

ANALISIS PERFORMANSI *SEGMENT ROUTING* PADA *SOFTWARE DEFINED NETWORK* MENGUNAKAN KONTROLER ONOS

PERFORMANCE ANALYSIS OF *SEGMENT ROUTING* IN *SOFTWARE DEFINED NETWORK* USING *ONOS CONTROLLER*

Safida Reynita Sari¹, Dr.Ir Rendy Munadi, M.T.², Danu Dwi Sanjoyo, S.T.,M.T³

^{1,2,3}Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

safidars@gmail.com, rendymunadi@telkomuniversity.ac.id, danudwj@telkomuniversity.ac.id

Abstrak

Jaringan SDN merupakan sebuah arsitektur yang baru pada sebuah jaringan yang memisahkan fungsi *control plane* dan *data plane*. Sehingga dalam pengelolaannya bisa menjadi lebih efisien dan sederhana dibandingkan pada jaringan konvensional. Salah satu parameter untuk menyediakan efisiensi pada jaringan SDN adalah dalam pemilihan jalur yang digunakan. Salah satunya menggunakan metode *segment routing*, dengan adanya *segment routing* nantinya dapat mengurangi aturan – aturan yang terdapat pada *forwarding table*, sehingga proses pengiriman data menjadi lebih efisien. Pada penelitian kali ini akan disimulasikan penggunaan *segment routing* dan membandingkannya dengan performansi jaringan tanpa menggunakan aturan *segment routing* pada SDN dengan menggunakan ONOS sebagai kontrolernya. Topologi yang digunakan adalah *leaf spine*. Parameter yang akan diukur adalah *resource utilization* dan QoS. Pada Penelitian ini penggunaan memori pada *segment routing* sebesar 84% dan 50% untuk tanpa *segment routing*. Untuk penggunaan CPU 32% untuk *segment routing* dan 54% untuk tanpa *segment routing*. Untuk *packet loss* pada layanan VoIP dan video ketika terjadi pengiriman secara bersamaan pada jaringan dengan menggunakan *segment routing* sebanyak 1,2% sedangkan tanpa menggunakan *segment routing* sebanyak 1,3%. Pada pengujian *delay*, *throughput* dan *jitter* pada layanan VoIP dan video menggunakan *segment routing* lebih besar nilainya dibandingkan dengan tanpa *segment routing*

Kata Kunci : SDN, ONOS, Segment routing, Resource utilization dan QoS

Abstract

SDN network is a new architecture on a network that separates the control plane function and data plane. So that in its management can be more efficient and simpler than conventional networks. One of the parameters to provide efficiency on an SDN network is in the selection of the path used. One of them uses the *segment routing* method, with the *segment routing* later can reduce the rules contained in the *forwarding table*, so that the process of sending data becomes more efficient. In this research we will simulate the use of *segment routing* and compare them with network performance without using the *segment routing* rules on SDN using ONOS as the controller. The topology used is *leaf spine*. The parameters to be measured are *resource utilization* and QoS. In this research memory usage in the *segment routing* is 84% and 50% for without *segment routing*. For CPU usage 32% for *segment routing* and 54% for without *segment routing*. For *packet loss* on VoIP and video services when using *segment routing* as much as 1,2% while without using *segment routing* as much as 1,3%. In testing the *delay*, *throughput* and *jitter* on VoIP and video services using the *segment routing* the value is greater than without *segment routing*

Keywords: SDN, ONOS, Segment routing, Resource utilization and QoS

1. Pendahuluan

Saat ini SDN (*Software Defined Networking*) menjadi topik yang populer dikalangan peneliti. SDN merupakan sebuah arsitektur baru pada jaringan yang memisahkan *control plane* dengan *data plane* dalam sebuah jaringan. Pada SDN hanya perlu kontroler yang dirubah jika ada pembaruan sistem. Kontroler yang mendukung protokol OpenFlow pada SDN adalah ONOS, POX, Opendaylight dll [1]. Pada SDN saat tahap pengiriman, sebuah paket harus memilih jalur yang akan dilewatinya. Salah satunya adalah menggunakan *segment routing*. Pada *segment routing* jalur informasi dilewatkan melalui daftar urutan multi-protokol label switching (MPLS), kemudian pada *segment routing* ini juga terdapat *segment ID* yang nantinya proses labeling pada MPLS akan disatukan dengan proses *segment* sehingga menjadikan sebuah jaringan menjadi lebih cepat dan efisien. Pada penelitian sebelumnya yang berjudul “An Efficient Routing Algorithm Based on *Segment routing* in Software Defined Networking” [11], sudah dilakukan penelitian tentang *segment routing* dengan konsep SDN untuk melihat performansinya. Namun belum dibandingkan dengan jaringan yang tanpa adanya *segment routing* pada SDN menggunakan kontroler ONOS

untuk melihat performansinya. Oleh karena itu, pada penelitian ini diusulkan untuk melakukan penelitian tugas akhir mengenai analisis performansi *segment routing* pada SDN dengan ONOS sebagai kontrolernya.

2. Dasar Teori

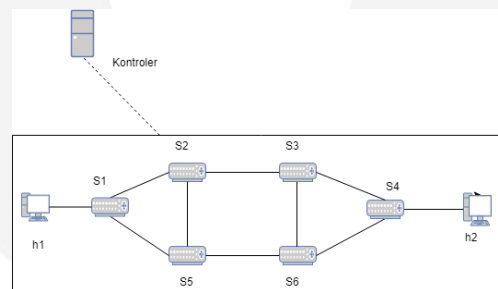
2.1 Software Defined Network

Software Defined Network (SDN) adalah paradigma jaringan yang muncul yang memberi harapan untuk mengubah keterbatasan infrastruktur jaringan saat ini yaitu memecah integrasi vertikal dengan memisahkan *control plane* dan *data plane*. Dengan pemisahan *control plane* dan *data plane*, switch jaringan menjadi perangkat *forwarding* yang sederhana dan *control plane* diimplementasikan sebagai pengendali terpusat yaitu menjadi lebih mudah untuk mengkonfigurasi jaringan dan dapat mengurangi kerawanan kesalahan untuk memodifikasi kebijakan jaringan [1].

2.2 Segment routing

Segment routing adalah teknologi jaringan yang menawarkan metode baru penerusan paket yang meminimalkan kebutuhan untuk menjaga banyaknya status informasi jaringan [11]. *Segment routing* mendefinisikan jalur informasi melalui daftar urutan multi-protokol label switching (MPLS) mekanisme pada header paket, kemudian juga terdapat Segment ID atau Group ID. Pada segmen ID atau group ID juga terdapat SR-MPLS SID yang merupakan label MPLS atau nilai indeks ke dalam ruang label MPLS secara eksplisit dikaitkan dengan segmen [13]. Daftar pengidentifikasi segmen (SID) mewakili jalur dalam jaringan, yang merupakan kandidat jalur terbaik yang dipilih dari banyak kandidat jalur yang dipelajari dari berbagai sumber. Group ID sendiri bisa terdiri atas beberapa perangkat [13].

Cara kerja dari *segment routing* dapat dilihat pada Gambar 1. Kontroler menghitung jalur dengan modul routing, sehingga mengkonfigurasi tabel *forwarding* yang masuk dengan daftar segmen yang dipesan. Pada gambar tersebut terjadi pengiriman paket dari h1 ke h2. Informasi jalur yang didapat (h1-S1-S2-S3-S4-h2) dikodekan dalam header paket dengan label MPLS, yang berakibat menghilangkan aturan pemansangan di switch di sepanjang jalan. Pada switch 4 merupakan label MPLS 1004. sehingga paket akan dikirimkan dengan label tujuan 1004. Pertama host 1 mengirim paket melalui switch 1 kemudian switch 1 memproses header label yang merupakan node segmen, dan kemudian meneruskan paket dengan jalur terpendek menuju perangkat jaringan dengan group ID 19 yang dimana label tersebut akan meneruskan paket ke label 1004. Pada group ID 19 terdapat dua perangkat switch yang harus dilalui yaitu switch 2 dan switch 3. Oleh karena itu switch 2 dan 3 meneruskan paket menuju MPLS 1004 yang merupakan switch 4. Setelah tiba di switch 4, switch meneruskan paket ke host 2 yang merupakan tujuan paket tersebut dikirimkan. Dengan demikian, *segment routing* dapat mengurangi jumlah aturan *forwarding* dalam penggunaannya.



Gambar 1 *Segment routing*[11]

2.2 ONOS

Komponen utama dari Software Defined Networking adalah SDN kontroler, yang disebut sistem operasi jaringan. Pengendali inilah yang mendefinisikan sifat paradigma SDN. Kontroler bertanggung jawab untuk memusatkan komunikasi dengan semua elemen jaringan yang dapat diprogram, memberikan tampilan jaringan yang terpadu [1]. ONOS (*Open Network Operating System*) adalah sistem operasi jaringan SDN open source terkemuka untuk membangun solusi SDN generasi selanjutnya yang ditargetkan secara khusus di *Service Provider* dan *Mission Critical Network*. ONOS dibangun untuk memberikan ketersediaan tinggi (*High Availability*), *scale-out*, dan performansi kinerja jaringan yang dibutuhkan. Kontroler ini berbasis java.

2.3 Mininet

Mininet adalah sistem yang memungkinkan *prototyping* cepat dengan jaringan yang besar pada satu komputer. Ada banyak alasan yang memicu penggunaan emulator yang baru yang disebut Mininet. Pertama, hanya ada sedikit perangkat jaringan yang tersedia untuk tujuan penerapan SDN standar karena belum meluasnya teknologi dari perspektif industri. Selain itu, implementasi jaringan dengan jumlah besar mengakibatkan perangkat jaringan sangat sulit dan mahal. Oleh karena itu, untuk mengatasi masalah tersebut, strategi model secara virtual telah dilakukan dengan tujuan untuk membuat *prototype*.

2.4 Parameter Uji

2.4.1 Packet loss

Packet loss merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang.

$$\text{Packet Loss} = \frac{\text{paket yang dikirim} - \text{paket yang diterima}}{\text{paket yang dikirim}} \times 100\% \quad [9]$$

2.4.2 Delay

Delay adalah waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. *Delay* dapat dipengaruhi oleh jarak, media fisik, kongesti atau juga waktu proses yang lama [9].

2.4.3 Jitter

Jitter biasa disebut dengan variasi *delay* pada taransmisi data di jaringan. *Jitter* diakibatkan oleh variasi-variasi dalam panjang antrian, dalam waktu pengolahan data, dan juga dalam waktu penghimpunan ulang paket-paket di akhir perjalanan.

2.4.4 Throughput

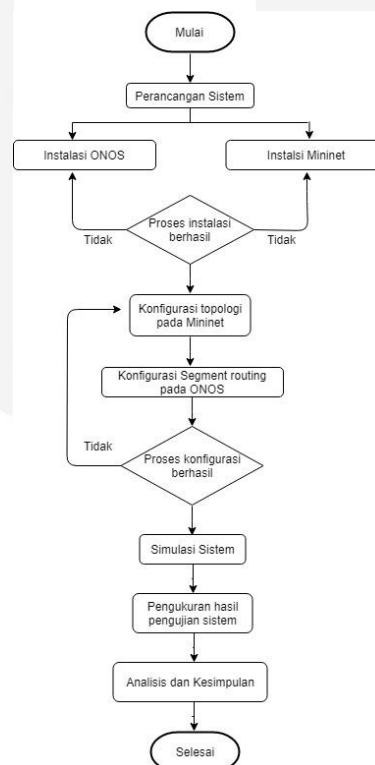
Throughput yaitu kecepatan (*rate*) transfer data efektif, yang diukur dalam bps (*bit per second*).

$$\text{Throughput} = \frac{\text{Jumlah paket yang sukses}}{\text{waktu pengiriman}} \quad [9]$$

2.4.5 Resource utilization

Resource utilization untuk mengetahui konsumsi memori kontroler dan konsumsi CPU selama kontroler digunakan. Pengukuran dilakukan saat kontroler mulai dijalankan hingga jaringan berada dalam kondisi stabil.

2.5 Perancangan Sistem

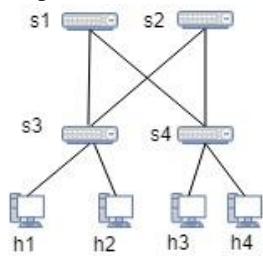


Gambar 2 Diagram perancangan system

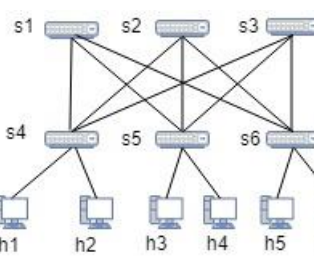
Perancangan sistem simulasi yang akan dibentuk seperti pada **Gambar 2**. Awalnya pada perancangan sistem akan dijelaskan bagaimana topologi SDN yang akan digunakan dan komponen apa saja yang ada dalam topologi tersebut. Kemudian instalasi Mininet dan Kontroler sebagai fungsi control plane dan data plane. Jika proses instalasi berhasil maka akan diteruskan untuk mengkonfigurasi topologi (data plane) pada Mininet dan konfigurasi

Kontroler, kemudian jika konfigurasi yang dijalankan berhasil maka akan dilanjutkan dengan proses simulasi dengan parameter *packet loss*, *delay*, *jitter* dan *resource utilization*. Setelah pengambilan data selesai maka akan dianalisa performansi dari *segment routing* pada masing - masing skenario.

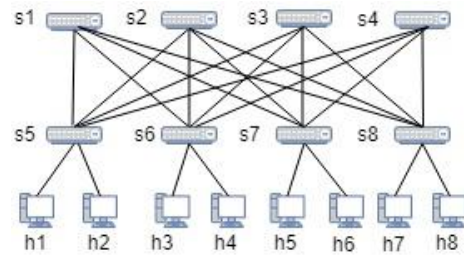
Topologi yang digunakan adalah topologi leaf spine yang terdiri atas 8 switch, 6 switch dan 4 switch, seperti pada **Gambar 3**, **Gambar 4** dan **Gambar 5**.



Gambar 3 Topologi 4 switch



Gambar 4 Topologi 6 switch



Gambar 5 Topologi 8 switch

2.6 Desain Perangkat

Perangkat keras yang digunakan untuk penelitian ini terdiri dari satu buah laptop, spesifikasi perangkat keras sebagai berikut:

Tabel 1 Perangkat keras yang digunakan

| Komponen Laptop | Spesifikasi |
|-----------------|--|
| Processor | Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz |
| RAM | 6.00 GB (5.71 GB usable) |

Perangkat lunak yang digunakan untuk membuat simulasi dari penelitian ini dijelaskan sebagai berikut:

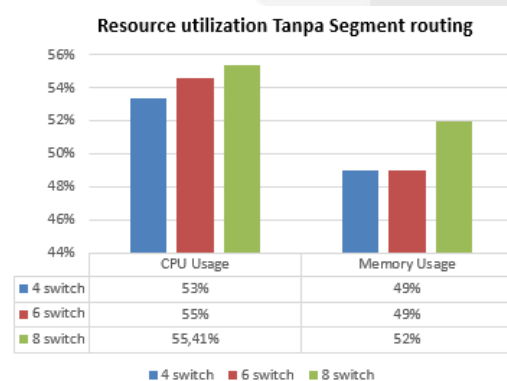
Tabel 2 Perangkat lunak yang digunakan

| Tipe | Keterangan |
|-------------------|-------------------------|
| Sistem Jaringan | Ubuntu 14.04 LTS 64 bit |
| Emulator Jaringan | Mininet |
| Kontroler SDN | ONOS |
| Perekaman Trafik | D-ITG 2.8.1 |
| Monitoring CPU | sysstat |

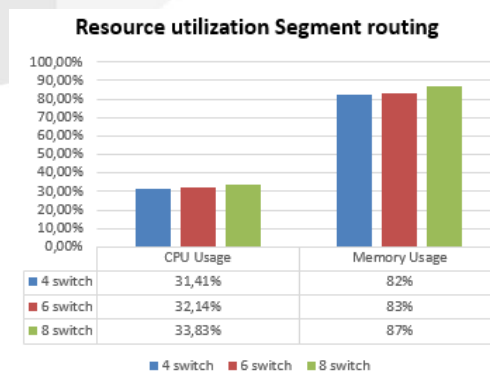
3. Pembahasan

3.1 Pengujian Resource Utilization

Pengujian ini dilakukan untuk mengetahui konsumsi penggunaan CPU dan konsumsi memori dengan jaringan yang menggunakan *segment routing* dan tanpa *segment routing*. Pengukuran dilakukan dari kontroler dinyalakan hingga jaringan stabil



Gambar 4 Resource utilization tanpa segment routing

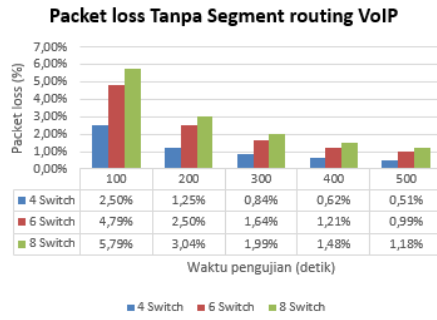


Gambar 5 Resource utilization segment routing

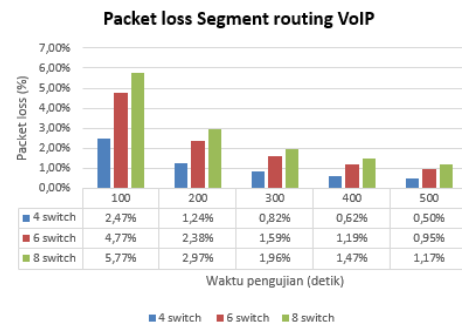
Berdasarkan hasil diatas penggunaan CPU dengan *segment routing* jauh lebih kecil dibandingkan dengan tanpa *segment routing*. Kemudian pada konsumsi memori dengan *segment routing* jauh lebih besar jika dibandingkan dengan tanpa *segment routing*.

3.2 Pengujian Packet loss

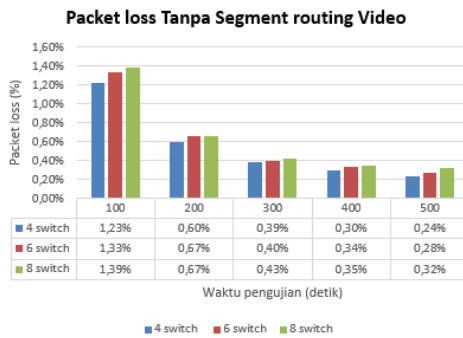
Pengujian *packet loss* dilakukan untuk mengetahui paket yang tidak terkirim ketika terjadi pengiriman paket. Hasil pengujian dapat dilihat pada grafik dibawah



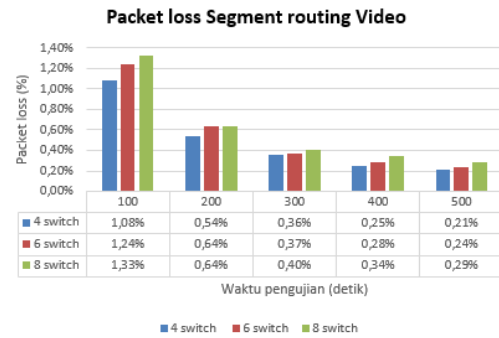
Gambar 6 Packet loss tanpa segment routing VoIP



Gambar 7 Packet loss segment routing VoIP



Gambar 8 Packet loss tanpa segment routing video

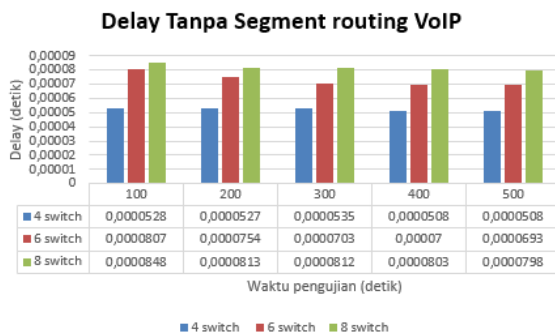


Gambar 9 Packet loss segment routing video

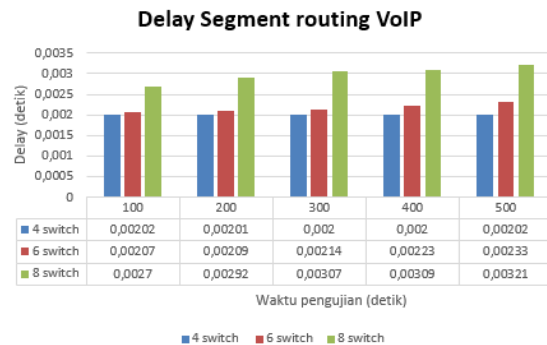
Secara umum pada pengujian *packet loss* ini dengan menggunakan *segment routing* lebih kecil *packet loss*nya dari pada tanpa *segment routing*, baik pada layanan VoIP maupun video. Hal ini disebabkan karena pada *segment routing* pada saat pengiriman data tidak melalui satu jalur saja melainkan ada beberapa jalur sehingga tidak terjadi kepadatan pada jalur yang digunakan saat melakukan pengiriman dalam satu waktu.

3.3 Pengujian Delay

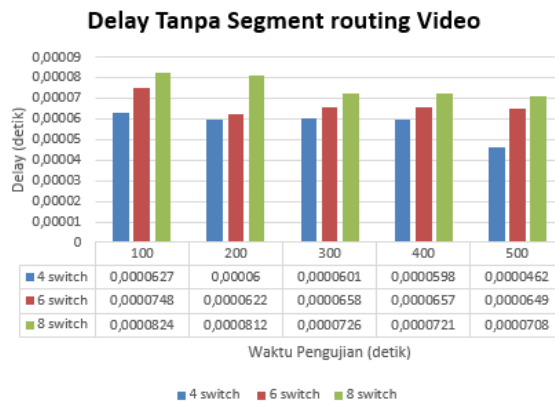
Delay adalah waktu yang dibutuhkan data untuk menempuh jarak dari sumber hingga sampai ke tujuan. Hasil dari pengujian ini dapat dilihat pada grafik berikut



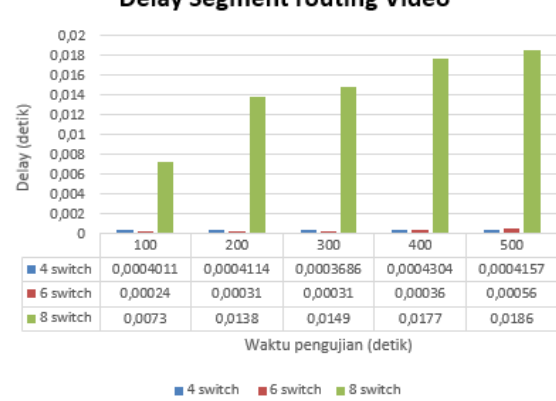
Gambar 10 Delay tanpa segment routing VoIP



Gambar 11 Delay segment routing VoIP



Gambar 12 Delay tanpa segment routing video

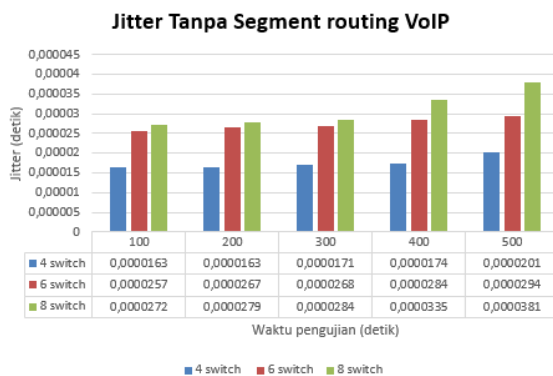


Gambar 13 Delay segment routing video

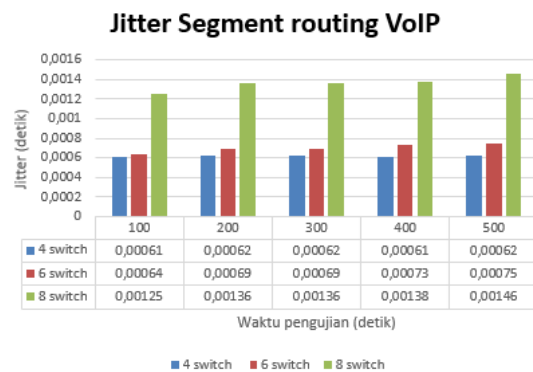
Secara umum dapat disimpulkan bahwa *delay* dengan tanpa *segment routing* lebih kecil daripada menggunakan *segment routing*. Hal ini disebabkan karena pengiriman data pada *segment routing* itu disebarkan ke semua jalur yang tersedia sehingga hal itu menyebabkan *delay*nya semakin besar nilainya dibanding dengan tanpa *segment routing* yang hanya menggunakan satu jalur pengiriman. Pada pertambahan waktu pengiriman dapat disimpulkan bahwa tidak menunjukkan perbedaan karena jumlah pengiriman data pada setiap pertambahan waktu berbanding lurus.

3.4 Pengujian Jitter

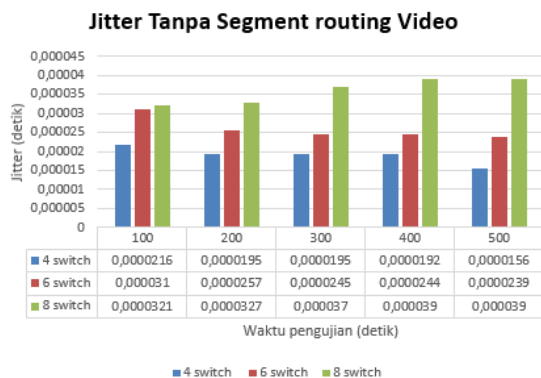
Jitter adalah suatu metode pengukuran ntuk melihat variasi *delay* pada taransmisi data di jaringan. Hasil pengujiannya dapat dilihat pada gambar berikut



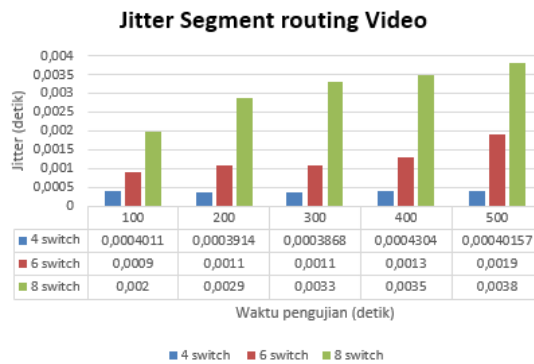
Gambar 14 Jitter tanpa segment routing VoIP



Gambar 15 Jitter segment routing VoIP



Gambar 16 Jitter tanpa segment routing video

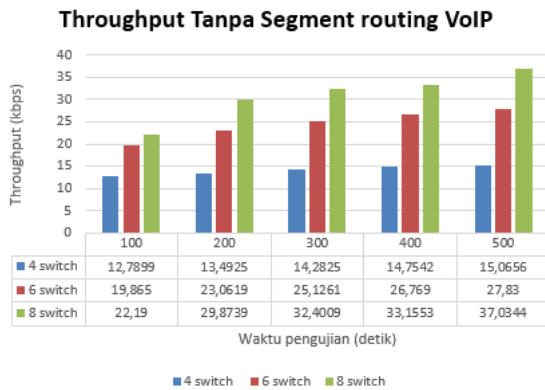


Gambar 17 Jitter segment routing video

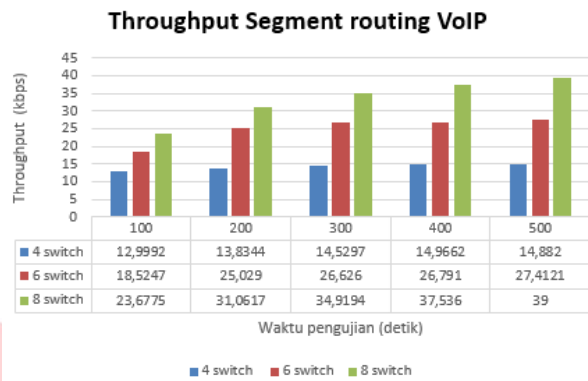
Secara umum nilai *jitter* dengan *segment routing* jauh lebih besar dibandingkan dengan tanpa *segment routing*. Hal ini terjadi karena dampak dari *segment routing* yang menggunakan semua jalur yang terhubung untuk mengirim sebuah paket. Sehingga jarak antara *delay* pertama dan *delay* selanjutnya menjadi lebih besar. Kemudian pada setiap pertambahan jumlah switch pada layanan VoIP dan video mengalami peningkatan. Untuk setiap pertambahan waktu pengiriman, nilai *jitter* menunjukkan persamaan, hal ini disebabkan karena jumlah paket yang dikirim semakin bertambah dengan waktu pengiriman semakin bertambah pula.

3.5 Pengujian Throughput

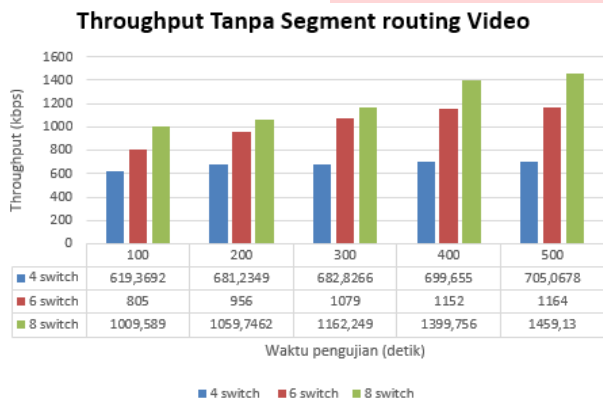
Throughput merupakan jumlah total kedatangan paket yang sukses yang diamati pada penerima selama interval waktu tertentu. Hasil pengujiannya dapat dilihat pada gambar dibawah ini.



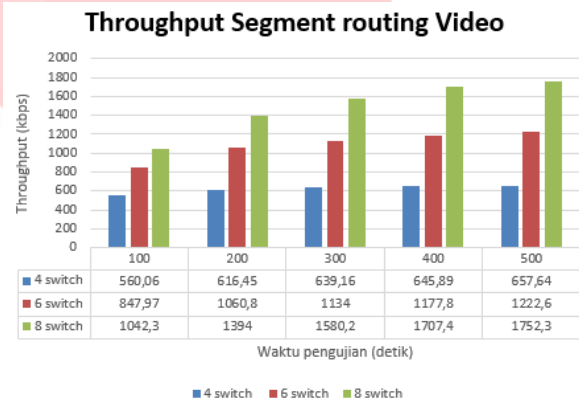
Gambar 18 Throughput tanpa segment routing VoIP



Gambar 19 Throughput segment routing VoIP



Gambar 20 Throughput tanpa segment routing video



Gambar 21 Throughput segment routing video

Secara umum nilai *throughput* pada layanan VoIP dan video dengan *segment routing* lebih besar dibandingkan tanpa *segment routing*. Hal itu disebabkan karena jalur yang digunakan *segment routing* untuk mengirim data lebih dari satu maka peluang paket yang sukses dikirim makin banyak pula. Pertambahan jumlah switch juga memberi dampak nilai *throughput* yang makin besar.

4. Kesimpulan

Dari pengujian dan analisis yang telah dilakukan dapat dibuat kesimpulan sebagai berikut:

1. Berdasarkan pengujian diketahui *Resource utilization* untuk penggunaan memori pada *segment routing* sebesar 84% dan 50% untuk tanpa *segment routing*. Untuk penggunaan CPU 32% untuk *segment routing* dan 54% untuk tanpa *segment routing*. Sehingga dengan menggunakan *segment routing* lebih efisien dalam penggunaan CPU namun tidak efisien dalam penggunaan memori..
2. Berdasarkan pengujian ini diketahui bahwa paket yang hilang pada layanan VoIP dan video ketika terjadi pengiriman secara bersamaan pada jaringan dengan menggunakan *segment routing* rata – rata sebanyak 1,2% sedangkan tanpa menggunakan *segment routing* sebanyak 1,3%. Sehingga *segment routing* lebih baik untuk *packet lossnya*.
3. Berdasarkan pengujian *delay*, *throughput* dan *jitter* pada layanan VoIP dan video menggunakan *segment routing* lebih besar nilainya dibandingkan dengan tanpa *segment routing* .
4. Berdasarkan hasil keseluruhan dari penelitian ini, menggunakan *segment routing* sebagai metode *forwarding* lebih baik daripada tanpa menggunakan *segment routing* jika dilihat dari penggunaan CPU dan *packet lossnya*.

DAFTAR PUSTAKA

- [1] D Kreutz, FMV Ramos, PE Verissimo, "Software Defined Networking A Comprehensive Survey" Proceedings of the IEEE, 2015
- [2] Fei Hu, Qi Hao, Ke Bao, "A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation" IEEE Communication Surveys & Tutorials, Vol. 16, No. 4, 2014
- [3] Wenfeng Xia, dkk, "A Survey on Software-Defined Networking" IEEE Communication Surveys & Tutorials, Vol 17, No 1, 2015
- [4] Junjie Pang, dkk "SDN-Based Data Center Networking With Collaboration of Multipath TCP and *Segment routing*" IEEE Access, Vol 5, 2017.
- [5] BAA Nunes, dkk, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks" IEEE Communication Surveys & Tutorials, Vol 16, 2014
- [6] Open Networking Foundation, "OpenFlow Switching Specification Version 1.3.0" 2012.
- [7] "Software-Defined Networking: The New Norm for Networks" Open Networking Foundation WhitePaper, 2012
- [8] Mohammad Alizadeh, Tom Edsal, "On the Data Path Performance of Leaf-Spine Data center Fabrics" IEEE 21st Annual Symposium on High-Performance Interconnects, 2013
- [9] F Ketih, S Askar, "Emulation of Software Defined Networks Using Mininet in Different Simulation Environments" Modelling and Simulation IEEE, 2015
- [10] "Introducing ONOS - a SDN network operating system for Service Providers" ON.LAB WhitePaper, 2014
- [11] M.-C. Lee and J.-P. Sheu, "An efficient routing algorithm based on *segment routing* in software-defined networking," Comput. Netw., vol. 103, pp. 44–55, Apr. 2016.
- [12] C Filsfils dkk "The *Segment routing* Architecture" IEEE Global Communications Conference (GLOBECOM), 2015
- [13] S. Previdi, Ed. dkk (2018, Juli). *Segment routing* Architecture. Dikutip 20 Agustus 2019 dari <https://tools.ietf.org/html/rfc8402>