilakukan nilai *throughput* untuk semua metode semakin naik dan nilai *throughput* layanan *video streaming* ketika semakin banyak percobaan dilakukan nilai *throughput* untuk semua metode semakin turun. Hal ini berbanding terbalik dengan nilai *delay video streaming* yang semakin lama semakin naik, dapat disimpulkan bahwa nilai *throughput* berbanding terbalik dengan nilai *delay*. Menurut standar TIHPON untuk semua skenario tergolong indeks sangat bagus yaitu 100% dari *bandwidth* yang dibutuhkan dilihat dari *traffic* VoIP yang membutuhkan *bandwidth* 128 kbps dan *video streaming* yang membutuhkan *bandwidth* 1024 kbps. Pada skenario MPLS-Diffserv dan MPLS-Diffserv WRED mempunyai perbedaan nilai *throughput* yang tidak terlalu signifikan. Hal ini disebabkan karena pada *policy-map bandwidth* paket VoIP pada DSCP field EF dan *bandwidth* paket *video streaming* pada DSCP filed CS4 telah diprioritaskan, sehingga memaksimalkan *bandwidth* yang ada untuk paket *traffic* VOIP dan *video streaming*.

4.4 Jitter

Berikut ini merupakan hasil pengukuran parameter QoS *jitter* berdasarkan skenario pengujian yang telah ditentukan sebelumnya.



Gambar 9 Hasil pengukuran *jitter*

Dilihat dari data grafik pada Gambar 10 *jitter* layanan VoIP dengan semua metode tergolong indeks bagus menurut standar TIPHON yaitu dibawah 75 ms dan sama halnya dengan hasil *delay*, semakin banyak percobaan *jitter* pada semua metode semakin turun nilainya. Nilai *jitter* pada metode MPLS-Diffserv dan MPLS-Diffserv WRED mempunyai selisih yang tidak terlalu signifikan pada setiap percobaan. Hal ini menunjukkan bahwa proses *marking* paket atau pemberian prioritas pengiriman pada paket dapat mengurangi tingginya *jitter* yang timbul akibat adanya antrian paket pada semua *router*. Berbeda dengan skenario Diffserv yang hanya memberikan prioritas pada *router* CE namun saat melewati jaringan MPLS yang *router*nya tidak dikonfigurasi Diffserv maka paket tidak menjadi prioritas lagi dan memungkinkan variasi *delay* naik.

Pada layanan *video streaming* dengan semua skenario tergolong indeks sangat bagus menurut standar TIPHON yaitu 0 ms. Pada grafik “*Video Streaming Jitter*” nilai *jitter* di semua skenario pengujian adalah 0 ms, hal ini disebabkan *video streaming* menggunakan protokol RTP (*Real-Time Transfer Protocol*) pada UDP karena protokol TCP tidak mendukung pada pengiriman data *real-time*. RTP didesain untuk mengkompensasi *jitter* yang terjadi pada jaringan dikarenakan apabila semakin besar *jitter*, semakin besar pula peluang terjadinya *congestion* pada jaringan.

5. SIMPULAN

Berdasarkan hasil pengujian pada semua skenario dengan parameter QoS (*packet loss*, *delay*, *throughput* dan *jitter*), dapat disimpulkan bahwa ketiga skenario Diffserv, MPLS-Diffserv dan MPLS-Diffserv WRED mampu mempertahankan QoS dari komunikasi VoIP maupun *video streaming*. Dari parameter *delay* dan *throughput*, ketiga

skenario tersebut mampu menghasilkan *delay* dan *throughput* dalam kategori sangat bagus berdasarkan standar TIPHON. Untuk parameter nilai *packet loss* dan *jitter*, skenario MPLS-Diffserv WRED memiliki nilai *packet loss* dan *jitter* paling baik dalam layanan VoIP maupun *video streaming*. Skenario penambahan WRED sebagai algoritma *threshold* pada jaringan MPLS-Diffserv memberikan pengaruh yang signifikan dalam menangani *dropping* paket. Dilihat dari hasil nilai parameter QoS dalam layanan VoIP yaitu mampu mengurangi *packet loss* hingga 43,1%, mengurangi *delay* 0,005%, memaksimalkan *throughput* 1,26% dan mampu mengurangi *jitter* 48,56%. Dalam layanan *video streaming* mampu mengurangi *packet loss* hingga 15,93% dan memaksimalkan *throughput* 1,6% dari sebelum menerapkan algoritma *threshold*.

Jadi, dapat disimpulkan bahwa MPLS-Diffserv WRED pada penelitian ini memiliki kinerja paling bagus dalam layanan VoIP dan *video streaming*.

6. SARAN

Pada penelitian selanjutnya perlu dilakukan penyajian data QoS lebih detail, baik pada Diffserv maupun MPLS dengan penambahan proses *error identification* maupun *correction* paket data yang terjadi pada *traffic*. Penambahan ini berkaitan dengan analisa yang lebih mendalam terhadap transmisi data yang terjadi.

7. DAFTAR PUSTAKA

- DHANI, A., HAFIDUDIN, ARIF R. 2007. Analisa Perbandingan Performansi Skema Multi-Level Red Untuk Differentiated Services Di Internet, *Seminar Nasional Aplikasi Teknologi Informasi 2007 (SNATI 2007)*, ISSN: 1907-5022.
- MIRAH, CHRISTIAN YORDAN. Analisis Perbandingan Quality of Service (QoS) Antara Metode Differentiated Service (DiffServ) dan Metode Multiprotocol Label Switching (MPLS). *Repository Jurnal Universitas Kristen Satya Wacana*.
- MISBAHUL, F. 2016. Simulasi Antrian Paket Data Jaringan dengan Mekanisme Drop Tail. *Repository Jurnal Universitas Mercu Buana*. Vol. 8 No. 2.
- SALLY, F., JACOBSON, V. 1993. *Random Early Detection Gateways for Congestion Avoidance*, University of California, USA.
- NINGSIH, KURNIA, Y., DKK. 2004, *Analisis Quality Of Service (QoS) pada Simulasi Jaringan Multiprotocol label Switching Virtual Private Network (Mpls Vpn)*, JETri, vol.3, no.2, pp.33-48.

- NURDIANSYAH, F., NIKEN, D. C., BAYU, E. 2010. *Analisis QoS Diffserv pada Jaringan Mpls Menggunakan Algoritma WF2Q dan CBQ-WRR*. Bandung: Telkom University.
- PETERSSON, JOHAN, M. O. 2005. *MPLS Based Recovery Mechanism*. Master Thesis, University of Oslo, USA.
- MARSELI, P. E. 2010. *Penerapan metode QoS pada jaringan traffic yang padat*. Sriwijaya: Teknik Informatika Bilingual Fakultas Ilmu Komputer University Sriwijaya.
- WILKINS, S. 2011. *Cisco's PPDIIO Network Cycle.*, Indianapolis: CISCO Press.
- SUGIARTO, V. W., SOFIA, N. H., MULYANA, A. 2011. *Simulasi dan Analisis Manajemen Buffer pada Jaringan Metro Ethernet dengan Algoritma threshold*. Bandung: Telkom University.
- JHON, W., SONS LTD. 2005. *Network Congestion Control Manage Internet Traffic*, British Library Cataloguing. ISBN-13 978-0-470-02528-4.