

Implementasi *Quality of Service* pada Manajemen Jaringan di SMKN Negeri 3 Bangkalan Menggunakan Teknologi *OpenDaylight Controller* Menggunakan Metode *HTB*

Muhlis Tahir¹, Septyan Hendra Sugara², Nova Estu Harsiwi³

^{1,2} Program Studi Pendidikan Informatika, Universitas Trunojoyo Madura, Bangkalan, Indonesia

³ Program Studi Pendidikan Guru Sekolah Dasar, Universitas Trunojoyo Madura, Bangkalan, Indonesia

email: ¹muhlis.tahir@trunojoyo.ac.id, ²170631100035@student.trunojoyo.ac.id, ³nova.harsiwi@trunojoyo.ac.id

DOI: <https://doi.org/10.21107/edutic.v9i1.17094>

Diterima: 30 Juni 2022 | Direvisi: 2 Oktober 2022 | Diterbitkan : 15 Nopember 2022

Abstrak

Quality of Service (QoS) pada jaringan TCP/IP difokuskan pada pengalokasian bandwidth yang mengalir sesuai dengan kebutuhan layanan pada jaringan. Hal ini bertujuan untuk meningkatkan kualitas layanan pada jaringan dengan memberikan jaminan bandwidth untuk setiap layanan yang berjalan. Peningkatan jumlah pengguna internet pada jaringan kampus secara otomatis akan mempengaruhi kualitas layanan pada jaringan tersebut. Oleh karena itu diperlukan suatu sistem yang dapat meningkatkan kualitas layanan jaringan dengan melakukan pembatasan bandwidth untuk setiap layanan yang berjalan. Hierarchical Token Bucket (HTB) merupakan salah satu metode dalam alokasi bandwidth. HTB bertugas membatasi bandwidth dan mengatur antrian paket berdasarkan prioritas. Penerapan metode HTB dilakukan dengan menggunakan teknologi Software-Defined Networking (SDN). SDN adalah teknologi jaringan yang menggunakan OpenFlow sebagai protokolnya. OpenDaylight adalah salah satu pengontrol berbasis SDN yang mendukung protokol OpenFlow. Metode HTB terhubung ke pengontrol OpenDaylight menggunakan fitur API lainnya. Hasil pengujian menunjukkan bahwa metode HTB dapat berjalan dengan baik dalam membatasi bandwidth dan mampu meningkatkan kualitas layanan jaringan dengan memberikan jaminan bandwidth pada setiap layanan yang telah ditentukan.

Kata Kunci: *QoS, Software-Defined Networking, Hierarchical Token Bucket, OpenDaylight*

Abstract

Quality of Service (QoS) on a TCP/IP network is focused on allocating bandwidth that flows according to service requirements on the network. It aims to improve the quality of service on the network by providing a guarantee of bandwidth for each service running. Increasing the number of Internet users on the campus network will automatically affect the quality of service on the network. Therefore required a system that can improve the quality of network services by doing bandwidth limitation for every service running. Hierarchical Token Bucket (HTB) is one of the methods in bandwidth allocation. HTB is tasked with limiting bandwidth and managing packet queues based on priority. Application of HTB method is done using Software-Defined Networking (SDN) technology. SDN is a network technology that uses OpenFlow as its protocol. Open Daylight is one of the SDN-based controllers that support the OpenFlow protocol. The HTB method is connected to the Open Daylight controller using the rest API feature. The results of the test show that the HTB method can run well in limiting bandwidth and able to improve the quality of network services by providing bandwidth guarantee on each service that has been determined.

Keywords: *QoS, Software-Defined Networking, Hierarchical Token Bucket, OpenDaylight*



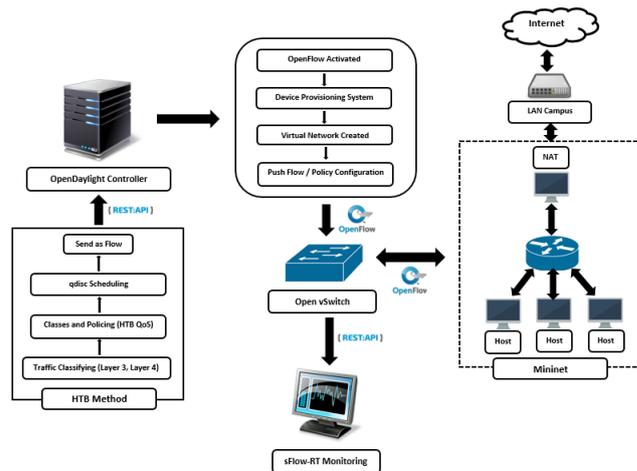
© Author (s)

PENDAHULUAN

Tingkat pertumbuhan trafik internet saat ini terus meningkat berdasarkan semakin banyaknya pengguna. Salah satu sumber terkemuka yang memantau tingkat lalu lintas internet menunjukkan bahwa tingkat pertumbuhan saat ini adalah 40 hingga 50%, oleh karena itu kemampuan jaringan, khususnya kemampuan transmisi, routing dan switching perlu menyesuaikan arus lalu lintas agar pertumbuhan internet yang stabil dapat dipertahankan. QoS (Quality of Service) adalah metode pengukuran tentang seberapa baik jaringan dan merupakan usaha untuk mendefinisikan karakteristik dan sifat suatu layanan. QoS digunakan untuk mengukur sekumpulan atribut kinerja yang telah di spesifikasikan dan biasanya diasosiasikan dengan suatu layanan(Ahdan et al., 2018). QoS dalam suatu jaringan adalah suatu permintaan untuk melakukan suatu kinerja yang diperlukan untuk mentransfer paket-paket IP dalam suatu jaringan(Ahdan et al., 2018)(Antodi et al., 2017).

Meningkatnya jumlah Siswa SMKN 3 Bangkalan serta makin mudahnya mendapatkan sarana untuk menghubungkan ke layanan internet memungkinkan penggunaan internet di kalangan akademis kampus akan semakin meningkat. Perkembangan pemakaian internet di kampus secara otomatis akan mempengaruhi kualitas layanan pada jaringan tersebut. Penurunan kualitas layanan pada jaringan ditandai dengan melambatnya proses pengiriman paket pada suatu service. Hal tersebut terjadi akibat sibuknya lalu lintas pada jaringan dalam menggunakan bandwidth serta tidak adanya jaminan bandwidth pada setiap service yang berjalan, sehingga terjadi perebutan bandwidth antar service(Antodi et al., 2017). Akibat penurunan kualitas layanan pada jaringan di sekolah tentu akan mengganggu proses kenyamanan pembelajaran siswa di sekolah, mengingat pentingnya fasilitas internet dalam kegiatan akademik. Penurunan kualitas layanan pada jaringan komputer dapat bervariasi akibat beberapa masalah, seperti halnya masalah bandwidth, latency, dan jitter yang dapat membuat efek yang cukup besar bagi banyak service aplikasi. Sebagai contoh, komunikasi suara (seperti VoIP atau IP Telephony) serta video streaming dapat membuat pengguna frustrasi ketika paket data aplikasi tersebut dialirkan di atas jaringan dengan bandwidth yang tidak cukup, dengan latency yang tidak dapat diprediksi, atau jitter yang berlebih. Fitur Quality of Service (QoS) ini dapat menjadikan bandwidth, latency, dan jitter dapat diprediksi dan dicocokkan dengan kebutuhan service dari aplikasi yang digunakan di dalam jaringan tersebut. Semakin besar sebuah jaringan komputer mengakibatkan semakin sulit seorang administrator untuk mengontrol jaringan tersebut. Sebab, arsitektur jaringan tradisional menggunakan switch konvensional seperti RoterBoard Mikrotik dan lain - lain. Padatnya pengguna akses internet maka diperlukan suatu akses yang lancar dan memiliki kestabilan dan kecepatan yang tinggi, bahkan membutuhkan bandwidth yang besar untuk memikul data yang besar, seperti data gambar, video, suara, file, steaming, dan akses langsung live. Masalah akan berbeda jika besar bandwidth yang dibeli oleh penyedia jasa akses internet pada network internet area memiliki keterbatasan dan tidak adanya management bandwidth yang optimal, tidak adanya pembagian prioritas penggunaan akses internet, Masalah dapat muncul dan terjadi seperti alokasi bandwidth akan dominan pada user/pelanggan internet tertentu saja sehingga user lain kesulitan akses internet(Aprianto Budiman et al., 2020). Terjadi juga user prioritas sulit mendapatkan akses karena kecepatan yang terbatas, bahkan website tertentu yang penting tidak dapat diaksesnya. Pembagian bandwidth sebelumnya sudah banyak mencoba seperti teknik bagi rata dengan model jatah dan memiliki kelemahan jika terdapat user tidak aktif bandwidth menjadi tidak terpakai, teknik lain menerapkan optimasi penggunaan bandwidth walau user tidak terpakai akan dialihkan secara merata ke user yang aktif namun memiliki kelemahan besar bandwidth yang teralokasi jika terbiasa jumlah user yang aktif sedikit maka user merasakan kecepatan yang tinggi dan jika user yang aktif banyak maka user sering terjadi komplain karena merasakan akses yang lambat. Penerapan teknik perpaket membatasi streaming, download bahkan sampai perekstensi hasilnya banyak komplain karena user yang membutuhkan apa yang diinginkan dirasa kurang cepat atau stabil dan kadang sering tidak bisa diakses(Antodi et al., 2017).

sistemnya sendiri dengan fasilitas rest API sebagai media interkoneksi dengan sistem pada controller. Dalam kasus ini penulis memanfaatkan rest API berbasis Java untuk menerapkan sistem bandwidth limitation dan Packet queues menggunakan metode Hierarchical Token Bucket (HTB) dengan controller OpenDaylight Boron SR4. Teknologi ini menggunakan sebuah standar protokol untuk interkoneksi antar perangkat jaringan yaitu OpenFlow(IGO., 2012). Dalam kasus ini penulis menggunakan OpenFlow versi 1.3. Saat ini teknologi virtualisasi sangat berkembang, SDN sangat mendukung teknologi ini. Teknologi virtualisasi memiliki banyak kelebihan seperti irit cost dan koneksi yang lebih stabil karena tanpa kabel fisik. Untuk itu penulis menerapkan teknologi virtualisasi dalam penelitian ini. Dalam kasus ini penulis menggunakan switch virtual yaitu open vSwitch versi 2.0.2 dan menggunakan infastruktur jaringan virtual yaitu Mininet versi 2.2.2 yang akan diterapkan pada jaringan di SMKN 3 Bangkalan yang diharapkan mampu meningkatkan kualitas layanan jaringan dengan memberikan jaminan ketersediaan bandwidth setiap service yang pada jaringan tersebut. Pada Gambar 2. dijelaskan arsitektur secara umum yang digunakan penulis pada penelitian ini.



Gambar 2. Arsitektur Umum Sistem

Alur Kerja Sistem

Penerapan Quality of Service (QoS) pada penelitian ini menggunakan teknologi Software-defined Networking (SDN) dan melibatkan beberapa tahapan seperti yang telah digambarkan pada arsitektur umum Gambar 2. Tahap pertama controller dan OpenFlow diaktifkan pada setiap perangkat jaringan. Ketika controller dan OpenFlow diaktifkan dalam jaringan, controller akan mengenali semua perangkat yang ada dalam jaringan (Device Provisioning System) kemudian controller akan memisahkan control plane dan data plane dari setiap switch, dan seluruh konfigurasi seperti route flow mutlak diambil alih oleh controller(Melyana & Indriyani, 2016). Tahap pertama controller dan OpenFlow diaktifkan pada setiap perangkat jaringan. Ketika controller dan OpenFlow diaktifkan dalam jaringan, controller akan mengenali semua perangkat yang ada dalam jaringan (Device Provisioning System) kemudian controller akan memisahkan control plane dan data plane dari setiap switch, dan seluruh konfigurasi seperti route flow mutlak diambil alih oleh controller sehingga switch yang ada hanya berfungsi sebagai data plane yang bertugas meneruskan aliran data yang diberikan oleh controller. Selanjutnya Virtual Network akan dibuat didalam controller, sekaligus mengenali interkoneksi antar switch. Virtual Network yang telah dibuat oleh controller dapat dilihat pada menu topology yang terdapat pada sistem web GUI controller. Selanjutnya yaitu proses pengoneksian metode Hierarchical Token Bucket (HTB) dengan controller. Metode HTB menggunakan aplikasi HTB QoS untuk menentukan rate bandwidth per kelas dalam bentuk id antrian (queue id) yang di distribusikan pada semua port open vSwitch.. HTB QoS dibangun dengan menggunakan bahasa pemrograman python. Proses ini menggunakan Java rest API sebagai media interkoneksinya. Controller yang sudah berjalan dan sudah terkoneksi dengan metode HTB akan langsung melakukan proses klasifikasi lalu lintas jaringan (traffic classifying) melalui protokol pada layer-3 (IP tcp/udp) dan layer-4 (Port tcp/udp) dan menghasilkan data berupa jenis-jenis paket jaringan. Dari data tersebut kemudian dibuatlah kelas-kelas (classes) sesuai jenis paket dan diberikan policing dengan parameter min_rate, max_rate, dan priority untuk menentukan besaran bandwidth minimum, bandwidth maximum dan prioritas yang telah ditentukan sebelumnya pada HTB QoS dan sistem akan membangun skema antrian (queue tree) dari tiap kelas. Kemudian kelas-kelas tersebut dimasukan

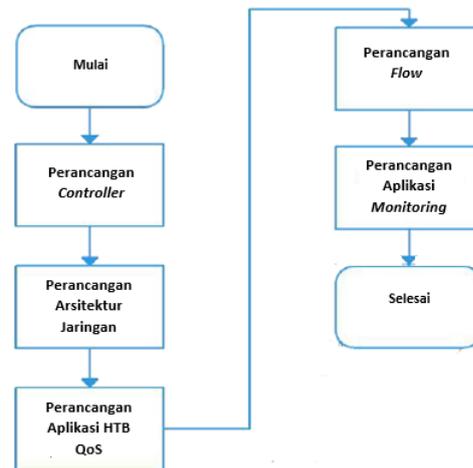
kedalam disiplin antrian queuing disciplines (qdisc) scheduling untuk dilakukan penjadwalan. Selanjutnya hasil akan dikirimkan dalam bentuk flow/policy menuju tabel flow yang ada pada controller. Tahap terakhir, controller akan melakukan push flow/policy configuration yang ada pada tabel flow menuju setiap switch yang ada pada jaringan. Selanjutnya switch akan menjalankan flow dan menerapkan bandwidth limitation dan Packet queues pada setiap host. Dengan demikian ketika host mengakses internet dengan port service tujuan seperti yang terdaftar dalam flow, maka bandwidth akan terlimit sesuai ukuran yang telah ditentukan. Setelah sistem berjalan dengan baik maka perlu dilakukan monitoring untuk melihat statistik dari traffic bandwidth yang ada. Untuk itu penulis menambahkan sistem monitoring menggunakan framework sFlow-RT yang dibuat dan dikoneksikan menggunakan Java API pada open vSwitch untuk memantau traffic bandwidth secara realtime.

Metode Hierarchical Token Bucket (HTB)

Pada penelitian ini Hierarchical Token Bucket (HTB) dipilih sebagai metode yang digunakan penulis dalam penerapan sistem Quality of Service (QoS) pada jaringan di SMKN 3 Bangkalan. Proses metode tersebut memiliki beberapa tahapan yaitu Traffic Classifier, Classes and Policing, Queuing disciplines (qdisc)(Nurfiana & Ramanda, 2019).

Perancangan Sistem

Perancangan sistem pada penelitian ini memiliki beberapa tahapan. Perancangan meliputi tahap instalasi dan konfigurasi pada setiap komponen sistem. Adapun tahapan dalam perancangan sistem ditunjukkan pada Gambar 3. berikut ini.



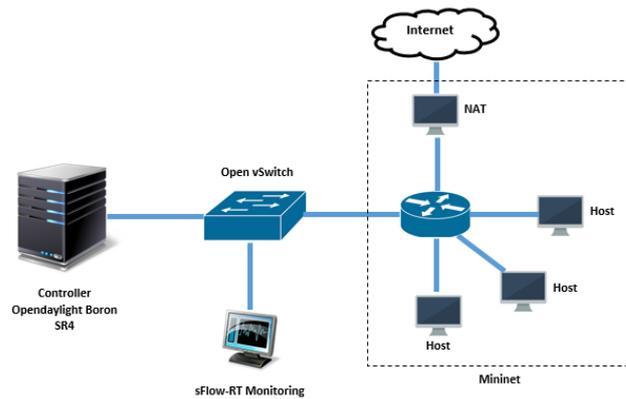
Gambar 3. Diagram Alur Perancangan Sistem

Perancangan Controller

Controller yang digunakan pada penelitian ini adalah OpenDaylight Boron SR4. Controller berfungsi sebagai server yang mengatur lalu lintas jaringan dari mulai mengatur aliran paket (packet flow), policy, dan firewall. OpenDaylight Boron SR4 merupakan framework berbasis Java dengan JDK/JRE versi 8 keatas. Sebelum bisa berjalan controller, perlu dilakukan beberapa tahapan diantaranya adalah tahapan instalasi dan build modul controller. Controller di instal pada sistem operasi Ubuntu desktop 14.04 32-Bit dan proses instalasi dilakukan secara online(Parasian Silitonga, 2014).

Perancangan Arsitektur Jaringan

Perancangan arsitektur jaringan pada penelitian ini menggunakan jaringan virtual yang dikoneksikan ke jaringan real sekolah. Sistem ini membutuhkan minimal satu buah komputer yang sudah di instal aplikasi virtualisasi yaitu Virtualbox. Komponen yang digunakan dalam arsitektur jaringan ini yaitu Opendaylight Boron SR4 sebagai controller, open vSwitch 2.0.2 sebagai virtual switch OpenFlow, dan Mininet emulator sebagai aplikasi infrastruktur jaringan virtual yang terhubung langsung ke jaringan real sekolah(Siddik, 2019). Pada penelitian ini penulis menggunakan tiga buah host sebagai media uji coba sistem. Sistem akan dipantau menggunakan sFlow-RT monitoring. Secara umum arsitektur jaringan yang dibuat pada penelitian ini dapat dilihat pada Gambar 4. berikut ini.



Gambar 4. Perancangan Arsitektur Jaringan

Pada arsitektur jaringan, penulis melakukan perancangan ip address pada setiap komponen sistem. Kelas ip yang digunakan yaitu kelas A pada host Mininet dan kelas C pada controller dan sFlow-RT. Perancangan ip address dapat dilihat pada Tabel 1. berikut ini.

Tabel 1. Perancangan ip address

No	Komponen	Ip Address	Netmask
1	Controller	192.168.100.2	255.255.255.0
2	sFlow-RT	192.168.137.2	255.255.255.0
3	Gateway	192.168.137.1	255.255.255.0
4	Host 1	10.0.0.1	255.0.0.0
5	Host 2	10.0.0.2	255.0.0.0
6	Host 3	10.0.0.3	255.0.0.0
7	NAT	10.0.0.4	255.0.0.0

Perancangan Aplikasi HTB QoS

OpenFlow memberikan dukungan terbatas untuk Quality of Service (QoS) menggunakan fungsi HTB QoS dan meter. Penelitian ini hanya memanfaatkan fungsi HTB QoS saja. HTB QoS dibuat dan dikonfigurasi di luar protokol OpenFlow. Untuk menerapkan HTB QoS perlu di buat kebijakan (policy) dengan menentukan rate bandwidth yang berbeda untuk setiap kelas dalam bentuk id antrian (queue id). Selanjutnya antrian ini ditambahkan dalam QoS policy dan didistribusikan pada seluruh port open vSwitch. Setelah selesai barulah dibuat flow pada controller dengan memanggil queue id menggunakan Java rest API. Pendistribusian HTB QoS harus dilakukan satu persatu pada port open vSwitch. Hal tersebut tentu akan sangat merepotkan bagi penulis mengingat jaringan yang akan di implementasikan terbilang cukup besar. Untuk itu penulis membuat aplikasi untuk mendistribusikan HTB QoS secara otomatis menggunakan bahasa pemrograman python. Aplikasi yang dibuat diberi nama push_htb_qos.py. Ketika aplikasi dijalankan sistem akan melakukan discovery switch yang ada di jaringan. Selanjutnya sistem akan mengecek apakah ada switch pada jaringan. jika tidak ada, maka aplikasi akan memberikan notifikasi bahwa switch tidak ditemukan dan jika switch ada, maka sistem akan menampilkan list dari setiap switch(Sudarsono et al., 2016). Selanjutnya sistem akan memilih switch dan melakukan build queue pada setiap switch tersebut. Nilai QoS Policy berupa min_rate, max_rate, priority dan queue id sudah dirancang dan ditentukan sebelumnya pada aplikasi HTB QoS. Pada protokol OpenFlow, pengekusian priority dimulai dari angka terbesar (priority 1000, lalu 100, lalu 10, dan seterusnya). Nilai total dari min_rate juga tidak boleh diatas nilai rate parent. Nilai QoS Policy yang dirancang dapat dilihat pada Tabel 2 berikut ini.

Tabel 2. Perancangan QoS Policy

No	Queue Id	Priority	Min_Rate	Max_Rate	Class
1	Newqos	1000	20 Mbps	20 Mbps	Parent
2	11	1000	10 Mbps	20 Mbps	Leaf
3	12	100	7 Mbps	20 Mbps	Leaf
4	13	10	3 Mbps	20 Mbps	Leaf

Perancangan Flow

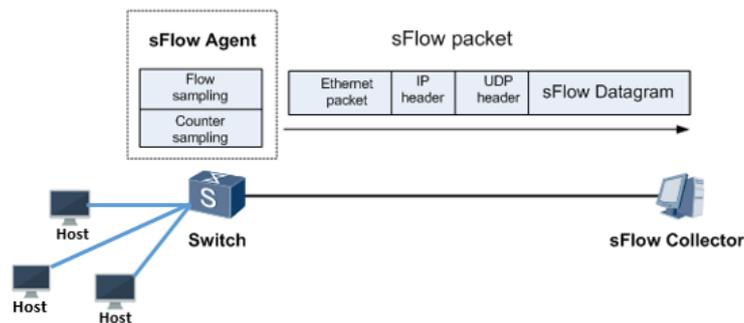
Perancangan flow merupakan tahapan pengoneksian antara hasil QoS Policy yang dibuat menggunakan aplikasi HTB QoS sebelumnya dengan controller. QoS Policy berupa id antrian (queue id) yang akan dipanggil pada controller dalam bentuk flow(Wulandari, 2016). Interkoneksi ini menggunakan Java rest API dengan memanfaatkan aplikasi Postman 6.0.10 sebagai media dalam pembuatan flow. Mehod yang yaitu PUT untuk membuat flow dan GET untuk melihat flow yang telah dibuat. Pada Tabel 3. dijelaskan perancangan flow pada penelitian ini.

Tabel 3. Perancangan Flow

No	Nodes	Tabel Id	Flow Id	Queue Id	Port	Protocol	Service
1	Openflow:1	0	Flow 1	11	80	6 (TCP)	http
2	Openflow:1	0	Flow 2	12	21	6 (TCP)	ftp
3	Openflow:1	0	Flow 3	13	25	6 (TCP)	smtp

Perancangan Sistem Monitoring

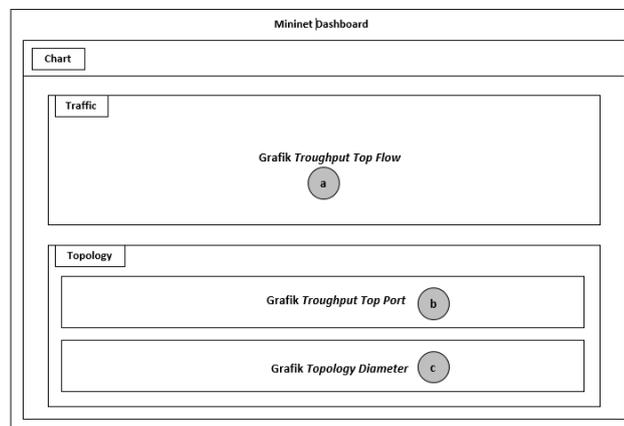
sFlow-RT merupakan framework network analytics berbasis Java API. sFlow-RT sangat fleksibel, selain bisa digunakan sebagai monitoring juga bisa sebagai firewall pada sistem yang berjalan diatas protokol OpenFlow. Pada penelitian ini sFlow-RT digunakan sebagai framework untuk membuat sistem monitoring yang digunakan untuk memantau throughput dari sistem yang dibangun. Sistem sFlow-RT melakukan dua jenis sampling yaitu random sampling dari paket atau operasi application layer dan counter sampling berbasis waktu. Sampel paket/operasi dan informasi counter disebut sebagai flow agent yang dikirim sebagai datagram sFlow ke server pusat dimana perangkat lunak yang menganalisis lalu lintas jaringan berada. Komponen ini disebut sebagai kolektor sFlow.



Gambar 5. Proses sFlow Datagram

Rancangan Antarmuka Sistem Monitoring

Rancangan antarmuka merupakan design secara tampilan sistem monitoring yang akan dibangun. Pada penelitian ini(Sudarsono et al., 2016), sistem monitoring hanya memiliki satu tampilan saja, yaitu hanya tampilan grafik statistik throughput yang ditampilkan secara real-time. Rancangan tampilan sistem monitoring ini dapat dilihat pada Gambar 6.



Gambar 6. Rancangan Antarmuka Sistem Monitoring

Keterangan Gambar 6.

- a. Menampilkan grafik throughput dari flow yang sedang aktif. Informasi yang ditampilkan berupa ip source, port source, ip destination, dan port destination.
- b. Menampilkan grafik throughput dari port yang sedang aktif. Informasi yang ditampilkan berupa port-in dan port-out.
- c. Menampilkan diameter topologi dalam bentuk grafik.

HASIL DAN PEMBAHASAN

Pengujian sistem dilakukan dengan menerapkan beberapa skenario pengujian. Tujuan dari pengujian ini adalah untuk mengetahui apakah sistem dapat berjalan dengan baik dan mampu melakukan limitasi bandwidth pada service yang sudah dirancang sebelumnya. Parameter yang digunakan pada pengujian ini berupa throughput dari bandwidth pada setiap service. Setiap tahapan pengujian akan dilakukan proses monitoring.

Pengujian Sistem

Sebelum melakukan pengujian sistem, penulis merancang beberapa skenario dengan kondisi yang berbeda-beda pada proses penggunaan bandwidth yang bertujuan untuk mendapatkan nilai throughput dari setiap skenario sebagai bahan analisis. Skenario pengujian sistem dapat dilihat pada tabel 4.

Tabel 4. Skenario Pengujian Sistem

ID	Pengujian	Tujuan
Skenario 1	Host 1 melakukan pengiriman paket pada port 80 (http), sedangkan host 2 dan host 3 melakukan pengiriman paket pada port acak. Pengiriman paket dilakukan secara bersamaan selama 100 detik menggunakan tool IPERF.	Mendapatkan nilai
Skenario 2	Host 1 melakukan pengiriman paket pada port 21 (ftp), sedangkan host 2 dan host 3 melakukan pengiriman paket pada port acak. Pengiriman paket dilakukan secara bersamaan selama 100 detik menggunakan tool IPERF.	<i>throughput</i>
Skenario 3	Host 1 melakukan pengiriman paket pada port 25 (ftp), sedangkan host 2 dan host 3 melakukan pengiriman paket pada port acak. Pengiriman paket dilakukan secara bersamaan selama 100 detik menggunakan tool IPERF.	Mendapatkan nilai
Skenario 4	Host 1, host 2, dan host 3 melakukan pengiriman paket secara bersamaan pada port 80 (http), 21 (ftp), dan 25 (smtp) selama 100 detik menggunakan tool IPERF.	<i>throughput</i>

Proses Pengujian Sistem

Pada tahap ini penulis akan melakukan pengujian dari setiap skenario yang telah dirancang pada Tabel 4 Setiap proses pengujian akan di capture dan ditampilkan dalam bentuk gambar.

Pengujian Skenario 1

Pada pengujian ini Host 1 melakukan pengiriman paket pada port 80 (http), sedangkan Host 2 dan Host 3 melakukan pengiriman paket pada port acak. Pengiriman paket dilakukan secara bersamaan selama 100 detik menggunakan tool IPERF. Hasil pengujian ini berupa nilai rata-rata throughput yang berhasil dilewati setiap host.

```

"Node: nat0" (root)
root@odlboronsr4-VirtualBox:/home/odl-boron-sr4/sflow-rt# iperf -s -f n -p 80
Server listening on TCP port 80
TCP window size: 0,08 MByte (default)
-----
[ 18] local 10.0.0.4 port 80 connected with 10.0.0.1 port 38079
[ ID] Interval      Transfer     Bandwidth
[ 18] 0,0-103,4 sec  200 MBytes  16,3 Mbits/sec
-----

"Node: nat0" (root)
root@odlboronsr4-VirtualBox:/home/odl-boron-sr4/sflow-rt# iperf -s -f n -p 100
Server listening on TCP port 100
TCP window size: 0,08 MByte (default)
-----
[ 18] local 10.0.0.4 port 100 connected with 10.0.0.2 port 44386
[ ID] Interval      Transfer     Bandwidth
[ 18] 0,0-102,0 sec  5,38 MBytes  0,44 Mbits/sec
-----

"Node: nat0" (root)
root@odlboronsr4-VirtualBox:/home/odl-boron-sr4/sflow-rt# iperf -s -f n -p 200
Server listening on TCP port 200
TCP window size: 0,08 MByte (default)
-----
[ 18] local 10.0.0.4 port 200 connected with 10.0.0.3 port 60605
[ ID] Interval      Transfer     Bandwidth
[ 18] 0,0-100,8 sec  15,1 MBytes  1,26 Mbits/sec
-----
    
```

Gambar 7. Pengujian Skenario 1



Gambar 8. Grafik Throughput Skenario 1

Pada Gambar 7. dari proses pengujian, host 1 mendapatkan nilai rata-rata throughput sebesar 16,3 Mbps, host 2 mendapatkan nilai rata-rata throughput sebesar 0,44 Mbps, dan host 3 mendapatkan nilai rata-rata throughput sebesar 1,26 Mbps. Pada Gambar 8. merupakan tampilan grafik dari throughput setiap host yang melakukan pengiriman paket. Dapat dilihat bahwa grafik throughput pada host 1 dengan port service tujuan 80 (http) berjalan stabil sesuai limitnya, yaitu berjalan pada bandwidth diantara min_rate 10 Mbps dan max_rate nya 20 Mbps. Sedangkan, pada host 2 dan host 3 mendapatkan bandwidth yang sangat kecil. Hal itu terjadi karena pada host 2 dan host 3 menjalankan service dengan port yang tidak diberikan QoS Policy, sehingga tidak ada proses limitasi bandwidth

Pengujian Skenario 2

Pada pengujian ini host 1 melakukan pengiriman paket pada port 21 (ftp), sedangkan host 2 dan host 3 melakukan pengiriman paket pada port acak. Pengiriman paket dilakukan secara bersamaan selama 100 detik menggunakan tool IPERF. Hasil pengujian ini berupa nilai rata-rata throughput yang berhasil dilewati setiap host.

```

"Node: nat0" (root)
root@odlboronsr4-VirtualBox:/home/odl-boron-sr4/sflow-rt# iperf -s -f n -p 21
Server listening on TCP port 21
TCP window size: 0,08 MByte (default)
-----
[ 18] local 10.0.0.4 port 21 connected with 10.0.0.1 port 38541
[ ID] Interval      Transfer     Bandwidth
[ 18] 0,0-106,6 sec  211 MBytes  16,6 Mbits/sec
-----

"Node: nat0" (root)
root@odlboronsr4-VirtualBox:/home/odl-boron-sr4/sflow-rt# iperf -s -f n -p 100
Server listening on TCP port 100
TCP window size: 0,08 MByte (default)
-----
[ 18] local 10.0.0.4 port 100 connected with 10.0.0.2 port 44376
[ ID] Interval      Transfer     Bandwidth
[ 18] 0,0-105,5 sec  11,9 MBytes  0,94 Mbits/sec
-----

"Node: nat0" (root)
root@odlboronsr4-VirtualBox:/home/odl-boron-sr4/sflow-rt# iperf -s -f n -p 200
Server listening on TCP port 200
TCP window size: 0,08 MByte (default)
-----
[ 18] local 10.0.0.4 port 200 connected with 10.0.0.3 port 60695
[ ID] Interval      Transfer     Bandwidth
[ 18] 0,0-121,7 sec  9,50 MBytes  0,85 Mbits/sec
-----
    
```

Gambar 9. Pengujian Skenario 2

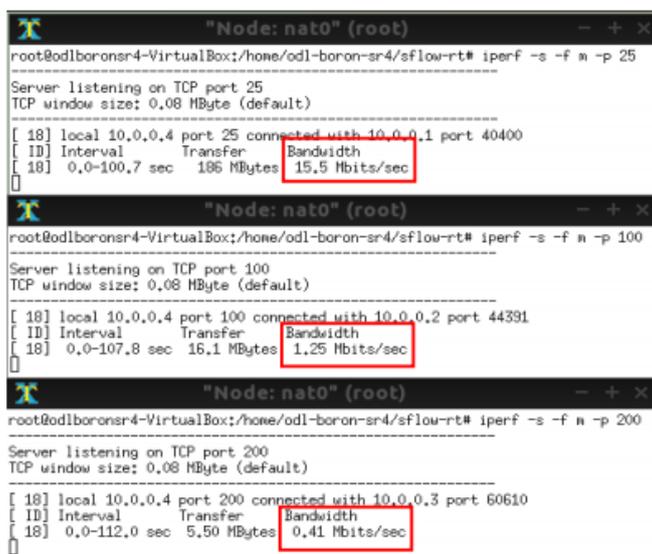


Gambar 10. Grafik Throughput Skenario 2

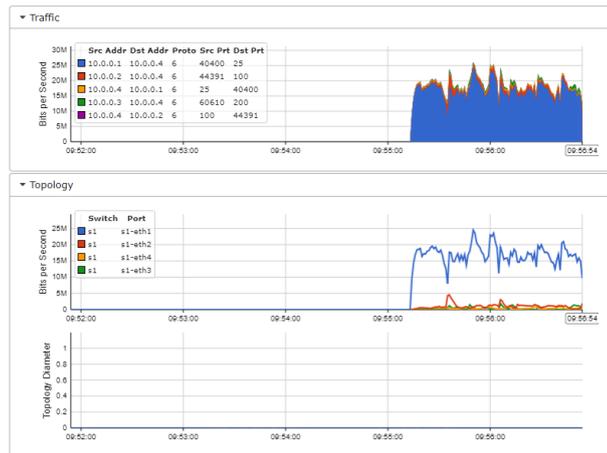
Pada Gambar 9. dari proses pengujian, host 1 mendapatkan nilai rata-rata throughput sebesar 16,6 Mbps, host 2 mendapatkan nilai rata-rata throughput sebesar 0,96 Mbps, dan host 3 mendapatkan nilai rata-rata throughput sebesar 0,65 Mbps. Pada Gambar 10. merupakan tampilan grafik dari throughput setiap host yang melakukan pengiriman paket. Dapat dilihat bahwa grafik throughput pada host 1 dengan port service tujuan 21 (ftp) berjalan stabil sesuai limitnya, yaitu berjalan pada bandwidth diantara min_rate 7 Mbps dan max_rate nya 20 Mbps. Sedangkan, pada host 2 dan host 3 mendapatkan bandwidth yang sangat kecil. Hal itu terjadi karena pada host 2 dan host 3 menjalankan service dengan port yang tidak diberikan QoS Policy, sehingga tidak ada proses limitasi bandwidth (bandwidth limitation) yang dapat menjamin ketersediaan bandwidth pada service tersebut ketika dalam kondisi jaringan sibuk.

Pengujian Skenario 3

Pada pengujian ini host 1 melakukan pengiriman paket pada port 25 (smtp), sedangkan host 2 dan host 3 melakukan pengiriman paket pada port acak. Pengiriman paket dilakukan secara bersamaan selama 100 detik menggunakan tool IPERF. Hasil pengujian ini berupa nilai rata-rata throughput yang berhasil dilewati setiap host.



Gambar 9. Pengujian Skenario 3



Gambar 11. Grafik Throughput Skenario 3

Pada Gambar 10. dari proses pengujian, host 1 mendapatkan nilai rata-rata throughput sebesar 15,5 Mbps, host 2 mendapatkan nilai rata-rata throughput sebesar 1,25 Mbps, dan host 3 mendapatkan nilai rata-rata throughput sebesar 0,41 Mbps. Pada Gambar 11. merupakan tampilan grafik dari throughput setiap host yang melakukan pengiriman paket. Dapat dilihat bahwa grafik throughput pada host 1 dengan port service tujuan 25 (smtp) berjalan stabil sesuai limitnya, yaitu berjalan pada bandwidth diantara min_rate 3 Mbps dan max_rate nya 20 Mbps. Sedangkan, pada host 2 dan host 3 mendapatkan bandwidth yang sangat kecil. Hal itu terjadi karena pada host 2 dan host 3 menjalankan service dengan port yang tidak diberikan QoS Policy, sehingga tidak ada proses limitasi bandwidth (bandwidth limitation) yang dapat menjamin ketersediaan bandwidth pada service tersebut ketika dalam kondisi jaringan sibuk.

Pengujian Skenario 4

Pada pengujian ini host 1, host 2, dan host 3 melakukan pengiriman paket secara bersamaan pada port 80 (http), 21 (ftp), dan 25 (smtp) selama 100 detik menggunakan tool IPERF. Hasil pengujian ini berupa nilai rata-rata throughput yang berhasil dilewati setiap host.

```

"Node: nat0" (root)
root@odlboron-sr4-VirtualBox:~/home/odl-boron-sr4/sflow-rt# iperf -s -p 80
-----
Server listening on TCP port 80
TCP window size: 85,3 KByte (default)
-----
[ 18] local 10.0.0.4 port 80 connected with 10.0.0.1 port 42122
[ ID] Interval      Transfer    Bandwidth
[ 18] 0.0-102.3 sec  112 MBytes  10,15 Mbits/sec
[]

"Node: nat0" (root)
root@odlboron-sr4-VirtualBox:~/home/odl-boron-sr4/sflow-rt# iperf -s -p 21
-----
Server listening on TCP port 21
TCP window size: 85,3 KByte (default)
-----
[ 18] local 10.0.0.4 port 21 connected with 10.0.0.2 port 52017
[ ID] Interval      Transfer    Bandwidth
[ 18] 0.0-106.9 sec  85.9 MBytes  7,08 Mbits/sec
[]

"Node: nat0" (root)
root@odlboron-sr4-VirtualBox:~/home/odl-boron-sr4/sflow-rt# iperf -s -p 25
-----
Server listening on TCP port 25
TCP window size: 85,3 KByte (default)
-----
[ 18] local 10.0.0.4 port 25 connected with 10.0.0.3 port 53157
[ ID] Interval      Transfer    Bandwidth
[ 18] 0.0-111.6 sec  45.9 MBytes  3,08 Mbits/sec
[]
    
```

Gambar 12. Pengujian Skenario 4



Gambar 13. Grafik Throughput Skenario 4

Pada Gambar 12. dari proses pengujian, host 1 mendapatkan nilai rata-rata throughput sebesar 10,15 Mbps, host 2 mendapatkan nilai rata-rata throughput sebesar 7,08 Mbps, dan host 3 mendapatkan nilai rata-rata throughput sebesar 3,08 Mbps. Pada Gambar 13. merupakan tampilan grafik dari throughput setiap host yang melakukan pengiriman paket. Dapat dilihat bahwa grafik throughput pada ketiga host berjalan stabil sesuai limit dan prioritas yang telah dirancang pada QoS Policy. Pada QoS Policy, port service 80 (ftp) telah diatur min_rate sebesar 10 Mbps dan max_rate sebesar 20 Mbps dengan prioritas tertinggi yaitu 1000, port service 21 (ftp) telah diatur min_rate sebesar 7 Mbps dan max_rate sebesar 20 Mbps dengan prioritas 100, dan port service 25 (smtp) telah diatur min_rate sebesar 3 Mbps dan max_rate sebesar 20 Mbps dengan prioritas 10. Besar limit max_rate pada ketiga service disamakan dengan max_rate pada parent yaitu 20 Mbps, bertujuan agar setiap host dapat menggunakan bandwidth secara penuh ketika host lain tidak sedang menggunakan bandwidth. Sedangkan untuk total jumlah nilai min_rate dari setiap service diatur sebesar max_rate pada parent. Hal tersebut bertujuan agar tidak ada bandwidth yang terbuang ketika setiap service diakses secara bersamaan. Hasil pengujian skenario 4 menunjukkan bahwa host 1, host 2, dan host 3 mendapatkan bandwidth mendekati nilai min_rate nya. Hal itu terjadi dikarenakan ketiga host melakukan pengiriman paket pada port service secara bersamaan sehingga traffic menjadi sibuk. Pada kondisi ini, sistem akan memastikan setiap host mendapatkan bandwidth sesuai limit pada masing-masing service yang diakses. Service dengan prioritas tertinggi akan lebih dahulu mencapai nilai min_rate nya, kemudian dilanjutkan pada prioritas dibawahnya.

KESIMPULAN

Proses pengujian skenario 1, skenario 2, skenario 3, dan skenario 4 yang dilakukan sebelumnya bertujuan untuk mendapatkan hasil nilai throughput pada masing-masing service yang berjalan. Dari hasil nilai throughput yang didapat, penulis dapat mengetahui apakah sistem mampu memberikan jaminan bandwidth dengan melakukan limitasi bandwidth untuk setiap service yang telah diberi QoS Policy. Sebuah service dapat dikatakan mendapatkan jaminan bandwidth apabila nilai rata - rata throughput berada antara min_rate dan max_rate pada setiap kondisi jaringan. Hasil pengujian sistem ditunjukkan pada Tabel 4.

Tabel 5. Hasil Pengujian Sistem

No	ID	Host	Port	Service	QoS Policy		Throug hput (Mbps)	Mendapat Jaminan Bandwidth? (Y/N)
					Min Rate (Mbps)	Max Rate (Mbps)		
1	Skenario 1	1	80	http	10	20	16,3	Y
		2	acak	-	-	-	0,44	N
		3	acak	-	-	-	1,26	N
2	Skenario 2	1	21	ftp	7	20	16,6	Y
		2	acak	-	-	-	0,94	N
		3	acak	-	-	-	0,65	N
3	Skenario 3	1	25	smtp	3	20	15,5	Y
		2	acak	-	-	-	1,25	N
		3	acak	-	-	-	0,41	N
4	Skenario 4	1	80	http	10	20	10,15	Y
		2	21	ftp	7	20	7,08	Y
		3	25	smtp	3	20	3,08	Y

Berdasarkan data hasil pengujian pada Tabel 5, pengujian service dengan port 80 (http), 21 (ftp), dan 25 (smtp) berhasil mendapatkan jaminan bandwidth pada setiap skenario. Hal itu dibuktikan dengan nilai throughput yang didapat pada port 80 (http), port 21 (ftp), dan port 25 (smtp) berada antara min_rate dan max_rate sesuai dengan QoS Policy yang sudah dirancang. Pada skenario 1, service dengan port 80 (http) berhasil mendapatkan throughput sebesar 16,3 Mbps. pada skenario 2, service dengan port 21 (ftp) berhasil mendapatkan throughput sebesar 16,6 Mbps. pada skenario 3, service dengan port 25 (smtp) berhasil mendapatkan throughput sebesar 15,5 Mbps. Pada skenario 4, ketika semua service yang diberi QoS Policy dijalankan secara bersamaan, sistem juga mampu memberikan jaminan bandwidth untuk setiap service, yaitu port 80 (http) berhasil mendapatkan throughput sebesar 10,15 Mbps, port 21 (ftp) berhasil mendapatkan throughput sebesar 7,08 Mbps, dan port 25 (smtp) berhasil mendapatkan throughput sebesar 3,08 Mbps. Dari hasil pengujian, maka dapat disimpulkan bahwa sistem yang dibangun dapat melakukan limitasi bandwidth dengan baik dan dapat memberikan jaminan bandwidth pada setiap service walaupun dalam kondisi jaringan yang sibuk. Nilai throughput yang diperoleh selama proses pengujian sistem menunjukkan bahwa limitasi bandwidth dilakukan berdasarkan urutan prioritas. Service dengan prioritas tertinggi akan mendapatkan limitnya terlebih dulu, kemudian sisa bandwidth yang ada akan didistribusikan pada service yang berada dibawah prioritasnya. Sistem monitoring yang dibangun menggunakan framework sFlow-RT dapat berjalan dengan baik dan dapat menampilkan informasi grafik troughput dari setiap service yang berjalan.

UCAPAN TERIMA KASIH

Terima Kasih Kepada Civitas Akademik SMKN 3 Bangkalan Khusus Penanggung Jawab Laboratorium Ibu Afi. Terima Kasih Juga Kepada Ardian Fadli Yang Banyak Memberikan Referensi Terhadap Penelitian Ini.

DAFTAR PUSTAKA

- Ahdan, S., Firmanto, O., & Ramadona, S. (2018). Rancang Bangun dan Analisis QoS (Quality of Service) Menggunakan Metode HTB (Hierarchical Token Bucket) pada RT/RW Net Perumahan Prasanti 2. *Jurnal Teknoinfo*, 12(2), 49. <https://doi.org/10.33365/jti.v12i2.89>
- Antodi, C. P., Prasetijo, A. B., & Widiyanto, E. D. (2017). Penerapan Quality of Service Pada Jaringan Internet Menggunakan Metode Hierarchical Token Bucket. *Jurnal Teknologi Dan Sistem Komputer*, 5(1), 23. <https://doi.org/10.14710/jtsiskom.5.1.2017.23-28>
- Aprianto Budiman, M. Ficky Duskarnaen, & Hamidillah Ajie. (2020). Analisis Quality of Service (Qos)

-
- Pada Jaringan Internet Smk Negeri 7 Jakarta. *PINTER : Jurnal Pendidikan Teknik Informatika Dan Komputer*, 4(2), 32–36. <https://doi.org/10.21009/pinter.4.2.6>
- Fahlepi Roma Doni. (2019). IMPLEMENTASI MANAJEMEN BANDWIDTH PADA JARINGAN KOMPUTER DENGAN ROUTER MIKROTIK. *EVOLUSI : Jurnal Sains Dan Manajemen*, 7(2), 52–57.
- Hanifia, R. (2019). Penerapan Quality of Service (QoS) Differentiated Service Pada Jaringan Multi-Protocol Label Switching (MPLS). *Jurnal Manajemen Informatika*, 9(2), 1–7.
- IGO., G. tuberculosis report 2019. G. W. H. O. 2019. L. C. B.-N.-S. 3. . (2012). No Titleתמונת: ענף הקיבוי: תמונת: עלון הנושע, 66, 39–37.
- Melyana, I., & Indriyani, T. (2016). Analisa Quality Of Service Dan Implementasi Voice Over Internet Protocol Dengan Menggunakan IPSEC VPN. *Integer Journal*, 1. No. 2, 53–66.
- Nurfiana, N., & Ramanda, D. (2019). Implementasi Metode Pcq-Queue Tree Pada Router Mikrotik Dan Monitoring Cacti Untuk Peningkatan Quality of Service. *Jurnal Ilmiah Teknologi Informasi Dan Robotika*, 1(1), 1–7. <https://doi.org/10.33005/jifti.v1i1.4>
- Parasian Silitonga, I. S. M. (2014). Analisis QoS (Quality of Service) Jaringan Kampus dengan Menggunakan Microtic Routerboard. *TIMES*, 152(March), 73–75.
- Siddik, M. (2019). ANALISIS QUALITY OF SERVICE JARINGAN LOCAL AREA NETWORK MENGGUNAKAN MIKROTIK ROUTERBOARD750 (Studi Kasus: STMIK Royal Kisaran). *JURTEKSI (Jurnal Teknologi Dan Sistem Informasi)*, 5(2), 113–118. <https://doi.org/10.33330/jurteksi.v5i2.354>
- Sudarsono, A., Siswanto, A., Iswanto, H., & Setiawan, Q. (2016). Traffic Analysis of Quality of Service (QoS) for Video Conferencing between Main Campus and Sub Campus in Laboratory Scale. *EMITTER International Journal of Engineering Technology*, 3(2). <https://doi.org/10.24003/emitter.v3i2.41>
- Wulandari, R. (2016). ANALISIS QoS (QUALITY OF SERVICE) PADA JARINGAN INTERNET (STUDI KASUS : UPT LOKA UJI TEKNIK PENAMBANGAN JAMPANG KULON – LIPI). *Jurnal Teknik Informatika Dan Sistem Informasi*, 2(2), 162–172. <https://doi.org/10.28932/jutisi.v2i2.454>