PERANCANGAN DAN ANALISIS SISTEM PADA KONTROLER POX, RYU, DAN OPENDAYLIGHT PADA SOFTWARE DEFINED NETWORK
DESIGN AND ANALYSIS SYSTEM ON CONTROLLER POX, RYU, AND OPENDAYLIGHT ON SOFTWARE DEFINED NETWORK

Romasenna Edgar[1], Ahmad Tri Hanuranto[2], Osphanie Mentari[3]

[1,2,3]Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

[1]edgar.napitupulu@gmail.com [2]at.hanuranto@gmail.com [3]osphanie.mentari@gmail.com

Abstrak

SDN (*Software Defined Network*) adalah suatu arsitektur yang dapat berkembang, dengan perkembangan jaringan yang cukup signifikan maka SDN akan membuat jauh lebih mudah arsitektur jaringan. Pada SDN kendali jaringan dipusatkan pada kontroller yang diprogram mampu untuk mengatur proses jalannya data ke pusat dan mengirim ke *client.* Pada penelitian tugas akhir ini, analisa yang akan dilakukan adalah membandingkan QoS (*Quality of Service*) jaringan yang akan dibangun menggunakan kontroler POX, kontroler Ryu, dan kontroler Opendaylight. Dari kedua kontroler ini akan diuji pada topologi tree dengan jumlah switch 7, 9, dan 11. Dan 6 dari 7 switch memiliki 2 host, 8 dari 9 switch memiliki 2 host, dan 10 dari 11 switch memiliki 2 host.

Kata Kunci : Software Defined Network, POX, Ryu, Opendaylight, Mininet, OpenFlow

Abstract

SDN (Software Defined Network) is an architecture that can develop, with a significant network development, SDN will make network architecture much easier. At SDN network control is centered on the controller programmed to be able to manage the process of running the data to the center and sending it to the client. In this final project research, the analysis that will be carried out is comparing the QoS (Quality of Service) network that will be built using POX controllers, Ryu controllers, and Opendaylight controllers. Of the two controllers will be tested on the tree topology with the number of switches 7, 9, and 11. And 6 of the 7 switches have 2 hosts, 8 of the 9 switches have 2 hosts, and 10 of the 11 switches have 2 hosts.

***Keywords :* Software Defined Network, POX, Ryu, Opendaylight, Mininet, OpenFlow**.

1. Pendahuluan

Software Defined Network adalah arsitektur jaringan baru. Salah satu komponennya adalah controller, yang merupakan bagian cerdas dari SDN. Banyak pengendali seperti Opendaylight, Open Daylight, Maestro, NOX, POX, Ryu, dan masih banyak lagi. Untuk mengatasi masalah jaringan yang mendesak seperti virtualisasi jaringan dan kompleksitas pusat data, prinsip yang sangat umum untuk diadopsi dalam merancang platform baru seperti membuat, mengelola, dan meningkatkan layanan jaringan sesuai permintaan, adalah dengan cara yang lebih cepat dan lebih mengoptimalkan sumber daya. Sehingga memunculkan pertanyaan kontroler yang memiliki kinerja yang lebih baik serta melihat kondisi akhir jaringan ketika beban di berikan kedalam jaringan tersebut. dalam Tugas Akhir ini kontroler POX, Ryu, serta Opendaylight akan dilakukan serangkaian uji coba dengan menggunakan virtualisasi pada Mininet. POX dan Ryu merupakan kontroller berbasis *python* dan Opendaylight merupakan kontroler berbasis *java*. Analisis ini berdasarkan kemampuan sebuah kontroler yang mampu lebih baik dan dibandingkan ulang ke kontroler lainnya.2.1 Software Difiend Network.

2. Dasar Teori

2.1 Software Defined Network

Software-defined networking (SDN) adalah sebuah konsep pendekatan jaringan komputer dimana sistem pengkontrol dari arus data dipisahkan dari perankat kerasnya. Umumnya, sistem pembuat keputusan kemana arus data dikirimkan dibuat menyatu dengan perangkat kerasnya. Sebuah konfigurasi SDN dapat menciptakan jaringan dimana perankat keras pengontrol lalu lintas data secara fisik dipisahkan dari perangkat keras data forwarding plane Konsep ini dikembangkan di UC Berkeley and Stanford University sekitar tahun 2008.[1]

2.2 Kontroller OpenFlow

Openflow adalah salah satu jenis APIs (*Application Protocol Interfaces*) dalam jaringan SDN yang digunakan untuk mengontrol/mengatur traffic flows pada switch dalam sebuah jaringan, jadi singkatnya control plane berkomunikasi dengan data plane melalui OpenFlow. OpenFlow memungkinkan mengakses langsung dan manipulasi perangkat forwarding plane seperti switch dan router, baik fisik dan virtual (hypervisor-based)[2].

Kontroler openflow adalah perangkat lunak yang memiliki fungsi sebagai pengatur dan konfigurasi kontrol plane. Kontroler openflow bekerja dengan mengkonfigurasi switch pada open flow dengan memperbarui *table flow*. Kontroler juga dapat dikembangkan dibeberapa jaringan yang berbeda, sehingga dapat menyesuaikan dengan kondisi jaringan yang ada.

2.3 POX

POX adalah platform pengembangan sumber terbuka untuk aplikasi SDN dengan bahasa pemograman berbasis *Python*, seperti kontroler OpenFlow SDN. POX, memungkinkan pengembang dan pembuatan prototipe yang cepat, menjadi lebih umum digunakan daripada NOX, proyek sejenis. POX dapat menjalankan aplikasi yang berbeda seperti hub, switch, load balancer, dan firewall. Alat capture paket Tcpdump bisa digunakan untuk menangkap dan melihat paket yang mengalir di antaranya POX controller dan perangkat OpenFlow. POX mempunyai komponen forwarding.l2_learning yang membuat switch OpenFlow bertindak sebagai jenis learning switch L2. Yang satu ini beroperasi seperti contoh "pyswitch" NOX, meskipun implementasinya sangat berbeda[3].

2.4 RYU

Ryu adalah kerangka kerja SDN berbasis komponen. Ryu menyediakan komponen perangkat lunak dengan API terdefinisi dengan baik yang memudahkan pengembang untuk membuat manajemen jaringan baru dan aplikasi kontrol. Ryu mendukung berbagai protokol untuk mengelola perangkat jaringan, seperti OpenFlow, Netconf, OF-config, dll. Tentang OpenFlow, Ryu mendukung sepenuhnya 1.0, 1.2, 1.3, 1.4, 1.5, dan Nicira Extensions. Semua kode tersedia secara bebas di bawah lisensi Apache 2.0. Ryu berbasis bahasa python dan bersifat Open source[4].

2.5 OpenDayLight

Opendaylight adalah Pengontrol SDN yang dikembangkan oleh komunitas pengembang terbuka LINUX, dan merupakan jawaban atas kebutuhan industri pada jaringan yang memiliki kemampuan pemograman yang mandiri. banyak di antaranya dari Big Switch Networks, yang menggunakan protokol OpenFlow untuk mengatur arus lalu lintas dalam lingkungan jaringan yang ditentukan perangkat lunak (SDN).

2.6 Mininet

Mininet adalah emulator jaringan yang menciptakan jaringan host virtual, sakelar, pengontrol, dan tautan. Host Mininet menjalankan perangkat lunak jaringan Linux standar, dan sakelar-sakunya mendukung OpenFlow untuk perutean kustom yang sangat fleksibel dan Jaringan yang Ditentukan-Perangkat Lunak. Mininet memiliki kekurangan yaitu mininet tidak bisa dijalankan pada OS selain Linux atau yang tidak mendukung switch OpenFlow[5]. Paket diproses oleh apa yang tampak seperti switch Ethernet nyata, router, atau middlebox, dengan jumlah antrian yang diberikan. Ketika dua program, seperti klien dan server iperf, berkomunikasi melalui Mininet, kinerja yang diukur harus sesuai dengan dua mesin asli (lebih lambat).

2.7 Spanning Tree Protocol

Spanning Tree Protocol (STP) adalah protokol manajemen tautan yang menyediakan redundansi jalur sembari mencegah loop yang tidak diinginkan dalam jaringan. Ketika datang ke jaringan ethernet, agar mereka berfungsi dengan baik hanya satu jalur aktif yang bisa ada di antara dua stasiun. Loop terjadi di jaringan karena

berbagai alasan[6]. Alasan paling umum Anda menemukan loop dalam jaringan adalah hasil dari mencoba menyediakan hanya beberapa tingkat redundansi.

2.8 Topologi Tree

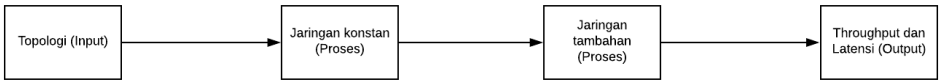
Topologi pohon adalah jenis struktur khusus di mana banyak elemen yang terhubung diatur seperti cabang-cabang pohon. Misalnya, topologi pohon sering digunakan untuk mengatur komputer dalam jaringan perusahaan, atau informasi dalam database. Dalam topologi pohon, hanya ada satu koneksi antara dua node yang terhubung. Karena dua node hanya dapat memiliki satu koneksi timbal balik, topologi pohon membentuk hirarki induk dan anak[7].

2.9 Parameter Penguji

Dalam menganalisa tugas akhir ini dibutuhkan parameter sebagai pengukur data yang akan diperbandingkan, yaitu *Quality of Service* dan *Resource Utilization.*

3. Model Sistem dan Perancangan Sistem

3.1 Gambaran Umum Sistem



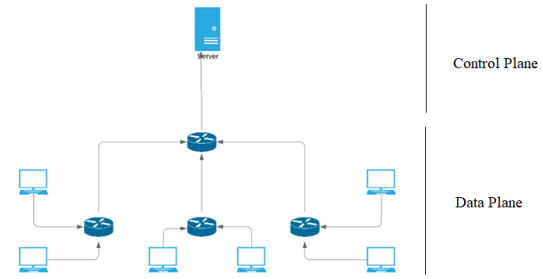
Gambar 3.1 Gambaran umum sistem.

Sistem yang dirancang pada tugas akhir ini adalah mensimulasikan suatu jaringan berbasiskan Software Defined Network yang terdiri dari topologi sebagai input, jaringan konstan dan jaringan tambahan sebagai proses, dan ditahap terakhir ada throughput dan latensi sebagai output. Simulasi ini akan dijalankan dengan mininet sebagai emulator dan mengunakan dua kontroler yaitu, POX, Ryu, dan Opendaylight. Kontroler ini akan digunakan sebagai skenario jaringan menggunakan topologi tree. Dalam simulasi ini akan melakukan pengukuran dari hasil QoS dari kedua kontroler.

Pada perancangan topologi ini terbagi menjadi 3 rancangan topologi tree dengan jumlah switch 7, 9, dan 11. Setiap switch akan terhubung dengan 2 host, dan sisakan 1 switch sebagai switch utama penyambung ke kontroler jadi ada 6, 8, dan 10 switch yang memiliki host. Pada setiap topologi akan diambil data-data untuk pengukuran QoS, data yang diambil pada 5 jenis keadaan jaringan yakni saat background traffic 0 Mbps, 50Mbps, 100 Mbps, 150Mbps dan 200 Mbps.

Pengukuran QoS yang dilakukan untuk pengukuran nilai throughput, jitter, delay, dan packet loss. Setiap pengujian akan diambil nilai rata-rata dan dibandingkan antara dua kontroler yaitu POX, Ryu, dan Opendaylight. Hasil pengujian akan diperbandingkan untuk mendapatkan suatu kesimpulan dengan pengaruh dua kontroler terhadap performansi sebuah jaringan SDN.

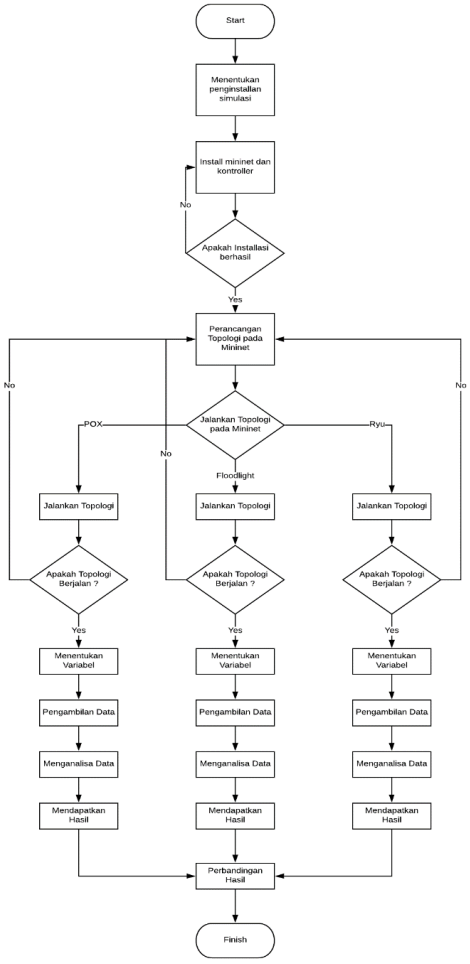
3.2 Design Model Sistem



Gambar 3.2 Blok Diagram Perancangan

Gambar 3.2 diatas menjabarkan sistem yang akan dibuat. Sistem simulasi ini mensimulasikan suatu jaringan yang terdiri dari *control plane* dan *data plane*. Untuk menjalankan simulasi ini controller yang digunakan ada 3 macam yaitu controller POX, controller Ryu, dan controller Opendaylight. Controller ini berkerja di bagian control plane yang bertugas mengontrol aliran data, melakukan routing dan melakukan semua peraturan SDN.

3.3 Diagram Alur Sistem



Gambar 3.3 Diagram alur sistem.

Gambar 3.3 diatas menjelaskan gambaran umum penggunaan sistem. Yang menjelaskan secara umum dan alur pengerjaan tugas akhir ini mulai dari pengumpulan data, perancangan sistem, melakukan simulasi sehingga mendapatkan data, analisa data, melakukan perbandingan, dan mengambil kesimpulan.

3.4 Perangkat yang digunakan

Perangkat yang digunakan untuk merancang tugas akhir ini terbagi menjadi 2 bagian yaitu perangkat keras (*Hardware)* dan perangkat lunak (*Software*).

3.4.1 Hardware

Pada perancangan sistem untuk tugas akhir ini, dibutuhkan perangkat keras untuk dapat merealisasikan sistem dengan baik. Berikut daftar perangkat keras yang digunakan.

Tabel 3.1 Spesifikasi hardware.

Nama Alat	Versi	Spesifikasi singkat
Laptop	Asus ROG	-Ram 16GB -HDD 1 Tera -Prosessor intel i7

3.4.2 Software

Selain perangkat keras, dibutuhkan juga perangkat lunak yang ditanamkan yang pada perangkat keras. Berikut ini daftar perangkat lunak yang digunakan.

Tabel 3.2 Spesifikasi software.

Speksifikasi	PC
Operating Sistem	Oracle VM Virtual Box Ubuntu 19.04 64bit dengan RAM 8 GB dan Ubuntu 19.04 64bit dengan RAM 4 GB
Installasi di Ubuntu	Control plane : POX, Ryu, dan Opendaylight controller Data plane : Mininet dan Openflow

3.5 Skenario Pengujian

Untuk mengetauhi kinerja kontroller dari sistem SDN yang telah dirancang maka akan dilakukan pengujian dengan parameter-parameter yaitu QoS dan resource utilization.

Untuk pengujian oleh QoS menggunakan topologi yang sudah disediakan. Hal ini dilakukan untuk mengetahui kinerja controller, pengujian QoS ini juga melibatkan parameter delay, jitter, throughput, and packet loss. Percobaan akan dilakukan dengan trafik UDP yang kita ambil datanya. Data yang diambil berupa data dari D-ITG.

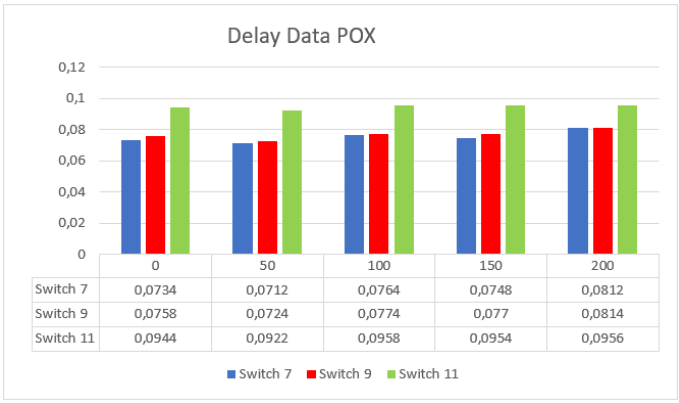
Untuk pengujian resource utilization mengetahui komsumsi memori controller selama controller dijalankan. Untuk mengetahui komsumsi memori dari controller dapat diambil dari sistem ubuntu itu sendiri.

Padapengujiankaliiniakanbanyakvariasipengujiandenganbackgroundtrafficdengan bantuan ipref. Terdapat jenis 5 background traffic yang digunakan yaitu 0 Mbps, 50Mbps, 100 Mbps, 150Mbps dan 200 Mbps. Nilai dari parameter QoS yang didapat akan dibandingkan dengan nilai yang telah di tetapkan oleh badan standarisasi ITU-T. sehingga dapat diketahui nilai dari parameter QoS tersebut sudah memenuhi standarisasi atau belum.

4. Hasil Pengujian dan Analisis

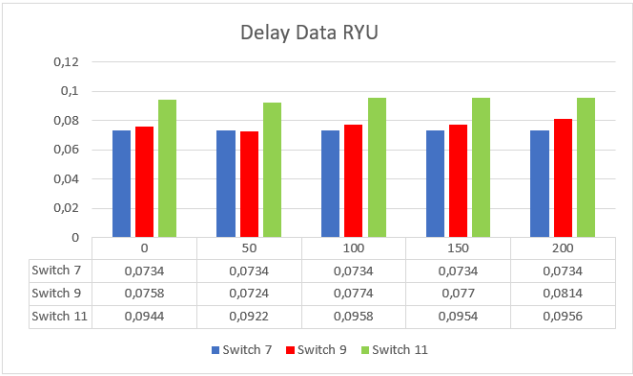
4.1 Delay

Delay adalah waktu tunda yang disebabkan oleh proses transmisi dari satu titik ke titik lain yang menjadi tujuannya. Delay yang diukur pada pengujian ini adalah one-way-delay.



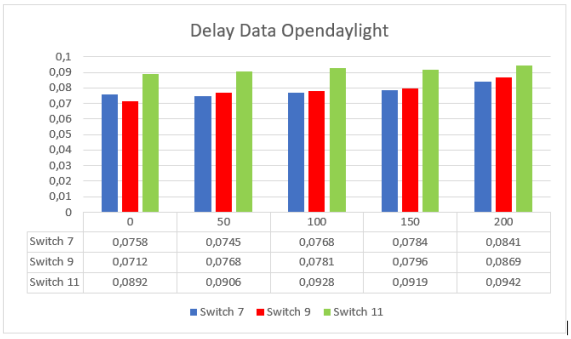
	0	50	100	150	200
Switch 7	0,0734	0,0712	0,0764	0,0748	0,0812
Switch 9	0,0758	0,0724	0,0774	0,077	0,0814
Switch 11	0,0944	0,0922	0,0958	0,0954	0,0956

Gambar 4.1 Hasil pengujian delay POX dengan layanan data.



	0	50	100	150	200
Switch 7	0,0734	0,0734	0,0734	0,0734	0,0734
Switch 9	0,0758	0,0724	0,0774	0,077	0,0814
Switch 11	0,0944	0,0922	0,0958	0,0954	0,0956

Gambar 4.2 Hasil pengujian delay RYU dengan layanan data.

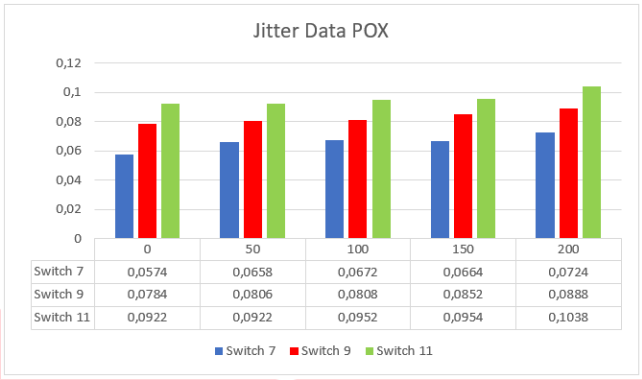


	0	50	100	150	200
Switch 7	0,0758	0,0745	0,0768	0,0784	0,0841
Switch 9	0,0712	0,0768	0,0781	0,0796	0,0869
Switch 11	0,0892	0,0906	0,0928	0,0919	0,0942

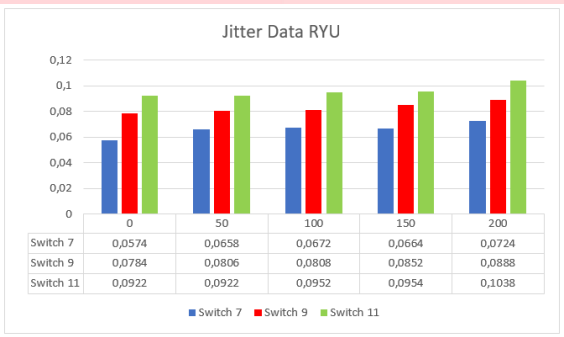
Gambar 4.3 Hasil pengujian delay OpenDayLight dengan layanan data.

4.2 Jitter

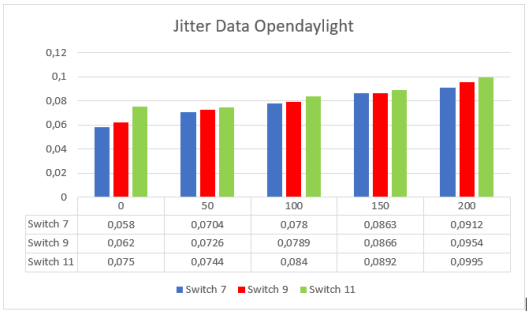
Jitter merupakan variasi delay antar paket yang terjadi padajaringan. Besarnya nilai jitter akan sangat dipengaruhi oleh variasi bebeban trafik dan banyaknya antrian yang terjadi. Pada grafik pengujian jitter menujukan hasil dari jitter dengan layanan data dan VoIP tidak jauh berbeda, namun pada layanan video nilai jitter meningkat. Nilai jitter yang dihasilkan pada layanan video berbeda dengan layanan data dan VoIP,hal ini dikarenakan ukuran paket video yang lebih besar dari layanan data dan VoIP.



Gambar 4.4 Hasil pengujian jitter POX dengan layanan data.



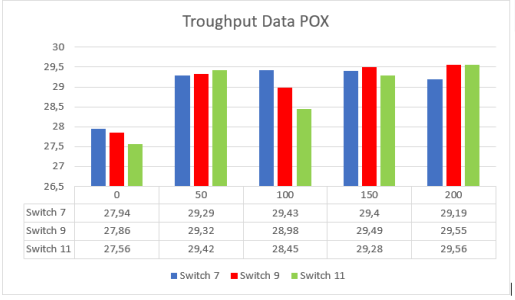
Gambar 4.5 Hasil pengujian jitter RYU dengan layanan data.



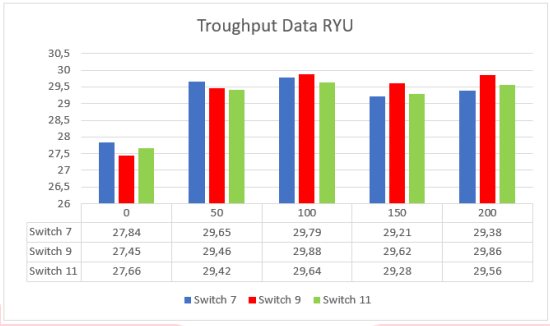
Gambar 4.6 Hasil pengujian jitter OpenDayLight dengan layanan data.

4.3 Throughput

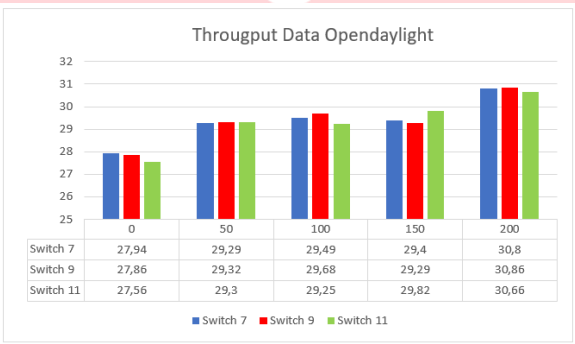
Throughput adalah perbandingan jumlah bit paket yang sukses diterima dengan total waktupengiriman. Dari hasil yang didapat menunjukan bahwa nilai throughput pada setiap layanan data, video, dan VoIP terlihat konstan walaupun terjadi penambahan jumlah switch dan peningkatan nilai background traffic. Hal ini dikarenakan besar background traffic masih dibawah besar maksimal yang disediakan link.



Gambar 4.7 Hasil pengujian throughput POX dengan layanan data.



Gambar 4.8 Hasil pengujian throughput RYU dengan layanan data.



Gambar 4.9 Hasil pengujian throughput RYU dengan layanan data.

4.4 Packet Loss

Packet loss ratio adalah persentase perbandingan antara jumlah paket yang gagal diterima oleh user dengan jumlah paket yang dikirimkan. Pada pengujian packet loss ditambahkan satu besaran *background traffic* selain 0Mbps, 50Mbps,100Mbps, dan 150Mbps yaitu 200Mbps.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Perbandingan hasil performa QoS pada uji coba jaringan pada controller POX, Ryu dan Opendaylight memiliki hasil seperti berikut:

1. Hasil *resource utilizatiom* dapat dilihat bahwa setiap kontroler POX, Ryu dan Opendaylight memiliki hasil yang tidak jauh berbeda sehingga ketiga kontroler tersebut memiliki performa jaringan yang baik.
2. Melakukan analisa pada kontroler selain POX, Ryu, dan OpenDayLight seperti Floodlight, Beacon, dll.
3. . Pada Opendaylight membutuhkan dua Ubuntu yaitu satu untuk mininet dan satu untuk Opendaylight.

5.2 Saran

1. Untuk percobaan selanjutnya di VoIP atau video streaming dengan minimal percobaan sepuluh kali tes ulang karena mencari data yang random butuh waktu untuk mengstabilkannya.
2. Penggunaan Opendaylight bisa menjadi satu Unbuntu dengan syarat kapasitas memori harus lebih dari 100 *Giga Byte*. karena untuk kapasitas memori Opendaylight sangatlah besar.
3. Melakukan analisa pada kontroler selain POX, Ryu, dan OpenDayLight seperti Floodlight, Beacon, dll.
4. Melakukan analisa ini dengan jaringan SDN secara *real* atau tidak lagi menggunakan emulator

Daftar Pustaka:

[1] Tryfon Theodorou and Lefteris Mamatas. " CORAL-SDN: A software-defined networking solution for the Internet of Things " Greece. University of Macedonia. 2017.

[2] Gopakumar, R., Unni, A. M., & Dhipin, V. P. (2015). "*An adaptive algorithm for searching in flow tables of openflow switches". 2015 39th National Systems Conference (NSC).*

[3] Fancy, C., & Pushpalatha, M. (2017). "*Performance evaluation of SDN controllers POX and Opendaylight in mininet emulation environment*". 2017 International Conference on Intelligent Sustainable Systems (ICISS).

[4] S. Ellinidou, G. Sharma, T. Rigas, T. Vanspouwen, O. Markowitch, and J.-M. Dricot, "Sspsoc: A secure sdn-based protocol over mpsoc," Security and Communication Networks, vol. 2019, 2019.

[5] R. Barrett, A. Facey, W. Nxumalo, J. Rogers, P. Vatcher, and M. St-Hilaire, "Dynamic traffic diversion in sdn: Testbed vs mininet," in 2017 International Conference on Computing, Networking and Communications (ICNC). IEEE, 2017, pp. 167–171

[6] Y. N. Krishnan, C. N. Bhagwat, and A. P. Utpat, "Optimizing spanning tree protocol using port channel,"in 2014 International Conferenceon Electronics and Communication Systems (ICECS). IEEE, 2014, pp. 1–5.

[7] S.Rowshanrad,V.Abdi,and M.Keshtgari,"Performance evaluation of sdncontrollers: Floodlight and opendaylight," IIUM Engineering Journal, vol. 17, no. 2, pp. 47–57, 2016.