

IMPLEMENTASI PROTOKOL ROUTING EBGP PADA SOFTWARE DEFINED NETWORK BERBASIS ROUTEFLOW

Henky Agie Friwansya¹, Indrarini Dyah Irawati, ST., MT.², Yuli Sun Hariyani, ST., MT.³

Prodi D3 Teknik Telekomunikasi, Fakultas Ilmu Terapan, Universitas Telkom

¹henkyagie123@gmail.com, ²indrarini@telkomuniversity.ac.id, ³yulisun@tass.telkomuniversity.ac.id

Abstrak

Meningkatnya penggunaan Internet menjadi tantangan bagi para peneliti untuk mempertahankan kinerja jaringan. Salah satu cara untuk mempertahankan kinerja jaringan tersebut adalah melalui mekanisme *routing* karena *routing* merupakan bagian utama dalam memberikan suatu performansi dalam jaringan. Semakin besar suatu jaringan maka semakin kompleks juga konfigurasi yang diperlukan. *Software defined network* (SDN) merupakan konsep yang memisahkan *data plane* dengan *control-plane* ke dalam sebuah perangkat tersendiri yang dikenal *controller*. Sehingga perangkat jaringan (*router*, *switch*) hanya meneruskan perintah dari *controller*. BGP adalah protokol *routing* yang berfungsi menghubungkan semua domain jaringan yang ada di Internet. Namun seperti protokol *routing* lainnya, protokol tersebut masih berjalan pada perangkat jaringan tradisional, yang mana *control plane* dan *data plane* nya berada dalam satu perangkat.

Dalam proyek akhir ini dilakukan penerapan *routing Border Gateway Protocol* (BGP) menggunakan mekanisme *routing external* BGP berbasis *RouteFlow* pada *Software Defined Network*. Dapat dibuktikan pada jaringan SDN dengan melakukan implementasi jaringan dengan menggunakan 5 buah *switch* yang telah *support* teknologi SDN dan setiap *switch* memiliki *autonomous system* yang berbeda untuk menghubungkan antar host sebagai *source* dan *destination*. Serta mengetahui jalur yang digunakan untuk mentransmisikan paket data dari pengirim ke penerima. Serta dilakukan skenario pemutusan jalur pada jaringan untuk membuktikan kinerja *routing* BGP.

Hasil pengujian performansi penerapan *routing* BGP pada jaringan SDN berbasis *RouteFlow* menunjukkan bahwa nilai *convergece time* 163.441 detik untuk implementasi. Nilai rata – rata QoS pada implementasi menghasilkan nilai *throughput* sebesar 1.049 Mbps, *delay* sebesar 69.03 ms, *jitter* sebesar 0.315 ms, dan hasil dari *packet loss* 0%.

Kata kunci : *software defined network*, *eBGP*, *openflow*, *routeflow*, *convergence time*

Abstract

The increasing use of the Internet is a challenge for scientists to maintain network performance. One way to maintain the performance of the network is through the routing mechanism because routing is the main part in providing a performance in the network. The larger the network the more complex the configuration is required. Software defined network (SDN) is a concept that separates plane data with control-plane into a device of its own known controller. So the network devices (routers, switches) just forward the command from the controller. BGP is a routing protocol that works to connect all network domains on the Internet. But like any other routing protocol, the protocol is still running on the traditional network devices, where the control plane and its data plane are in one device.

In this final project implementation of Border Gateway Protocol (BGP) routing using RouteFlow based BGP routing mechanism in Software Defined Network. Can be proven on the SDN network by implementing the network using 5 switches that have support SDN technology and each switch has a different autonomous system to connect between hosts as source and destination. And know the path used to transmit data packets from sender to receiver. As well as scenario disconnection scenarios on the network to prove the performance of BGP routing.

The result of BGP routing performance test on SDN network based on RouteFlow shows that the convergence time value is 163,441 seconds for implementation. The average value of QoS in the implementation resulted in a throughput value of 1.049 Mbps, delay of 69.03 ms, jitter 0.315 ms, and packet loss 0%.

Keywords: *software defined network*, *eBGP*, *openflow*, *routeflow*, *convergence time*

1. Pendahuluan

1.1 Latar Belakang

Peningkatan kebutuhan Internet menyebabkan semakin banyaknya tantangan yang dihadapi untuk mempertahankan kinerjanya. Salah satu cara untuk mengatasi hal tersebut adalah meningkatkan kinerja ruting pada jaringan Internet. Ruting adalah mekanisme untuk memindahkan suatu paket ke titik tujuan berdasarkan protokol ruting yang digunakan. Karena Internet mempunyai skala yang sangat besar, maka satu protokol ruting saja tidak bisa mengelola seluruh kegiatan ruting dalam jaringan Internet. *Border Gateway Protocol* (BGP) sebagai protokol ruting Internet menghubungkan semua jaringan-jaringan (*autonomous system*) di Internet. Sehingga BGP berperan penting untuk menjaga kinerja Internet karena peningkatan jumlah AS. *External BGP* (eBGP) merupakan salah satu mekanisme BGP yang terjadi ketika pertukaran informasi antar AS [3].

Dalam beberapa tahun terakhir banyaknya penelitian dan percobaan platform *Software Defined Network* (SDN) yang merupakan salah satu evolusi teknologi jaringan sesuai dengan tuntutan yang berkembang. Dibandingkan dengan jaringan konvensional, *Software Defined Networking* (SDN) memberikan kemudahan kepada pengguna dalam mengembangkan aplikasi pengontrol jaringan dengan memisahkan fungsi *data plane* dari *control plane* [3]. Pemisahan *data plane* dan *control plane* pada perangkat jaringan komputer seperti Router dan Switch memungkinkan untuk memprogram perangkat tersebut sesuai dengan yang diinginkan secara terpusat. Pemisahan inilah yang mendasari terbentuknya paradigma baru dalam jaringan komputer yang disebut *Software Defined Networking* (SDN) (US: Open Networking Foundation. 2013). Platform controller menyediakan Application Programming Interfaces (APIs) sehingga memudahkan dalam mengimplementasikan fitur dan layanan dalam jaringan komputer. Pada *Software Defined Networking* (SDN), controller terpusat mengkonfigurasi forwarding tabel (flow-table) Switch yang bertanggung jawab untuk meneruskan aliran paket komunikasi [4].

Penelitian yang telah dilakukan sebelumnya mengenai *Software Defined Networking* (SDN) oleh Ivan Hidayah menggunakan protokol *routing* RIP [6], Ayu Irmawati menggunakan protokol *routing* OSPF [8] dan Fahry Adnanya simulasi protokol *routing* BGP [3], maka dari itu penelitian tersebut dapat dikembangkan dengan mengimplementasikan *routing* BGP dengan mekanisme eBGP pada jaringan SDN.

Kehebatan teknologi SDN dalam jaringan komputer sangat menarik untuk dipelajari dan dikembangkan, sehingga menarik untuk dikembangkan dengan mengimplementasikan *routing* BGP berbasis RouteFlow. Dalam implementasi, diperlukan 5 buah switch yang telah *support* teknologi SDN dan masing – masing switch memiliki *autonomous system* yang berbeda untuk menghubungkan antar host sebagai *source*. dan *destination* dalam percobaan pengiriman paket *Internet Control Message Protocol* (ICMP) *ping* [8].

1.2 Tujuan dan Manfaat

Adapun tujuan proyek akhir dari implementasi Protokol Routing BGP pada jaringan SDN berbasis RouteFlow sebagai berikut :

- a. Dapat mengimplementasi jaringan *Software Defined Network* yang dapat melakukan fungsi *routing* BGP.
- b. Mengetahi penjaluran pada paket yang melewati topologi yang telah dibuat.
- c. Mendapatkan nilai *convergence time* dan QoS (*throughput*, *delay*, *jitter* dan *packet loss*) pada jaringan.

1.3 Rumusan masalah

Berdasarkan deskripsi latar belakang, maka dapat dirumuskan beberapa masalah dalam proyek akhir ini yaitu :

- a. Bagaimana cara menerapkan *routing* BGP berbasis *RouteFlow* yang digunakan pada jaringan SDN ?
- b. Bagaimana cara menggunakan POX *controller* dalam implementasi jaringan berbasis *Software Defined Network* ?
- c. Bagaimana cara konfigurasi OpenvSwitch dan TP-Link agar dapat digunakan untuk implementasi jaringan *Software Defined Network*?

1.4 Batasan Masalah

Untuk memperjelas ruang lingkup pembahasan, maka masalah yang dibahas dibatasi pada:

1. Untuk membangun jaringan diperlukan, lima perangkat *forwarding* (*switch* *OpenFlow*) dan sebuah *controller*.
2. Menggunakan *controller* POX sebagai *control plane* serta *OpenFlow* 1.0 sebagai protokol antarmuka yang menghubungkan *data plane* dan *control plane*.
3. Perangkat yang digunakan dalam implementasi jaringan ini adalah laptop tiga buah (dua sebagai *client* dan satu sebagai *controller*), dan lima perangkat TP-Link WR1043ND v2 sebagai fungsi *forwarding*.
4. Dalam proyek akhir ini dilakukan pemutusan *link* disesuaikan berdasarkan skenario untuk membuktikan kinerja protokol *routing* pada *routing* BGP berbasis *OpenFlow* serta pengukuran waktu konvergensi menggunakan paket ICMP *ping*.

1.5 Metodologi

Metodologi yang akan dilakukan pada proyek akhir ini adalah sebagai berikut:

1. Studi literatur
Pencarian informasi yang terkait bersumber dari buku, media, jurnal dan diskusi yang bertujuan menunjang selesainya Proyek Akhir ini.
2. Perancangan dan Implementasi protokol *routing* BGP pada jaringan
Melakukan perancangan dan pengimplementasian sistem sesuai dengan parameter yang diinginkan.
3. Analisa sistem
Mengamati hasil dari sistem yang dikerjakan sesuai dengan skenario yang telah ditetapkan serta menyimpulkan masalah yang ada.
4. Penarikan kesimpulan
Dari seluruh tahapan yang telah dilakukan diatas ditambah dengan masukan dari dosen pembimbing maka dapat diambil kesimpulan dari hasil yang telah dilakukan.

2. Dasar Teori

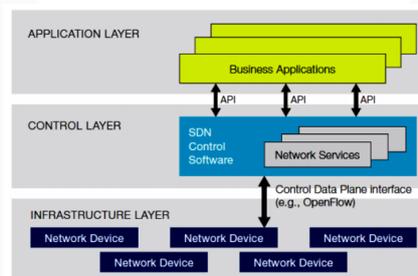
2.1 Software Defined Network (SDN)

Software Defined Networking (SDN) adalah sebuah konsep pendekatan jaringan komputer dimana sistem pengontrol dari arus data dipisahkan dari perangkat kerasnya. Awal mula terciptanya teknologi *Software Defined Networking* dimulai tidak lama setelah Sun Microsystems merilis Java pada tahun 1995, namun pada saat itu belum cukup membangunkan para periset untuk mengembangkan teknologi tersebut. Baru pada tahun 2008 *Software Defined Networking* ini dikembangkan di UC Berkeley and Stanford University. Dan kemudian teknologi tersebut mulai dipromosikan oleh *Open Networking Foundation* yang didirikan pada tahun 2011 untuk memperkenalkan teknologi SDN dan *OpenFlow* [8].

Pada teknologi jaringan konvensional, sistem pembuat keputusan kemana arus data dikirimkan dibuat menyatu dengan perangkat kerasnya. Namun di dalam teknologi SDN memiliki dua karakteristik. Pertama, SDN memisah antara *control plane* dari *data plane*. Kedua, SDN menggabungkan *control plane* setiap perangkat menjadi sebuah kontroler yang berbasis *programmable software*. Sehingga sebuah kontroler tersebut dapat mengontrol banyak perangkat dalam sebuah *data plane*. Di dalam SDN sebuah jaringan tersentralisasi dalam sebuah kontroler yang berbasis *software* yang dapat memelihara jaringan secara keseluruhan, Sehingga dapat mempermudah dalam mendesain dan mengoperasikan jaringan karena hanya melalui sebuah *logical point*.

2.2 Arsitektur Software Defined Network

Arsitektur SDN memiliki 3 layer yaitu *application layer*, *control layer*, dan *Infrastructure layer*. Ketiga layer tersebut mempunyai fungsi yang berbeda-beda. Berikut gambar arsitektur secara umum *Software Defined Network* [8]:



Gambar 2.1 Arsitektur *Software Defined Network*[8]

a) Application Layer

Application Layer merupakan layer yang berperan sebagai antar muka untuk memudahkan pengelola jaringan dalam melakukan fungsi konfigurasi, fungsi kontrol, dan fungsi evaluasi. Pada lapisan ini terbentuk dari integrasi dengan *control layer* yang memberikan informasi tentang jaringan yang selanjutnya ditampilkan dalam bentuk yang dapat dipahami oleh pengelola jaringan.

b) Control Layer

Control Layer merupakan lapisan yang berfungsi untuk mengontrol seluruh aktifitas jaringan. Dalam lapisan tersebut terdapat fungsi *control plane*.

c) Infrastructure Layer

Infrastruktur layer merupakan lapisan terbawah dari arsitektur *Software Defined Network*. Didalam lapisan tersebut berisi perangkat keras yang berfungsi sebagai *Forwarding Plane* berdasarkan *Data plane*. Perangkat keras pada lapisan tersebut yaitu *router* dan *switch*.

2.3 Openflow

OpenFlow adalah standar pertama yang digunakan sebagai antarmuka komunikasi antara *controlplane* dengan *dataplane*. OpenFlow adalah protokol SDN yang paling banyak digunakan. Hal ini dikarenakan OpenFlow memungkinkan peneliti untuk mengembangkan bahkan merancang protokol yang baru.

OpenFlow mendefinisikan infrastruktur *flow-based forwarding* dan *Application Programmatic interface* (API) standar yang memungkinkan *controller* untuk mengarahkan fungsi dari *switch* melalui saluran yang aman (*secure channel*)[8]. *Controller* yang telah diprogram sesuai dengan konfigurasi jaringan yang diinginkan dapat terhubung dengan *forward plane*. Dimana *scure channel* ini merupakan perantara bagi *controller* untuk mengontrol jaringan pada *forward plane*.

2.4 Protokol Routing BGP (Border Gateway Protocol)

Border Gateway Protocol atau yang sering disingkat BGP merupakan salah satu jenis routing protokol yang ada di dunia komunikasi data. Sebagai sebuah routing protokol, BGP memiliki kemampuan melakukan pengumpulan rute, pertukaran rute dan menentukan rute terbaik menuju ke sebuah lokasi dalam jaringan. Routing protokol juga pasti dilengkapi dengan algoritma yang pintar dalam mencari jalan terbaik. Namun yang membedakan BGP dengan routing protokol lain seperti misalnya OSPF dan IS – IS adalah BGP termasuk dalam kategori routing protokol jenis *Exterior Gateway Protocol* (EGP). Sesuai dengan namanya, EGP memiliki kemampuan pertukaran rute dari dan keluar jaringan lokal sebuah organisasi atau kelompok tertentu. Organisasi atau kelompok tertentu diluar organisasi pribadi sering disebut dengan *Autonomous System* (AS). Maksudnya rute – rute yang dimiliki oleh sebuah AS dapat dimiliki oleh AS lainnya yang berbeda kepentingan dan otoritas. Begitu juga dengan AS tersebut dapat juga memiliki rute – rute yang dipunya organisasi lain dimana keuntungannya adalah organisasi anda bisa dikenal dengan organisasi – organisasi lain yang anda kirim rute [20].

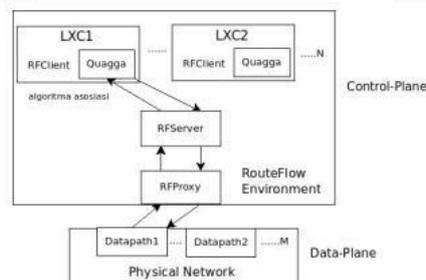
BGP dikenal sebagai routing protokol yang sangat kompleks dan rumit karena kemampuannya yang luar biasa ini, yaitu melayani pertukaran rute antar oraganisasi yang besar. Routing protokol ini memiliki tingkat skalabilitas yang tinggi karena beberapa organisasi dapat dilayaninya dalam melakukan pertukaran routing, sehingga luas sekali jangkauan BGP dalam melayani para pengguna jaringan.

2.5 RouteFlow

RouteFlow terbentuk atas penggabungan proyek *OpenFlow* dengan *routing engine* Quagga. Sistem ini terdiri dari *controller OpenFlow* (RFProxy), RFClient dan *Independent Server* (RFServer). Tujuan utama dibuatnya *RouteFlow* adalah menerapkan virtualisasi IP *routing* secara terpusat, dengan memisahkan fungsi *control plane* dan *data-plane*. Penelitian ini memanfaatkan *RouteFlow* sebagai sistem yang berjalan pada *control plan*[8].

Arsitektur *RouteFlow* terdiri dari :

1. RFServer adalah *standalone application* yang memegang kendali pusat kontrol jaringan. RFServer mengatur *Virtual Machine* (VMware) yang berjalan pada RFClient dan mengatur *logic process* (seperti *event processing*, VM mapping, *resource management*).
2. RFProxy adalah *controller* POX yang bertugas meneruskan kebijakan protokol (misalnya *update route*, konfigurasi *datapath*) dari RFServer ke *data-plane*.
3. RFClient adalah *daemon* pada VMware yang bertugas mendeteksi perubahan informasi *routing* dan memberitahunya ke RFServer. Tugas tersebut dikomunikasikan dengan *routing engine* (Quagga).



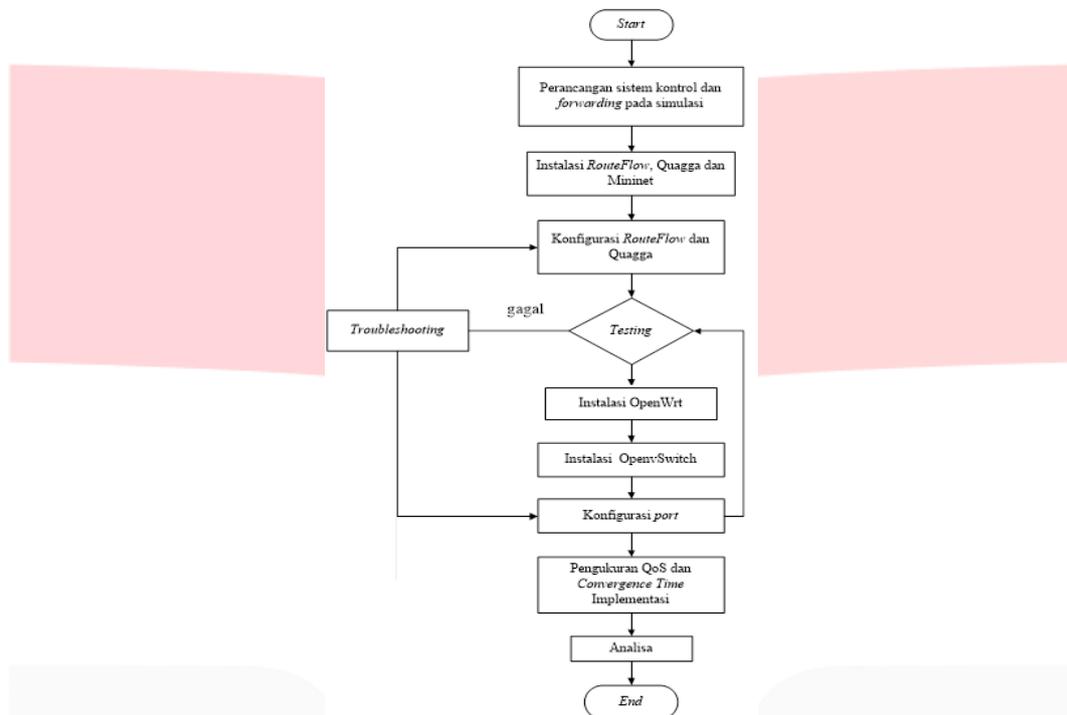
Gambar 2.2 Arsitektur Sistem RouteFlow

2.6 Quagga

Quagga adalah suatu *routing engine* yang dapat menjalankan protokol *routing* konvensional seperti RIP, OSPF, BGP, dan lain-lain. Quagga yang tujuannya secara umum implementasi *routing* bersifat *open source* yang cocok untuk digunakan di SDN *environment*. Penerapan quagga dapat digunakan untuk Unix *platform* khususnya FreeBSD, Linux, Solaris, dan NetBSD. 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 *client*. Pada Zserv *client* ini biasanya menerapkan protokol *routing* dan menyampaikan *update routing* ke zebra daemon[8].

3. Perancangan Sistem

3.1 Metode Penelitian

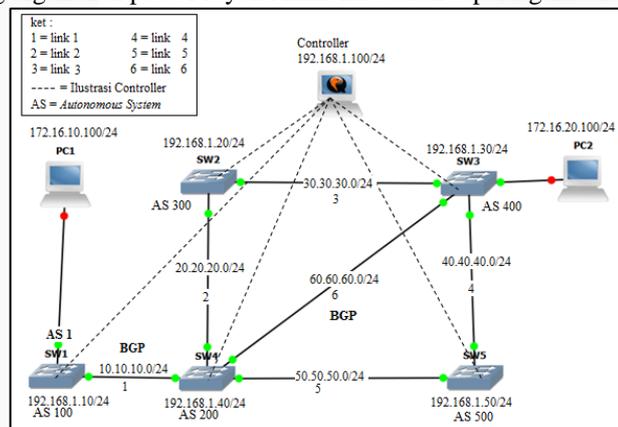


Gambar 3.1 Flowchart Metode Penelitian

3.2 Desain Topologi Jaringan

Desain topologi jaringan pada Proyek Akhir ini menggunakan Laptop sebagai kontroler, 5 buah TP-Link sebagai switch arus data forwarding, dan 2 buah Laptop sebagai host.

Topologi jaringan yang digunakan pada Proyek Akhir ini adalah seperti gambar dibawah ini:



Gambar 3.2 Desain Topologi Jaringan

3.3 Spesifikasi Perangkat Implementasi

Dalam melakukan proyek akhir ini digunakan beberapa perangkat keras dan perangkat lunak yang menunjang proses implementasi. Perangkat implementasi yang digunakan terdiri dari wireless router TP-Link WR1043ND v2 dan 1 buah laptop yang didalamnya terdapat perangkat lunak berupa VMware sebagai controller pada RouteFlow environment dan TP-Link WR1043ND v2 sebagai Physical Infrastructure.

3.4 Perancangan Sistem Kontrol

Perancangan sistem kontrol merupakan tahapan yang dilakukan untuk merancang dan membuat kontrol jaringan berbasis SDN. Pada Proyek Akhir ini, perancangan sistem kontrol dibagi menjadi tiga bagian dan akan dijelaskan lebih rinci.

3.4.1 Instalasi *RouteFlow*

Tahapan ini merupakan instalasi *RouteFlow* pada VMware dengan sistem operasi Ubuntu 12.04 LTS yang memiliki kemampuan fungsi kontrol *routing*.

Proses instalasi *RouteFlow* adalah melakukan instalasi *POX controller*, Quagga, dan komponen-komponen pendukung *RouteFlow* itu sendiri.

3.4.2 Konfigurasi *RouteFlow*

Tahap ini merupakan konfigurasi pada *RouteFlow* sesuai dengan topologi jaringan yang dirancang sebelumnya. Sehingga *switch* dapat menerima *message* dari *controller* (RFProxy). *Message* tersebut akan diteruskan ke RFServer dalam bentuk *mapping message* menuju Quagga. Hal ini bertujuan agar *port* pada *virtual switch* sesuai dengan *port data plane*.

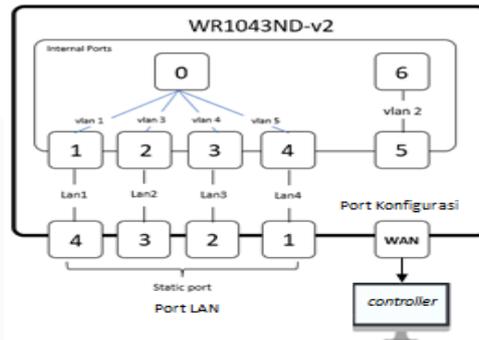
3.4.3 Konfigurasi Quagga

Setelah melakukan konfigurasi *RouteFlow* maka pada tahapan ini akan dilakukan konfigurasi Quagga, meliputi mengaktifkan *routing* BGP, dan mengatur *IP address* sesuai dengan topologi yang dirancang sebelumnya. Dalam melakukan konfigurasi Quagga dilakukan pada *file* *zebra* dan *daemon*. Karena pada Proyek Akhir ini menggunakan *routing* BGP, maka penjaluran dikonfigurasi pada *file* *bgpd.conf*.

3.5 Perancangan *Forwarding*

Perancangan *forwarding* merupakan tahapan yang dilakukan untuk merancang dan membuat sistem perutean pada jaringan berbasis SDN. Pada Proyek Akhir ini, perancangan dibagi menjadi dua bagian yaitu proses instalasi OpenWrt dan OpenvSwitch serta konfigurasi *port*.

Berikut ini *schematic* dari konfigurasi *port* pada TP-Link WR1043ND v2 menggunakan OpenWrt 15.05 Chaos Calmer:



Gambar 3.3 Konfigurasi *Port*

4. Hasil dan Analisa

4.1 Pengujian Pada Implementasi

Pengujian pada implementasi Proyek Akhir ini merupakan pengujian yang dilakukan untuk mengetahui performansi implementasi jaringan yang dirancang.

4.1.1 Pengujian *Switch Forwarding*

Pengujian *switch* pada Proyek Akhir ini dilakukan untuk mengetahui bahwa perangkat *data plane* dan *Controller* sudah saling terintegrasi. Pengujian ini dilakukan pada *switch* TP-Link WR1043ND v2 menggunakan OpenWrt 15.05 Chaos Calmer dalam keadaan *default*

```

CHAOS CALMER (Chaos Calmer, r49389)
-----
* 1 1/2 oz Gin          Shake with a glassful
* 1/4 oz Triple Sec     of broken ice and pour
* 3/4 oz Lime Juice     unstrained into a goblet.
* 1 1/2 oz Orange Juice
* 1 tsp. Grenadine Syrup
-----

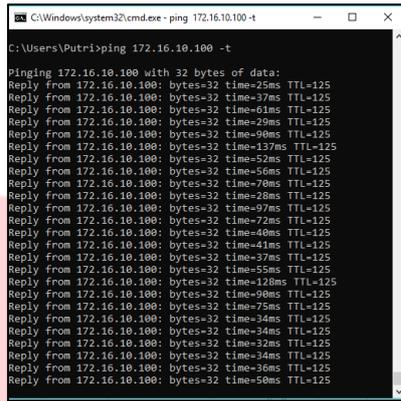
root@OpenWrt:~# ovs-vsctl show
7d326097-02f0-47d3-bbd9-5e29cb6e4eb
-----
Bridge "s1"
  Controller "tcp:192.168.1.100:6633"
    is connected: true
  Port "s1"
    Interface "s1"
      type: internal
  Port "eth1.5"
    Interface "eth1.5"
  Port "eth1.3"
    Interface "eth1.3"
  Port "eth1.4"
    Interface "eth1.4"
  Port "eth1.1"
    Interface "eth1.1"

```

Gambar 4.1 *Switch* dan *Controller* sudah terintegrasi

4.1.2 Pengujian *Fungsionalitas*

Untuk membuktikan *routing* bekerja dengan baik dan benar, maka dilakukan pengecekan koneksi dengan melakukan ping *host-to-host* antara H2 dan H3 seperti gambar dibawah ini:



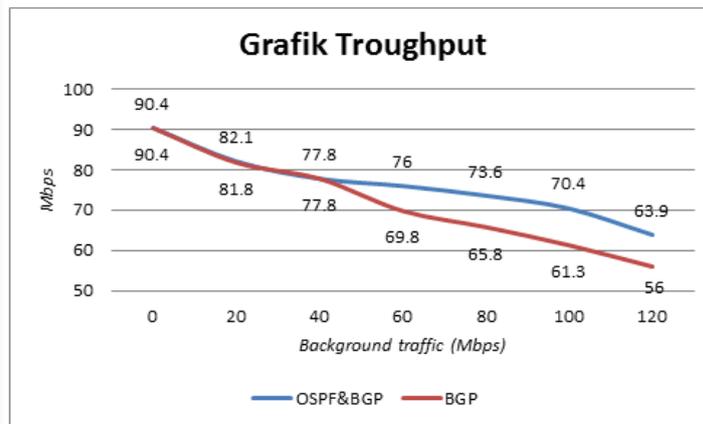
Gambar 4.2 Hasil Ping h2 ke h3

4.2 Pengukuran Performansi

Pengukuran performansi jaringan pada Proyek Akhir ini yang menerapkan protokol *routing* BGP dan protokol *routing* OSPF&BGP menggunakan *tool* iperf. Pengujian ini dilakukan pada h2 ke h3 yang sudah saling terhubung dan dapat berkomunikasi menggunakan protokol *routing* BGP dan *routing* OSPF&BGP dan menggunakan *measurement software* berupa *iperf* dengan metode *server-side* (h3) pada pengirim dan *client-side* (h2) pada penerima. Pengukuran dilakukan pada implementasi protokol *routing* BGP dengan waktu 60 detik dan pada protokol *routing* OSPF&BGP dengan waktu 60 detik. pengukuran ini dilakukan dengan *setting background traffic* yang awal nya 0 atau tidak ada sampai dengan *setting background traffic* 120Mbps. Parameter-parameter yang diukur adalah *Throughput*, *Delay*, *Jitter*, *Packet loss*, dan *Convergence time*. Berikut penjelasannya:

a. *Troughput*

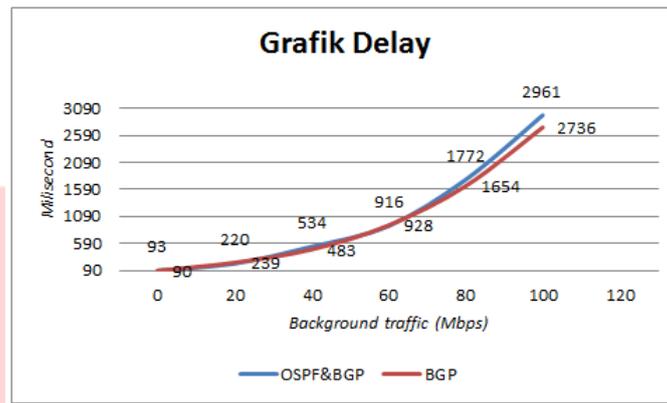
Pengukuran *throughput* ini bertujuan untuk mengetahui kemampuan jaringan dalam mentransmisikan data secara aktual. pengukuran ini dilakukan dengan *setting background traffic* yang awal nya 0 atau tidak ada sampai dengan *setting background traffic* 120Mbps.



Gambar 4.3 Grafik Troughput dengan background traffic

b. *Delay*

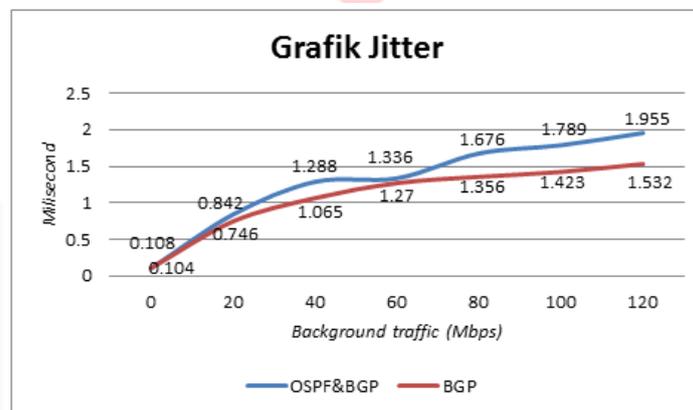
Pengukuran *delay* pada Proyek Akhir ini dilakukan menggunakan *ping* dalam waktu 60 detik, dengan dilakukan *setting background traffic* yang awal nya 0 atau tidak ada sampai dengan *setting background traffic* 120Mbps. Pada *background traffic* 120Mbps jaringan terjadi RTO (*Request Time Out*) sehingga koneksi terputus.



Gambar 4.4 Grafik Delay dengan background traffic

c. Jitter

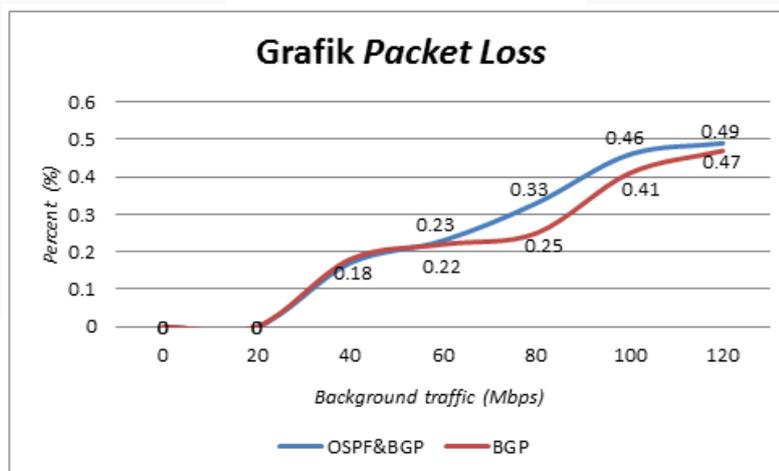
Pengukuran jitter pada Proyek Akhir ini dilakukan dalam 60 detik, dengan dilakukan setting background traffic yang awalnya 0 atau tidak ada sampai dengan setting background traffic 120Mbps. Dapat dilihat semakin besar background traffic maka nilai jitter akan semakin besar.



Gambar 4.5 Grafik Jitter dengan background traffic

d. Packet Loss

Pengukuran packet loss pada Proyek Akhir ini dilakukan dalam jangka waktu 60 detik dengan dilakukan setting background traffic yang awalnya 0 atau tidak ada sampai dengan setting background traffic 120Mbps. Pada background traffic 0 – 20 Mbps tidak terdapat packet loss sedangkan pada background traffic 40 – 120 terdapat packet loss.

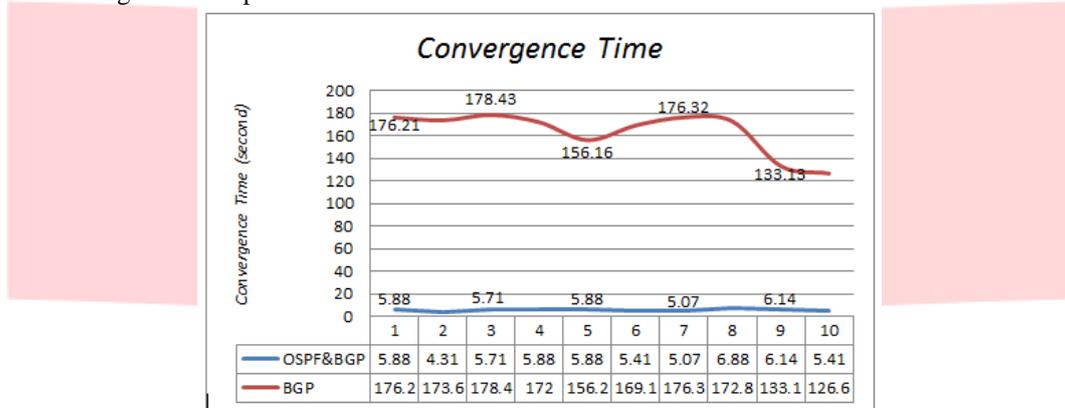


Gambar 4.6 Grafik packet loss dengan background traffic

a. *Convergence time*

Convergence time adalah seberapa cepat waktu yang dibutuhkan oleh *router* untuk mencapai keadaan konvergensi. Dalam setiap protokol *routing convergence time* sangat penting. Semakin cepat *convergence time* maka semakin baik.

Berikut ini merupakan grafik *convergence time* jaringan SDN implementasi protokol *routing* BGP dan OSPF&BGP dengan 10 kali percobaan.



Gambar 4.7 Grafik *convergence time*

Dari grafik diatas dapat dilihat bahwa *convergence time* pada protokol *routing* BGP membutuhkan waktu yang sangat lama dibandingkan dengan OSPF&BGP untuk mencapai keadaan konvergensi. Hasil tersebut dipengaruhi salah satunya oleh fitur BGP, yakni *keepalive* dan *holdtime* yang memiliki nilai *default*, yaitu 60 detik (*keepalive*) dan 180 detik (*holdtime*). Maka nilai *keepalive* adalah sepertiga dari *holdtime*.

Dalam mendeteksi adanya *link-failure*, BGP secara berkala mengirimkan *Keepalive message* antar *switch*. Jika sebuah *switch* tidak menerima *Keepalive message* tiga kali berturut-turut (*holdtime*), maka sebuah *link* dianggap *down*. Sehingga jaringan perlu waktu untuk melakukan konvergensi menggunakan protokol *routing* yang digunakan.

5. Kesimpulan

Berdasarkan hasil dari Proyek Akhir yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut:

1. Implementasi dan simulasi jaringan SDN yang memisahkan fungsi kontrol dengan fungsi *forwarding* dapat dilakukan dengan menggunakan *RouteFlow*.
2. Hasil pengujian performansi penerapan *routing* BGP berbasis *RouteFlow* pada *Software Defined Network* menunjukkan bahawa nilai dari keempat parameter QoS (*throughput*, *jitter*, *delay* dan *packet loss*) masih berada pada nilai yang menjadi standar ITU-T G.1010.
3. Implementasi jaringan *software defined network* berbasis *RouteFlow* telah selesai dibuat dan di uji performansinya menggunakan parameter *convergence time*, *throughput*, *delay*, *jitter* dan *packet loss* dengan hasil sebagai berikut:
 - a. Rata-rata *convergence time* implementasi dari hasil Proyek Akhir ini adalah 163.441s.
 - b. Rata-rata *throughput* implementasi dari hasil Proyek Akhir ini adalah 90.4 Mbit/sec,
 - c. Rata-rata *delay* implementasi dari hasil Proyek Akhir ini adalah 90 ms.
 - d. Rata-rata *jitter* implementasi dari hasil Proyek Akhir ini adalah 0.108 ms.
 - e. Rata-rata *packet loss* implementasi dari hasil Proyek Akhir ini adalah 0 %

DAFTAR PUSTAKA

- [1] Adrian ronald, Anam khoerul. 2017. "analisa performa jaringan *software defined network* berdasarkan penggunaan *cost* pada protokol *routing open shortest path first*," Universitas Gajah Mada, yogyakarta.
- [2] Cisco, "sample configuration for BGP with two different service providers (multihoming)", 17 Agustus 2005. [Online]. Diakses pada 21 mei 2018 Available: <https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/23675-27.html>
- [3] F. Adnantlya, Hertiana Sofia Naning.ST.,MT, Yovita Leanna Vidya.ST.,MT, 2015. "Simulasi dan Analisis Kinerja Protokol Ruting eBGP pada SDN (Software Defined Network)," Universitas Telkom, Bandung.
- [4] freeBSD, "bgpd.conf", july 2000. [Online]. Diakses pada 21 mei 2018 Available: <https://www.freebsd.org/cgi/man.cgi?query=bgpd&sektion=8&apropos=0&manpath=FreeBSD+11.1-RELEASE+and+Ports>
- [5] Github, "bgp-workshop/config-examples/quagga/bgpd.conf", 18 mai 2017. [Online]. Diakses pada 22 mei 2018 Available: <https://github.com/hackman/bgp-workshop/blob/master/config-examples/quagga/bgpd.conf>

- [6] Hidayah Irvan, Dyah Indrarini.ST.,MT, Harlyani Yuli Sun.ST.,MT, 2016. "Implementasi Protokol Routing OSPF Pada Software Defined Network Berbasis Routedflow," Universitas Telkom, Bandung.
- [7] Hikam Hidayat Muhammad, Rahman Nur, 2017. "Analisis Kinerja dan Karakteristik Arsitektur Software Define Network Berbasis Opendaylight Controller", Universitas Gajah Mada, Yogyakarta.
- [8] Irmawati Ayu, Dyah Indrarini.ST.,MT, Harlyani Yuli Sun.ST.,MT, 2016. "Implementasi Protokol Routing OSPF Pada Software Defined Network Berbasis Routedflow," Universitas Telkom, Bandung.
- [9] ITU-T. (2011). ITU-T G.1010 : *End-user multimedia QoS categories*. ITU-T Recommendation G.1010.
- [10] Kartadie rieke, Suryanto Tommi, 2015. "Uji performa software – based openflow Switch Berbasis Openwrt", STMIK Amikom, Yogyakarta.
- [11] Kaur Sukhveer, Singh Japinder, Singh Navtej, 2014. "*network Programmability Using POX Controller*. SBS State Technical Campus, India.
- [12] Lartc.org, "*setting BGP with zebra*", 5 April 2012. [Online]. Diakses pada 21 mei 2018 Available: <http://lartc.org/howto/lartc.dynamic-routing.bgp.html#LARTC.DYNAMIC-ROUTING.BGP.CONFIG>
- [13] Muhammad Aldilla Fauzi, 2016, "Software Defined Network Pemisahan Control Plane Dan Data Plane Sebagai Transformasi Network Menjadi Lebih Terpusat," Universitas Gunadarma.
- [14] Nur Lutfi Niswati , 2017. "Routing dengan setting EIGRP dan OSPF (Menggukanan Bandwidth dan cost)", Politeknik Negeri Semarang, Semarang.
- [15] "OpenvSwitch". 21 April 2014. [Online]. Diakses pada 22 Mei 2018 Available: <http://roan.logdown.com/posts/191801-set-openvswitch>
- [16] Pujo Aries. 2012. "pembangunan interface router sebagai implementasi routing dinamik Ipv6 dengan quagga", Universitas Sebelas Maret, Surakarta
- [17] Tulloh. Rohmat.,ST.,MT, Negara M. Ridha.ST.,MT, Hidayat. Arif. 2015. Simulasi Virtual Local Area Network (VLAN) Berbasis Software Defined Network (SDN) Menggunakan POX Controller, Jurnal Infotel.
- [18] Tulloh. Rohmat.,ST.,MT, Negara M. Ridha.ST.,MT. 2017. Analisis Simulasi Penerapan Algoritma OSPF Menggunakan *RouteFlow* pada Jaringan *Software Defined Network* (SDN), Jurnal Infotel.
- [19] "TP-Link WR-1043ND v2 *datasheet*". [Online]. Diakses pada 10 Mei 2017. Available: http://www.tp-link.com/en/download/TL-WR1043ND_V2.html.
- [20] Winda Veronica, lucas, 2017. "*Border Gateway Protocol* Pada Teknologi *Software Defined Network*", Universitas Katolik Indonesia, Jakarta.
- [21] Yanto, 2013, " analisis qos (*quality of service*) pada jaringan internet (studi kasus: fakultas teknik universitas tanjungpura)," Universitas Tanjungpura, Pontianak.
- [22] Youtube, "kelas SDN Telkom University". [Online]. Diakses pada 15 Januari 2018. Available: https://www.youtube.com/channel/UCQ0DTmYS-IxEL_4fw2WbXyQ