Optimasi *Cost* pada *Open Shortest Path First* di Jaringan *Software Defined-Network*

Cost Optimization on Open Shortest Path First in Software Defined-Network

Ronald Adrian
Departemen Teknik Elektro dan Informatika Sekolah Vokasi
Universitas Gadjah Mada
e-mail : ronald.adr@ugm.ac.id

Abstrak

Software Defined-Network merupakan teknologi baru di dunia jaringan. Teknologi SDN ini memungkinkan sebuah server yang disebut sebagai kontroler dapat mengendalikan semua perangkat jaringan yang terhubung dengannya. Semua konfigurasi dan sumberdaya dari perangkat jaringan menjadi terpusat pada kontroler. Salah satunya adalah routing yang dilakukan secara terpusat. Hal ini memudahkan administrator jaringan dalam mengkonfigurasi routing pada jaringan yang kompleks. Penelitian ini memiliki fokus pada implementasi OSPF dan analisis kinerja QoS pada jaringan SDN. Implementasi OSPF yang dilakukan menggunakan pengaturan cost dan tanpa pengaturan cost. Pengaturan cost ini dilakukan seperti konfigurasi pengaturan cost pada jaringan konvensional. Hal tersebut yang berpengaruh terhadap pemilihan jalur data utama pada routing OSPF. Sedangkan implementasi pada penelitian ini dilakukan menggunakan perangkat mikrotik. Pengambilan data melibatkan waktu konvergensi dan beberapa parameter QoS seperti Throughput, PLR, Jitter dan Delay. Pengujiannya menggunakan pembangkitan trafik data menggunakan iperf dan D-ITG dengan variasi jenis data yang ada. Penelitian ini memiliki tujuan mencari konfigurasi routing terbaik pada jaringan SDN khususnya yang menggunakan OSPF.

Kata Kunci--SDN, QoS, Delay, Jitter, Cost.

Abstract

Software Defined-Network is a new technology in the network engineering. This technology allows a server which is called by a controller and it control all connected devices. All configurations and resources of network devices become centralized to the controller. One of them is routing configuration. This makes it easier for network administrators to configure routing on complex networks. This study focused on OSPF implementation and QoS performance analysis on SDN networks. OSPF implementation can be configured using cost and no cost. This configuration can be done as a configuration of cost settings on conventional networks. It affects the selection of main data paths on OSPF routing. This implementation in this research used the mikrotik devices. Data retrieval involves convergence time and some QoS parameters such as Throughput, PLR, Jitter and Delay. In testing phase used traffic data which generated by iperf and D-ITG with variations of existing data types. The goal of this research is to finding the best routing configuration on SDN networks especially in OSPF.

Keywords--*SDN, QoS, Delay, Jitter, Cost.*

1. PENDAHULUAN

Teknologi jaringan internet khususnya bidang infrastruktur jaringan mengalami peningkatan pesat pada beberapa tahun terakhir ini. Menurut data yang dirilis APJII, bahwa jumlah pengguna internet di Indonesia mencapai 88 juta orang pada tahun 2014 [1]. Seiring bertumbuhnya penggunaan jaringan maka tingkat kompleksitas meningkat, seperti

meningkatnya kinerja *routing*. Konfigurasi *routing* pada jaringan konvensional masih dilakukan secara individual, hal tersebut menyebabkan tidak fleksibel terhadap perubahan. [2]

Beberapa tahun terakhir teknologi SDN menjadi salah satu topik menarik bagi peneliti. Teknologi SDN merupakan teknologi jaringan dimana bagian infrastruktur perangkat, yakni *control plane* dan *data plane* dilakukan pemisahan, sehingga kebijakan *routing* dapat dilakukan terpusat melalui *controller*. Pengkontrolan jaringan terpusat SDN membuat pengaturan jaringan lebih mudah dan fleksibel. Pengembangan SDN yang dilakukan saat ini telah meliputi bagian hal, seperti *load balancing*, VLAN, dan salah satunya mengenai protokol *routing* melalui *RouteFlow*.

Perbedaan arsitektur pada SDN dan konvensional memungkinkan akan menghasilkan perbedaan yang mendasar dari sektor kinerja. Berdasarkan hal tersebut penelitian ini dilakukan dengan melakukan analisis performa dari protokol *routing* OSPF yang ada pada jaringan SDN dan konvensional berdasarkan penggunaan *cost* dan tanpa penggunaan *cost* dengan memanfaatkan Routeflow. Pemilihan *routing protocol* OSPF karena OSPF menjadi salah satu *routing protocol* yang banyak digunakan, sementara penggunaan variabel *cost* sebagai instrumen pembandingan dikarenakan OSPF merupakan *routing protocol* LSA, dimana penentuan rute akan ditentukan berdasarkan nilai dari *cost* tersebut. Penelitian sejenis juga telah dilakukan dengan menggunakan jenis protokol berbeda, yakni RIP dan EBGP ataupun menggunakan layanan lain seperti SNHX-IP untuk RYU *controller*. Terdapat perbedaan mendasar karena penggunaan perangkat jaringan riil sebagai bagian yang berperan menjadi *forwarding plane* akan didapatkan hasil yang lebih realistis dengan kondisi pada lapangan. Melalui topik ini diharapkan nantinya akan dihasilkan keluaran dengan parameter setara yang dapat menentukan pilihan terbaik dalam menentukan *routing protocol* yang digunakan dalam jaringannya.

SDN merupakan paradigma baru dalam pengembangan jaringan. Penelitian pada bidang *routing* dilakukan untuk menganalisa bagaimana kinerja protokol *routing* yang sebelumnya diimplementasikan dalam jaringan konvensional, dan kemudian diimplementasikan pada teknologi paradigma baru. Penelitian SDN pada umumnya masih dilakukan pada *mininet*. Mininet *host* menjalankan standar perangkat lunak jaringan linux, dan *switch* yang mendukung *OpenFlow* untuk *custom routing* yang sangat fleksibel dan SDN [3].

Beberapa penelitian yang telah dilakukan pada jaringan SDN diantaranya pengujian performa kontroler SDN: POX dan Floodlight [4]. Floodlight lebih menjamin pengelolaan data dalam jumlah besar dan membutuhkan pengaturan aliran data yang sangat tinggi. Sementara POX memberikan jaminan penanganan yang lebih konstan untuk berapapun jumlah host yang dikelolanya.

Penelitian di atas menjadi salah satu contoh penelitian SDN tanpa dilakukan fungsi *routing*. Pada sebuah jaringan yang besar, kebutuhan *routing* tidak dapat dihindari baik pada jaringan konvensional maupun SDN. Kebutuhan *routing* pada jaringan SDN untuk saat ini dapat terpenuhi oleh beberapa kontroler, seperti penggunaan RouteFlow dan SNHX-IP yang berbasis pada kontroler RYU.

Penggunaan kontroler RouteFlow dalam penelitian telah berhasil diterapkan pada beberapa *routing protocol*, seperti OSPF, eBGP, dan RIP. Penelitian yang dilakukan melalui simulasi *routing* menggunakan protokol eBGP dan dilakukan pada jaringan SDN dengan mengambil data QoS (*Quality of Service)* menyatakan memenuhi standar ITU-T jika dialiri *background traffic* sampai 75 Mbps. Dari hasil uji coba juga didapatkan beberapa hal, dimana hasil waktu konvergensi, dan *Routing overhead* dipengaruhi oleh jumlah *switch* dan fitur [5].

Selain eBGP, pada *routing protocol* RIP melalui proses analisis yang dilakukan berdasarkan parameter yang sama, hasil dari simulasi diperoleh nilai parameter *convergence time* yang mendekati *timeout timer* dan terus meningkat seiring dengan pertambahan *switch*.

Nilai parameter *routing overhead* dan *memory utilization* setiap penambahan *switch* tidak terlalu besar sesuai karakteristik RIPv2. Selanjutnya, dihasilkan nilai parameter QoS juga memenuhi kriteria dan standar ITU-T [6].

Sedikit berbeda dengan apa yang dilakukan penelitian sebelumnya, uji coba yang dilakukan selanjutnya masih menggunakan RouteFlow namun menggunakan *routing protocol* yang berbeda, yakni OSPF. Ujicoba OSPF ini dilakukan dengan menggunakan satu jenis topologi yakni topologi Abilene dengan 7, 9 dan 11 *host*. [7].

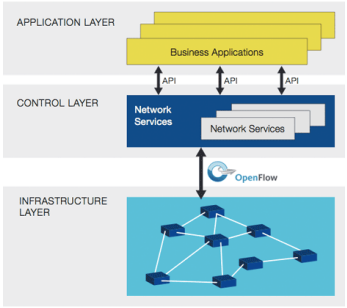
Routing OSPF juga dapat diterapkan dengan menggunakan SNHX-IP jika menggunakan kontroler RYU. Penggunaan SNHX-IP menunjukkan bahwa nilai dari keempat parameter QoS (*jitter, PLR, throughput,* dan *latency)* masih berada pada nilai yang menjadi standar ITU-T G.1010 serta memiliki nilai *delay* dan *jitter* yang lebih baik dibandingkan penerapan OSPF berbasis jaringan Konvensional. [8]. Pada penelitian tersebut juga dilakukan modifikasi *cost* pada tiap jaringan yang dilaluinya berdasarkan skenario yang telah dibuat.

Terdapat penelitian lain yakni penggunaan Mikrotik RB750 untuk melakukan uji coba atas protokol *OpenFlow*, dan hasil pengujian menunjukan MikroTik dapat menjalankan protokol *OpenFlow* serta penggunaan arsitektur SDN dengan protokol *OpenFlow* tidak akan menghambat performa perangkat. [9].

Berdasar uraian hasil penelitian sebelumnya dapat diketahui bahwa pada umumnya jaringan SDN dilakukan uji coba pada jaringan simulasi. Terdapat sejumlah peneliti yang menggunakan *mininet* sebagai representasi jaringan SDN terhadap jumlah *node* dalam jaringan, membandingkan kinerja antar kontroler, ataupun menjalankan *routing* dari berbagai *routing protocol*. Penelitian ini akan mengimplementasikan OSPF *routing protocol* pada SDN yang didalamnya akan dilakukan pengaturan *cost*. Berdasarkan hasil yang didapatkan, kemudian akan dilakukan perbandingan dengan kinerja yang dihasilkan OSPF pada arsitektur Konvensional.

A. *Software Defined Network (SDN)*

Software Defined Network merupakan paradigma baru dari perkembangan teknologi jaringan saat ini. SDN merupakan teknologi yang memisahkan bagian dari perangkat jaringan yakni, *Control Plane* dan *Data Plane*. Pemisahan tersebut akan menjadikan posisi kontroler memiliki peran sebagai pembuatan kebijakan yang akan dijalankan pada suatu jaringan dan akan pelaksana kebijakan tersebut adalah perangkat jaringan yang pada SDN perannya hanya sebagai *forwarding* atau *data plane*. Hal tersebut dapat dilihat melalui Gambar 1 di bawah ini.



Gambar 1. Arsitektur Software Defined Network

Pada Arsitektur *Software Defined Networking (SDN)* setiap layer dapat bekerja secara independen dan berkomunikasi melalui antarmuka jaringan untuk memberikan fungsi berlapis dari perangkat fisik yang berbeda. Aspek arsitektur ini memungkinkan administrator jaringan untuk mengatasi beberapa tantangan dalam dunia jaringan komputer.

Gambaran logis arsitektur *Software Defined Networking* pada gambar 1 menunjukkan terdapat 3 layer pada arsitektur *Software Defined Networking* (SDN) [2], yaitu :

1. Layer Infrastruktur, terdiri dari elemen-elemen jaringan dan perangkat keras yang menjalankan fungsi paket *switching* dan *forwarding*.

2. Layer Kontrol, menyediakan fungsionalitas kontrol secara padu yang mengawasi perilaku jaringan *forwarding* melalui *open interface*.

3. Layer Aplikasi, berfungsi untuk menyediakan interface dalam pembuatan program aplikasi yang kemudian akan mengatur dan mengoptimalkan jaringan secara baik dan fleksibel.

B. OpenFlow

OpenFlow merupakan protokol utama pada sebuah jaringan SDN yang berada di antara *controller* dan *forwarding layer*. OpenFlow memungkinkan akses langsung ke dan manipulasi *forwarding* bidang perangkat jaringan seperti switch dan router, baik fisik maupun virtual (*hypervisor-based*). Teknologi OpenFlow pada SDN memungkinkan untuk mengatasi *bandwidth* yang tinggi, untuk aplikasi bersifat dinamis, jaringan yang bersifat adaptif dan selalu berubah tergantung kebutuhan model bisnis, dan secara signifikan mengurangi operasi pada kompleksitas manajemen. *OpenFlow* dimungkinkan pengaturan *routing* serta pengiriman paket melalui sebuah switch. Komponen utama pada arsitektur jaringan SDN berbasis pada *OpenFlow* adalah *OpenFlow switch* dan Kontroler.

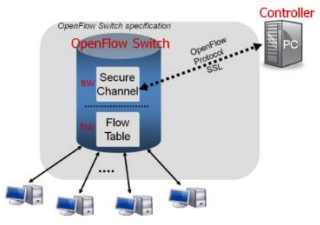
C. OpenFlow Switch

OpenFlow *switch* dapat dibagi menjadi tiga bagian yaitu:

1. Sebuah *flow table* yang mengindikasikan bahwa switch harus memroses *flow* yang ada di dalamnya. Daftar *flow* ini dibuat berdasarkan *actions* yang mana bersinggungan langsung dengan setiap *flow.*

2. Sebuah saluran yang aman dibutuhkan untuk menghubungkan *switch* dengan *controller*. Melalui saluran ini, OpenFlow menyediakan jalur komunikasi antara switch dan *controller* melalui protokol yang disebut protokol OpenFlow.

3. Komponen yang terakhir adalah protokol OpenFlow. Protokol ini menyediakan sebuah standard dan komunikasi terbuka antara kontroler dan switch. OpenFlow protocol menentukan ke *interface* manakah *flow* akan diterapkan dari *flow table*. Gambaran mengenai konsep dari OpenFlow switch dapat dilihat pada Gambar 2.

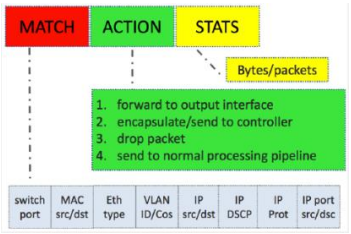


Gambar 2. Mekanisme Switch OpenFlow

D. OpenFlow Table

Pada switch OpenFlow terdapat tabel yang berisi dari tiga bagian yaitu : *rule, action* dan *statistic* seperti yang terdapat pada Gambar 3. *Rule* merupakan sekumpulan kondisi yang akan di bandingkan dengan paket yang akan masuk ke switch, yang di baca adalah *header-header* dari setiap lapisan seperti *mac address, ip address, port number, protocol* dan lain sebagainya. *Action* merupakan tindakan yang akan dilakukan jika terdapat paket yang masuk ke switch dan sesuai dengan rule, dapat berupa perintah untuk meneruskan paket keluar ke port sekian atau

men-drop paket dan lain sebagainya. Pada *flow table* juga terdapat statistik dari masing-masing *flow* berupa jumlah paket dan jumlah bytes.



Gambar 3. OpenFlow Table

E. RouteFlow

RouteFlow adalah sebuah proyek *open source* untuk menyediakan layanan *virtual IP routing* pada perangkat yang mengaktifkan OpenFlow. *Routing engine* pada RouteFlow dijalankan pada oleh Quagga. RouteFlow disusun oleh tiga aplikasi dasar: RFClient (sebagai daemon di *Virtual Machine*, RFServer (mengelola VM yang menjalankan daemon), dan RFProxy (interaksi dengan OF melalui protokol OF).

F. Quagga

Quagga adalah suatu *routing engine* yang dapat menjalankan protokol routing konvensional seperti RIP, OSPF, BGP, dan lain-lain.Quagga memiliki tujuan umum sebagai implementasi *routing* yang bersifat *open source*.

Arsitektur Quagga terdiri dari core daemon, zebra, yang bertindak sebagai layer abstraksi untuk Unix kernel yang mendasari dan menyajikan Zserv API diatas unix atau TCP stream untuk Quagga *clients.*

G. Protokol Ruting Open Shortest Path First.

Protokol routing adalah serangkaian proses, algoritma, dan *message* yang digunakan untuk pertukaran informasi *routing* dan menduduki tabel *routing* dengan pemilihan jalur terbaik berdasarkan *routing protocol* yang digunakan. Tujuan dari protokol diantaranya: deteksi jaringan *remote*, menjaga *up-to-date* informasi ruting, memilih jalur terbaik ke jaringan tujuan, dan kemampuan untuk menemukan jalur terbaik. OSPF merupakan protokol *link-state*. Alih-alih saling bertukar jarak (*distance*) ke tujuan, setiap *router* dalam jaringan memiliki peta jaringan yang dapat diperbarui dengan cepat setelah setiap perubahan.

H. Cost OSPF

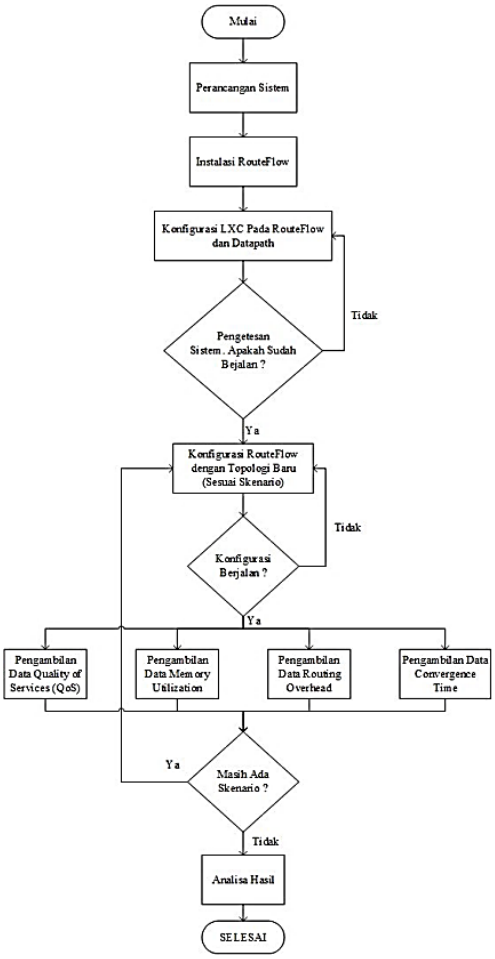
Cost dari sebuah antarmuka di OSPF merupakan indikasi dari *overhead* yang diperlukan untuk mengirim paket pada/di antarmuka tertentu. *Cost* antarmuka berbanding terbalik dengan bandwidth antarmuka. Perhitungan cost secara umum dapat dilihat pada Tabel 1.

Adapun perhitungan *cost* dilakukan melalui formula:

Cost = Reference bandwidth / Interface bandwidth in bps.

2. METODE PENELITIAN

Penelitian ini dilakukan dengan beberapa tahap yaitu: analisis kebutuhan, perancangan topologi, perancangan sistem, pengujian dan pengambilan data serta perhitungan dan analisa. Adapun alur penelitian digambarkan pada Gambar 4.



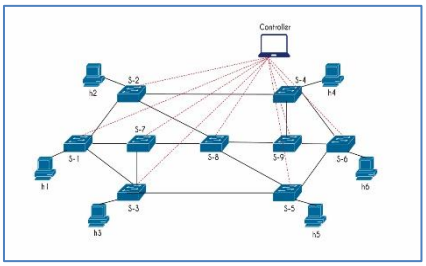
Gambar 4. Metode Penelitian

A. Analisis Kebutuhan

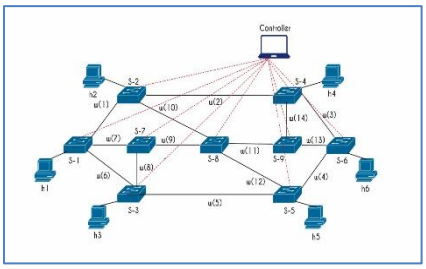
Kebutuhan yang digunakan pada penelitian ini cukup beragam. Kebutuhan dibedakan terhadap dua jenis, yakni perangkat keras dan perangkat lunak. Perangkat keras yang digunakan diantaranya: 7 buah laptop/pc yang digunakan 6 sebagai host dan 1 sebagai *controller* dalam SDN, 9 router MikroTik RB951 – MIPSBE (support openflow 1.0), 1 switch non OF, dan kabel LAN. Sedangkan perangkat lunak yang digunakan untuk menjalankan dan mengambil data saat pengujian antara lain: Ubuntu Server 12.04, RouteFlow Controller, Quagga, Wireshark, Iperf, dan D-ITG.

B. Perancangan Topologi

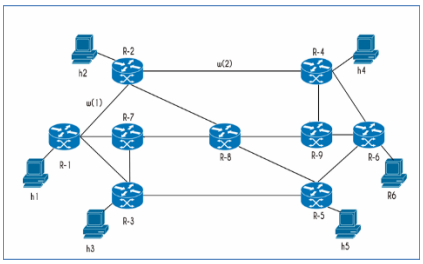
Pengujian dalam penelitian ini dilakukan dalam 2 skenario, keduanya menggunakan 9 router MikrotTik yang difungsikan sebagai switch OF. Perbedaan keduanya terletak pada topologi SDN dan konvensional yang menjalankan *routing protocol* OSPF menggunakan dan tanpa menggunakan *cost* seperti pada Gambar 5-8.



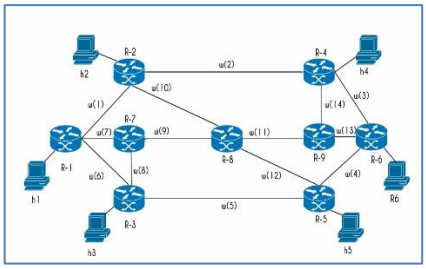
Gambar 5. Topologi SDN – Non Cost



Gambar 6. Topologi SDN – Cost



Gambar 7. Topologi Konvensional – Non Cost



Gambar 8. Topologi Konvensional – Cost

C. Skenario Pengujian

Pada penelitian ini, pengujian dilakukan terhadap *Quality of Service (delay, jitter, throughput, PLR).* Nilai yang dihasilkan berdasarkan parameter tersebut nantinya digunakan dalam analisis. Topologi yang digunakan sesuai dengan skenario pada Gambar 5-8. Fungsi *routing* pada topologi dijalankan melalui arsitektur RouteFlow. Sehingga setiap switch MikroTik yang berperan sebagai data plane terhubung dengan kontroler RouteFlow.

Pengukuran QoS dilakukan untuk mengukur tingkat performa jaringan apabila jaringan yang telah dibangun dan dibentuk dengan skenario tertentu dialiri dengan trafik dari beragam

jenis, seperti trafik data, VoIP, dan video [2]. Pengujian dilakukan dengan mengambil beberapa data pada setiap pemberian background traffic (1-100Mbps).

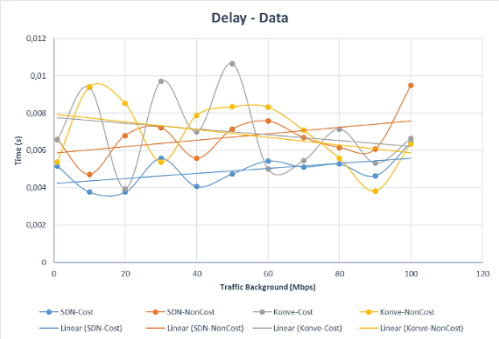
3. HASIL DAN PEMBAHASAN

Berdasarkan pengambilan data yang telah dilakukan dengan menggunakan *tools* berupa iperf, dan D-ITG sebagai pembangkit *traffic background* dan pengalir aliran trafik UDP dari beragam jenis tipe data [2], didapatkan gambaran hasil sebagai berikut.

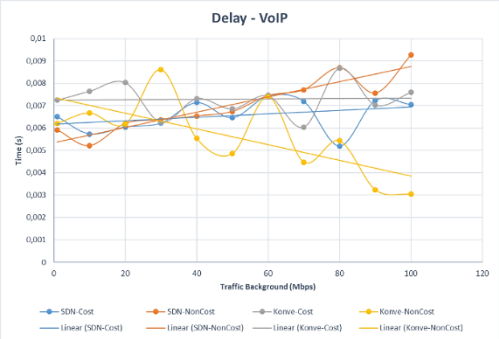
A. Quality of Services (QoS)

1. Delay

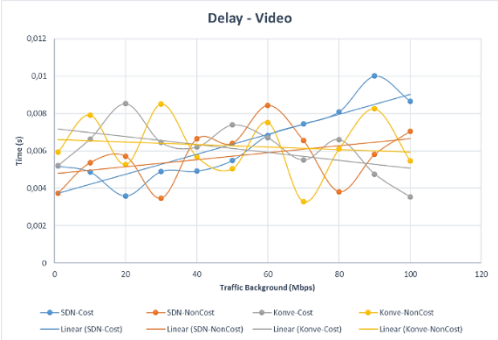
Pada gambar 7-9 di bawah ini menampilkan besaran nilai *delay* untuk setiap jenis trafik dan sampel yang berhasil dikirimkan kepada tujuan. Besaran delay yang berdasar ketiga trafik (data, VoIP, dan video) memiliki nilai yang bervariasi dan cenderung memiliki beberapa karakteristik kesamaan.



Gambar 9. Grafik Delay Trafik Data



Gambar 10. Grafik Delay Trafik VoIP



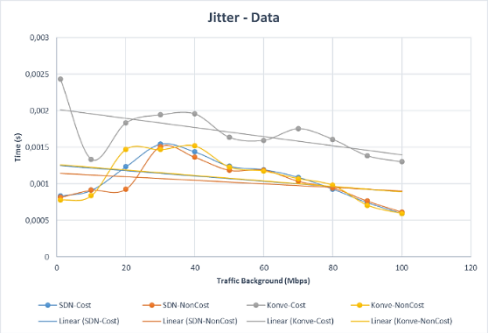
Gambar 11. Grafik Delay Trafik Video

Pada Gambar 7 diperlihatkan bahwa trafik dari keempat jenis skenario topologi memiliki respon yang berbeda terhadap kenaikan *background traffic*. Hal tersebut terlihat dimana topologi konvensional memiliki tren yang cenderung menurun baik yang menggunakan konfigurasi *cost* ataupun tidak, sedangkan topologi SDN mengalami tren kenaikan yang konstan naik terhadap aliran *background traffic* yang diberikan. Meskipun kedua topologi konvensional mengalami tren penurunan namun secara nilai keseluruhan menunjukkan bahwa topologi SDN yang menggunakan *cost* untuk jenis trafik data memiliki nilai delay yang lebih rendah dibanding ketiga topologi lainnya seperti yang terlihat pada Gambar 9. Sedangkan pada Gambar 10 dan 11 yang menunjukkan jenis trafik VoIP dan Video, memiliki karakteristik yang hampir sama dengan trafik data, dimana nilai *delay* dari topologi konvensional seakan tidak berpengaruh berbanding lurus terhadap aliran besaran background traffic meskipun telah dialiri background traffic sampai dengan 100 Mbps.

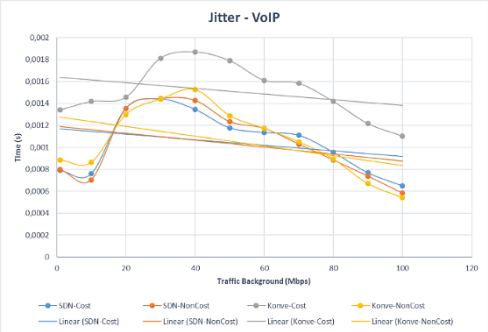
Peningkatan tren delay trafik data yang berbanding lurus dengan besar *background traffic* pada keempat skenario terjadi ketika *background traffic* dialiri 50Mbps dan 100 Mbps. Hal ini disebabkan oleh tingginya utilitas jaringan yang mengakibatkan tiap node mengalami tren penambahan, sehingga waktu kedatangan paket yang dikirimkan akan menempuh waktu yang relatif lebih lama. Namun trafik VoIP dan Video mengalami peningkatan delay yang signifikan untuk keempat topologi ketika jaringan dialiri background traffic 60 Mbps. Selain itu, Peningkatan delay kembali terjadi terhadap 3 topologi lainnya kecuali konvensional *non-cost* pada titik 90 Mbps (VoIP) dan 100 Mbps (Video) kecuali pada topologi konvensional menggunakan *cost*. Hal tersebut sebagai tanda bahwa kondisi jaringan telah mencapai batas maksimal atau *bottleneck*. Kapasitas link yang terbatas (100 Mbps), harus melewatkan jenis trafik beserta background traffic sebesar 1-100 Mbps.

Ketiga jenis trafik memiliki besaran nilai delay dalam rentang yang sama. Dalam hal ini topologi SDN menggunakan *cost* memiliki nilai *delay* yang lebih stabil jika mengambil berdasarkan besaran delay ketiga trafik tersebut. Pemberian *background traffic* pada pengujian *delay* juga membuktikan bahwa tidak semua jenis trafik data dan topologi terpengaruh oleh variabel tersebut.

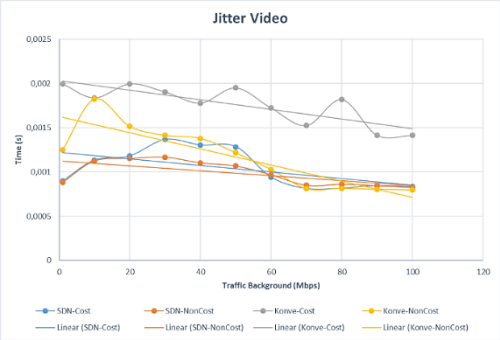
2. Jitter



Gambar 12. Grafik Jitter Trafik Data



Gambar 13. Grafik Jitter Trafik VoIP

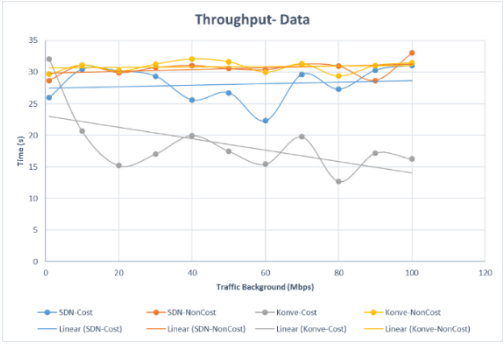


Gambar 14. Grafik Jitter Trafik Video

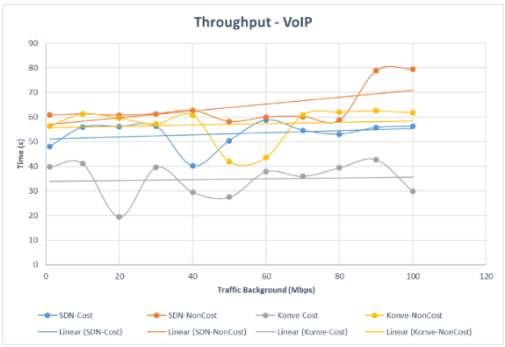
Pada Gambar 12-14, Ketiga grafik, yakni data, voip dan video memiliki karakteristik yang sama yakni mengalami tren penurunan terhadap peningkatan *background traffic* secara keseluruhan, besarnya ukuran paket layanan,dan kapasitas *link* yang dilalui setiap paket layanan data, voip, dan video sebesar 100 Mbps sehingga nilai jitter yang ketika diberi *background traffic* tiap kelipatan juga mengalami perbedaan. Peningkatan tertinggi dari ketiga trafik layanan di atas terjadi pada saat pemberian *background traffic* sebesar 30-40 Mbps berbanding terbalik dengan nilai jitter yang ketika diberi *background traffic* 50-100 Mbps yang cenderung mengalami penurunan nilai jitter. Hal tersebut berkaitan dengan paket dari ketiga jenis layanan di atas yang mengalami penumpukkan pada jaringan dan router ataupun switch sehingga mengakibatkan terjadinya congestion serta antrian antar paket layanan. Grafik di atas juga menunjukkan perbedaan yang cukup signifikan nilai jitter yang didapat pada topologi jaringan konvensional, namun memiliki pola yang sama saat peningkatan dan penurunan trafik. Semakin

berbeda dan semakin besar selisih waktu kedatangan ditujuan. Berdasarkan hasil pengukuran rata–rata jitter untuk layanan data, voip, dan video dengan trafik background 0 – 90 Mbps yang didapat tersebut memenuhi kriteria dan standar ITU-T G.1010 [10].

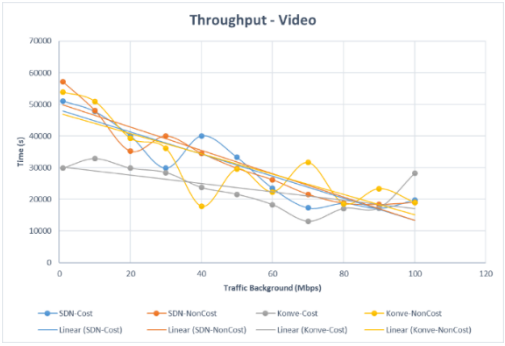
3. Throughput



Gambar 15. Grafik Throughput Trafik Data



Gambar 16. Grafik Throughput Trafik VoIP



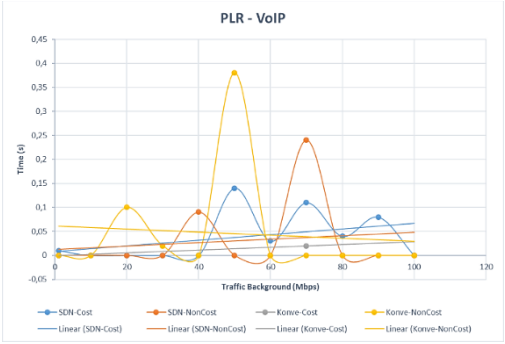
Gambar 17. Grafik Throughput Trafik Video

Pada Gambar 15-17, Hasil pengukuran rata–rata *throughput* tersebut dikarenakan kapasitas *link* yang dilalui setiap paket layanan data, voip, dan video sebesar 100 Mbps sehingga ketika diberi *background traffic* 1–100 Mbps, semua paket layanan masih dapat dilewatkan dan *throughput* yang didapat mendekati optimal untuk kedua paket layanan yakni data dan voip. Hal tersebut menunjukkan bahwa setiap penambahan *traffic background* tidak terlalu berpengaruh terhadap kedua paket tersebut, karena keduanya cenderung stabil dan mengalamai tren kenaikan

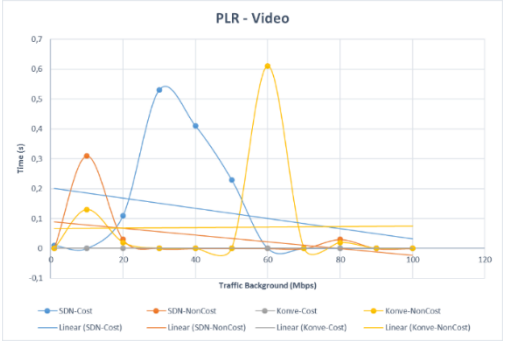
yang stabil. Namun hal berbeda terjadi pada trafik video, keempat topologi mengalami tren menurun seiring dengan penambahan *background traffic* tersebut. Berdasarkan hasil pengukuran rata – rata *throughput* untuk layanan data, voip, dan video dengan *background traffic* 1 sampai 100 Mbps yang didapat tersebut sesuai dan mendekati nilai *throughput* yang dikonfigurasikan pada D-ITG kecuali paket video yang optimal hanya saat 1-10 Mbps (masing–masing standar paket layanan, yaitu untuk layanan data dengan *throughput* sekitar 38,4 Kbps, untuk voip dengan *throughput* sekitar 70,4 Kbps, dan untuk video dengan ideal *throughput* sekitar 5,336 Mbps).

4. *Packet Lost Ratio*

Gambar 18. Grafik PLR Trafik Data



Gambar 19. Grafik PLR Trafik VoIP

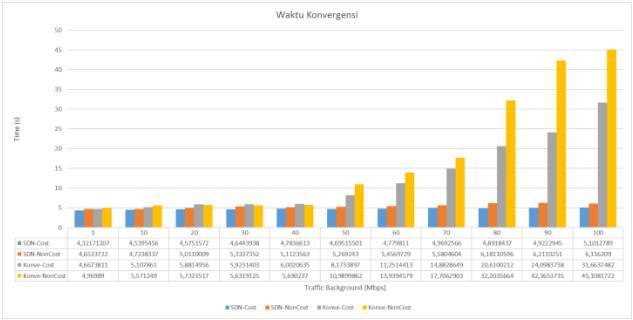


Gambar 20. Grafik PLR Trafik Video

Pada Gambar 18-29, Hasil pengukuran rata – rata *packet loss ratio* relatif baik dikarenakan kapasitas *link* yang dilalui setiap paket layanan data, voip, dan video sebesar 100 Mbps sehingga

setiap paket layanan data, voip, dan video masih dapat dilewatkan suatu link ketika diberi trafik *background* 1 – 100 Mbps, sehingga nilai PLR dapat berkisar di ± 0%. Meningkatnya *packet loss ratio* seperti gambar di atas juga berimplikasi terhadap hasil parameter lainnya, seperti *delay, throughput,* dan *jitter*. Berdasarkan hasil pengukuran rata – rata *packet loss ratio* untuk layanan data, voip, dan video dengan trafik *background* 0 – 100 Mbps yang didapat tersebut masih memenuhi kriteria dan standar ITU-T G.1010 [10].

B. Waktu Konvergensi.



Gambar 21. Perbandingan Waktu Konvergensi 1 link failure

Pada Gambar 21, Hasil pengukuran rata - rata waktu konvergensi OSPF pada jaringan SDN dan konvensional baik yang menggunakan *cost* ataupun tidak dapat mengindikasikan bahwa setiap penambahan atau peningkatan *background traffic* berpengaruh terhadap waktu konvergensi. Pengukuran waktu konvergensi di atas dilakukan dengan pengujian 1 *link failure*. Perbedaan waktu konvergensi dapat terlihat jelas saat *background traffic* dilakukan penambahan sebesar 50 Mbps untuk jaringan konvensional khususnya. Hal tersebut diketahui karena adanya pengaruh nilai *delay* [11]. Korelasi yang terdapat disini bahwa pengujian nilai *delay* sebelumnya terhadap masing-masing topologi menunjukkan peningkatan nilai *delay* terhadap *background* traffic meskipun sangat kecil, maka pemberian *background* traffic yang ada pada pengujian waktu konvergensi juga memiliki pengaruh. Pada penelitian [11] dikatakan bahwa nilai waktu konvergen yang dihasilkan SDN lebih kecil karena seluruh proses pembaruan *link state* dilakukan melalui *controler*, sehingga nilai *delay* tidak menjadi sangat berpengaruh terhadap waktu konvergen yang terjadi pada SDN. Hal lainnya menununjukkan bahwa penggunaan *cost* memperlihatkan hasil yang lebih kecil dan stabil dibanding dengan jaringan yang menggunakan modifikasi *cost,* karena penetapan nilai *cost* yang menjadi salah satu penentu pemilihan rute telah mempersingkat waktu konvergen secara tidak langsung. Hal tersebut juga berlaku pada jaringan konvensional yang menggunakan *cost* yang memiliki waktu konvergen lebih cepat dibanding dengan yang tidak menggunakan.

4. KESIMPULAN

Penerapan modifikasi konfigurasi *cost* pada jaringan SDN memiliki pengaruh terhadap kinerja, dilihat berdasar QoS dan Waktu Konvergensi. Pada jaringan SDN yang menggunakan nilai *cost* memiliki nilai yang lebih rendah dibandingkan dengan nilai *delay* dari topologi lainnya. Hasil pengukuran rata – rata jitter untuk layanan data, voip, dan video dengan trafik background 0 – 90 Mbps yang didapat tersebut memenuhi kriteria dan standar ITU-T G.1010. Waktu konvergensi dengan 1 *link failure* terendah didapatkan oleh topologi SDN yang menggunakan konfigurasi *cost.* Penerapan modifikasi konfigurasi *cost* pada jaringan SDN memiliki hasil yang lebih baik secara keseluruhan, seperti nilai delay dan jitter yang rendah, dan

waktu konvergensi yang cepat. Saran penelitian selanjutnya bisa mengembangkan algoritma tertentu yang bisa membuat pemilihan jalur data lebih efektif dan efisien.

DAFTAR PUSTAKA

[1] APJII, "Profil Pengguna Internet di Indonesia," Asosiasi Penyelenggara Jasa, Jakarta, 2014.

[2] Q. H. a. K. B. Fei Hu, "A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation," IEEE communication surveys & tutorials, vol. 16, no. IEEE, 2014.

[3] "Mininet Overview," [Online]. Available: http://mininet.org/overview/. [Diakses 17 November 2016].

[4] S. M. Anggara, "Pengujian Performa Kontroler Software-defined Network (SDN): POX dan Floodlight," STEI ITB, Yogyakarta, 2015.

[5] F. ADNANTYA, "Simulasi dan Analisis Kinerja Protokol Ruting eBGP pada SDN (Software Defined Network)," Universitas Telkom, Bandung, 2015.

[6] A. RAHMANTO, "SIMULASI DAN ANALISIS KINERJA PROTOKOL ROUTING RIP PADA SDN (SOFTWARE DEFINED NETWORK)," Universitas Telkom, Bandung, 2015.

[7] S. N. H. R. M. N. Abu Riza Sudiyatmoko, "Analisis Performansi Perutingan Link State Menggunakan Algoritma Djikstra Pada Platform Software Defined Network (SDN)," Jurnal Infotel, vol. 1, no. 1, pp. 40-46, 2016.

[8] A. V. B. I. Brayan Anggita Linuwih, "PERANCANGAN DAN ANALISIS SOFTWARE DEFINED NETWORK PADA JARINGAN LAN : PENERAPAN DAN ANALISIS METODE PENJALURAN PATH CALCULATING MENGGUNAKAN," Bandung, 2016.

[9] N. Abdillah, "ANALISIS PERFORMA ARSITEKTUR SOFTWARE DEFINED NETWORK DENGAN OPENFLOW PADA MIKROTIK RB750," Yogyakarta, 2016.

[10] T. H. Cormen, C. E. Leiserson, R. L. Rivest dan C. and Stein, Introduction to Algorithms., Cambridge, Massachusetts: MIT Press. 2nd edition, 2001.

[11] H. Zhang dan Y. Jinyao, "Performance of SDN Routing in Comparison with Legacy," dalam International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, 2015.