ANALISIS PERFORMANSI AGREGASI *LINK* DENGAN LACP PADA SDN MENGGUNAKAN RYU SEBAGAI CONTROLLER

Rohmat Tulloh
Program Studi D3 Teknik Telekomunikasi, Fakultas Ilmu Terapan, Universitas Telkom
Corresponding author, e-mail: rohmatth@telkomuniversity.ac.id

Abstrak—Dunia telekomunikasi saat ini berkembang dengan sangat pesat terutama dalam dunia jaringan. Salah satu teknologi jaringan masa kini adalah SDN (*Software Defined Network*) yaitu sebuah paradigma baru dalam teknologi jaringan yang dapat memvirtualisasi perangkat jaringan dengan mengambil fungsi *control plane* oleh *controller* dan memisahkan antara *control plane* dengan *data plane* dalam sebuah perangkat jaringan. Selain adanya paradigma baru tersebut, kebutuhan akan konsumsi *bandwidth* dalam komunikasi data saat ini juga meningkat. *Link aggregation* merupakan metode penggabungan *link* fisik menjadi satu *link* logis guna meningkatkan kapasitas *bandwidth*. *Link aggregation* dapat di implementasikan dalam jaringan SDN, hal ini membuat konsumsi sejumlah *bandwidth* pada jaringan berdampak baik terhadap kualitas layanan pada jaringan. Dalam penelitian ini, ditunjukkan analisis perbandingan terhadap jaringan SDN yang menggunakan metode *link aggregation* dengan jaringan SDN yang tanpa metode *link aggregation*. Analisis ini memperhatikan hasil pengukuran *throughput, packet loss, jitter* dan *fault tolerant* dari layanan video streaming pada jaringan.

Kata Kunci : SDN, Link aggregation, LACP, Redundansi, Bandwith.

Abstract— As time goes by, the world of communication has flourished quickly, especially in network. One of the network technologies by now called SDN (Software Defined Network), is a new paradigm in network technology that is able to virtualize network device by taking the function of control plane, that is usually taken by controller, and separate between control plane and data plane into a network device. Besides the coming of that new paradigm, the need of *bandwidth* consumption on data communication by this time has increased. *Link* aggregation is a method that combines several physical *link*s into a logical *link* in the form of multiplying the *bandwidth* capacity. *Link* aggregation can be implemented in SDN network, that makes *bandwidth* consumption in network has good effect on service quality. In this research, the writer will show the comparison analysis between *link* aggregation and not using *link* aggregation in SDN network. This comparison analysis is done by analyzing some parameters, like throughput, packetloss, jitter and fault tolerance from the service in network, called video streaming.

Keywords : SDN, Link aggregation, LACP, Redundansi, Bandwith.

1. PENDAHULUAN

Jaringan komputer dan data terdiri atas tiga macam *plane* yang masing-masing memiliki fungsi khusus, yakni *data plane* yang berfungsi untuk meneruskan paket data (*forwarding*), *control plane* yang berfungsi untuk mengatur *forwarding table* yang menjadi acuan *data plane* untuk meneruskan paket data, dan *management plane* yang menyediakan layanan-layanan berupa *software*.

Jaringan komputer konvensional memiliki *control plane* dan *data plane* yang terikat dalam sebuah perangkat. Hal ini merupakan konsep yang disengaja pada masa-masa awal Internet berkembang untuk menjamin ketahanan (*resilience*) jaringan. Namun dengan berjalannya waktu dan meningkatnya kompleksitas kebutuhan pengguna akan layanan telekomunikasi data, konsep ini dirasa menjadi faktor besar dalam menghambat inovasi layananan, fitur perangkat dan memberatkan dari sisi efisiensi dan efektifitas serta menghasilkan arsitektur jaringan yang semakin kompleks.

Software-Defined Network (SDN) merupakan suatu paradigma baru dalam jaringan telekomunikasi data yang memisahkan *control plane* dari perangkat jaringan ke dalam suatu perangkat eksternal yang disebut *controller*, sehingga perangkat *switch* yang hanya semata-mata berfungsi sebagai *forwarding devices*.

Received date 2017-08-13, Revised date 2017-09-25, Accepted date 2017-10-12
DOI : 10.20449/jnte.v6i3.444

Dalam implementasinya, dibutuhkan suatu API (*Application Programming Interface)* yang menghubungkan antara *switch* dan *SDN Controller*, contohnya adalah *OpenFlow* [1]. Dalam suatu *switch OpenFlow* terdapat banyak *rules table* yang mengatur bagaimana suatu paket yang masuk ke dalam *switch* diolah. Kebanyakan vendor perangkat jaringan telekomunikasi sudah memasukkan *OpenFlow* API ke dalam perangkat baru mereka.

Selain itu, semakin kompleksnya kebutuhan masyarakat saat ini menuntut suatu jaringan telekomunikasi bersifat *high availability*, biaya relatif murah, kecepatan tinggi, dan memiliki kapasitas yang besar diluar keterbatasan perangkat. Untuk mendukung hal tersebut maka perlu juga ada penambahan ketersediaan *bandwidth*. Diharapkan dengan penambahan *bandwidth*, maka layanan yang disediakan akan bertambah banyak. Untuk menjaga berbagai layanan tersebut tidak cukup hanya menggunakan satu *gateway*, jika *gateway* tersebut "*down*" maka koneksi terputus sehingga kegiatan yang memerlukan jaringan internet menjadi terhambat.

Link aggregation adalah metode penggabungan *link* fisik menjadi satu *link* logis untuk meningkatkan kapasitas *bandwidth*. *Link aggregation* mampu meningkatkan kapasitas dan ketersediaan saluran komunikasi antara perangkat yang menggunakan teknologi *Fast Ethernet* dan *Gigabit Ethernet*. *Link aggregation* juga menyediakan di mana pengolahan dan komunikasi aktivitas didistribusikan di beberapa *link* sehingga tidak ada *link* tunggal yang kewalahan menangani trafik. Fungsi lain dari *link aggregation* adalah menyediakan redundansi ketika salah satu *link* fisik putus, dimana fitur redundansi terbatas untuk perangkat jaringan standar [2]. *Link aggregation* diatur oleh standar IEEE 802.3ad [3].

Fungsi lain dari *link aggregation* adalah dapat menaikkan kecepatan koneksi antara *switch* dengan *router*, kemudian koneksi sesama switch serta koneksi server dengan router. Apabila terjadi kerusakan pada salah satu *port* maka akan menggunakan *port* yang lain dalam *port group* yang *sama* untuk menjaga sistem tetap bekerja [4]. Sehingga ditawarkan solusi untuk mengoptimasikan redundansi dan jaringan yang mampu melakukan *load balancing* sehingga paket yang dikirim lebih efektif dengan menggunakan LACP (*Link aggregation Control Protocol*) [5].

Penelitian ini bertujuan untuk mengukur dan menganalisis performansi jaringan dengan dan tanpa *link aggregation* dilakukan.

2. PENELITIAN TERKAIT DAN KONTRIBUSI

Pada [6], penulis melakukan implementasi *link aggregation* di jaringan VLAN yang diintegrasikan dengan algoritma *round-robin*. Penelitian ini menggunakan jaringan konvensional sehingga membutuhkan perangkat keras lain seperti router *Mikrotik*.

Pada [7], penulis meneliti mengenai kelebihan *link aggregation* beserta fungsi dan tipe dari *link aggregation* serta menjelaskan mengenai konfigurasi LACP pada *SysConnect* yang memungkinkan *load balancing* dengan informasi data berbasis IP, TCP dan UDP. Penulis masih mengimplementasikan LACP pada jaringan konvensional.

Pada [8], penulis menawarkan pengimplementasian LACP di jaringan SDN dengan topologi yang digunakan adalah spanning tree dan Floodlight sebagai controllernya. Paper ini hanya memberikan ide/gagasan, tidak terdapat hasil simulasi LACP yang ditawarkan.

Pada penelitian ini, penulis melakukan analisis performansi dari penggunaan protokol LACP pada jaringan SDN untuk layanan *video streaming* berbasis UDP dengan protokol MPEG. Parameter performansi yang dianalisis adalah *throughput, jitter dan packetloss*.

3. TINJAUAN PUSTAKA

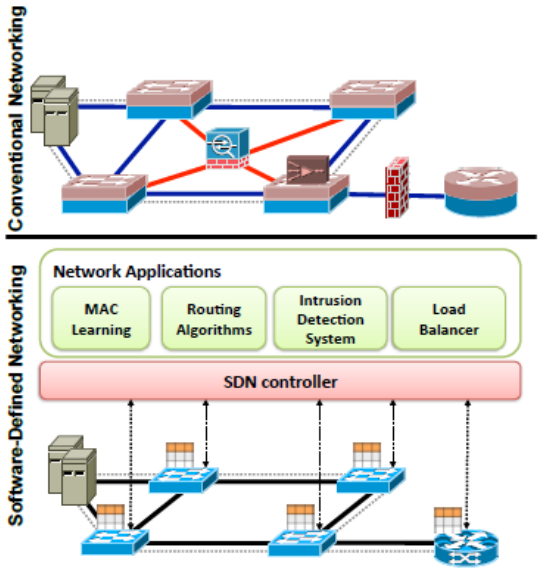
3.1. *Software Defined Network (SDN)*

Jaringan IP tradisional yang memiliki konsep pengikatan *control plane* dan *data plane* dalam satu perangkat jaringan terbukti menghambat inovasi fitur layanan dalam jaringan, menghasilkan arsitektur jaringan yang semakin kompleks, untuk diatur dan bersifat statis [9], [10]. Selain itu kesalahan konfigurasi perangkat jaringan sudah merupakan hal yang umum saat ini, sebagai contoh terdapat lebih dari 1000 kesalahan konfigurasi yang berhasil diselidiki pada perangkat router BGP [11]. Kesalahan konfigurasi dalam suatu perangkat

jaringan bisa mempengaruhi pola *forwarding*, *packet loss* atau bahkan membuat suatu layanan menjadi tidak berfungsi. Menurut statistik yang ada, kesalahan sebuah perangkat jaringan bisa mengganggu operasi seluruh jaringan Internet selama berjam-jam [12].

Setiap vendor perangkat jaringan menyediakan solusi atas sebuah fitur layanan jaringan yang memiliki paten, yang bisa digunakan dan diaplikasikan untuk perangkat, sistem operasi dan aplikasi jaringan yang merupakan produksi vendor tersebut. Akibatnya, operator jaringan yang menggunakan berbagai macam perangkat dari vendor berbeda perlu mempelajari solusi-solusi yang disediakan vendor-vendor tersebut dan bahkan membentuk suatu tim khusus untuk menangani perangkat dari masing-masing vendor. Tentu saja, biaya oeprasional untuk *maintenance* menjadi besar bagi operator jaringan.

Software-Defined Network (SDN) adalah suatu paradigma baru dalam jaringan, dimana data plane dikontrol oleh *control plane* yang bersifat *remote* dan tidak lagi terikat dalam perangkat jaringan. Gambar 1 menunjukkan arsitektur dari jaringan SDN dan perbedaannya dengan jaringan IP tradisional.



Gambar 1. Arsitektur SDN dibandingkan dengan jaringan tradisional [13]

Terdapat empat buah pilar utama yang membedakan jaringan SDN dengan jaringan IP tradisional [13], yaitu:

1. Pemisahan antara *control plane* dengan *data plane*. Fungsi kontrol dijalankan oleh perangkat eksternal sehingga perangkat jaringan hanya berfungsi untuk *forwarding* semata.
2. Keputusan *forwarding* data bersifat *flow-based*. *Flow* didefinisikan oleh kumpulan nilai *packet field* yang berfungsi sebagai *filter* dan sekumpulan aksi atau instruksi.
3. Logika kontrol berada pada perangkat eksternal yang disebut *Network Operating System* (NOS) atau yang lebih dikenal dengan SDN *controller*.
4. Jaringan dapat diprogram melalui aplikasi yang berjalan diatas NOS yang berinteraksi dengan perangkat pada level *data plane*.

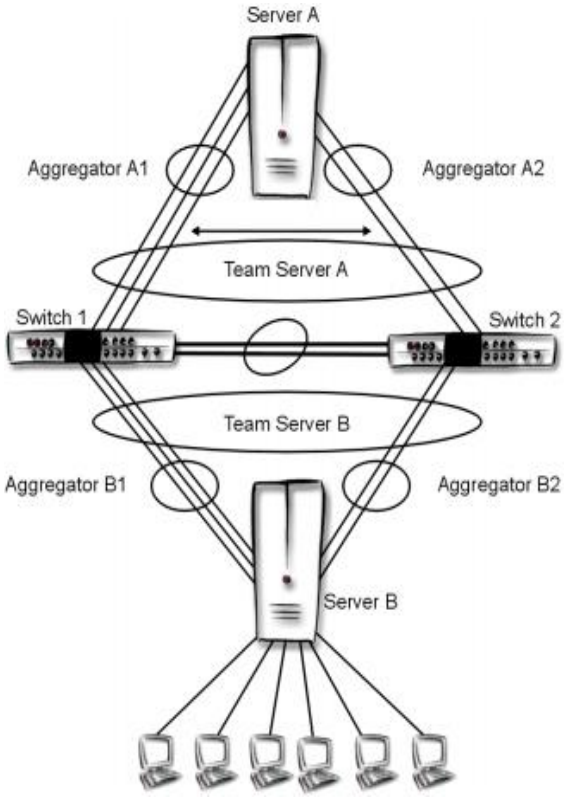
Konsep SDN yang dinilai berprospek dan menjanjikan masa depan jaringan komputer dan data yang lebih baik berhasil membuat perusahan-perusahan besar yang bergerak di bidang ICT seperti Google, Facebook, Yahoo, Microsoft, Verizon, dan Deutsche Telekom turut berpartisipasi dalam riset yang diwadahi oleh *Open Networking Foundation* (ONF) [14]. Terdapat beberapa kelebihan akibat penggunaan SDN dalam jaringan, yaitu:

1. Memudahkan dalam melakukan pemrograman jaringan karena abstraksi yang disediakan oleh *control platform* dan bahasa pemrograman dapat dibagikan satu sama lain.
2. Seluruh aplikasi dapat mengambil informasi dalam jaringan sehingga keputusan forwarding dapat dilakukan secara konsisten dan efektif.
3. Re-konfigurasi perangkat jaringan dapat dilakukan dari bagian manapun dalam jaringan sehingga tidak perlu strategi khusus untuk menentukan lokasi fungsi baru.
4. Integrasi berbagai macam aplikasi dan fitur dalam jaringan dapat dilakukan dengan mudah, semisal menggabungkan *load balancing* dengan *routing protocol.*

3.2. *Link aggregation*

Link aggregation adalah suatu metode yang mengkombinasikan beberapa *link* fisik menjadi suatu *link* logika tunggal. Melalui *link aggregation*, kapasitas dan *availability* kanal antar perangkat meningkat dengan menggunakan teknologi *Fast Ethernet* dan *Gigabit Ethernet* saat ini. *Link aggregation* juga menyediakan fitur *load balancing*, dimana

pemrosesan dan komunikasi data dilakukan pada beberapa *link* sehingga tidak terjadi suatu *link* yang kewalahan menangani trafik [7]. Standar dari *link aggregation* adalah IEEE 802.3ad. Gambar 2 menunjukkan contoh dari *link aggregation* dalam suatu arsitektur jaringan.



Gambar 2. *Link aggregation* dalam suatu arsitektur jaringan [7]

Untuk meningkatkan kapasitas *link*, ada dua cara yang dapat dilakukan yakni melakukan *upgrade* suatu *link* ke kapasitas yang lebih besar atau dengan melakukan agregasi dua atau lebih *link* yang memiliki kecepatan yang lebih rendah. Dengan menggunakan *link aggregation*, biaya yang diperlukan lebih rendah dibanding melakukan *upgrade* kecepatan *link*, dimana level performansi yang dihasilkan sama. Bahkan, *link aggregation* merupakan satu-satunya solusi untuk meningkatkan performa *link* ketika *data rate* tertinggi yang tersedia di pasaran tidak mencukupi [7].

Beberapa kelebihan yang ditawarkan dengan menggunakan *link aggregation* adalah sebagai berikut:

1. *Link availability* yang tinggi, komunikasi data akan tetap berlangsung meski salah satu *link* fisik putus.
2. Peningkatan kapasitas *link*.
3. Peningkatan performansi diperoleh dengan menggunakan perangkat keras saat itu juga (*existing*) sehingga tidak perlu melakukan *upgrade* perangkat yang membutuhkan biaya besar.

3.3. LACP *(Link aggregation Control Protocol)*

Saat ini, terdapat berbagai macam metode untuk melakukan agregasi *link* dengan tujuan meningkatkan throughput dan redundansi suatu *link* komunikasi. Terdapat dua jenis metode agregasi *link*, yaitu statis dan dinamis. Metode statis memerlukan suatu konfigurasi secara langsung ke setiap perangkat jaringan, sedangkan metode dinamis menggunakan LACP untuk memulai fungsi agregasi *link* secara dinamis [3].

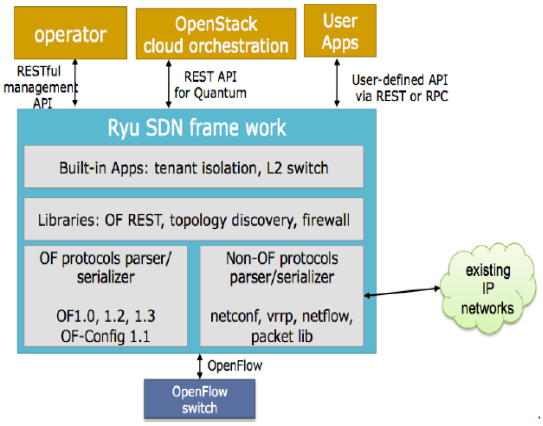
LACP yang memiliki standar IEEE 802.1ax memungkinkan perangkat jaringan untuk mengimplementasikan LACP untuk melakukan *bundling* dari *link* secara otomatis melalui perukaran paket LACP [2]. Dibandingkan dengan metode dinamis, LACP menawarkan 2 kelebihan, yaitu:

1. Konfigurasi otomatis, dimana end device di sisi lain mampu menangani *link aggregation*.
2. *Automatic failover*, penanganan otomatis ketika terjadi kegagalan dalam *link* komunikasi, sehingga sistem tidak mengalami permasalahan sambungan.

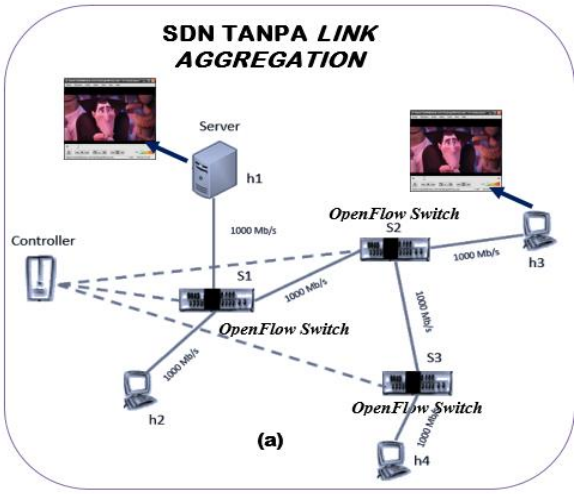
3.4. *Ryu Controller*

Ryu adalah *controller* dalam SDN yang didesain untuk meningkatkan kecerdasan jaringan dengan memudahkan untuk mengatur suatu trafik data [7]. Secara umum, *controller* merupakan otak atau inti dari jaringan berbasis SDN yang mengatur komunikasi dengan *switch* dan *router* dengan menggunakan *southbound APIs* dan komunikasi dengan aplikasi dan logika bisnis dengan menggunakan *northbound APIs.*

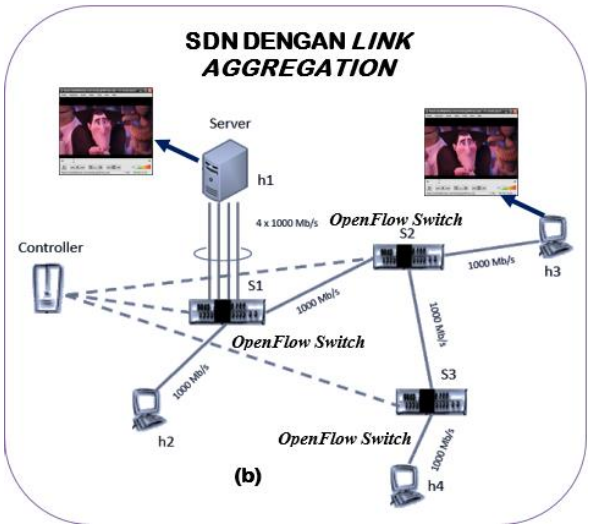
Ryu menyediakan komponen perangkat lunak dengan API yang memudahkan developer untuk membuat manajemen jaringan baru, sehingga memudahkan organisasi bisnis untuk mencapai spesifikasi jaringan yang diinginkan. Gambar 3 menunjukkan arsitektur *Ryu* sebagai *controller* dari SDN.



Gambar 3. Arsitektur dari *Ryu Controller* [8]



Gambar 4. Topologi jaringan SDN tanpa *link aggregation*



Gambar 5. Topologi jaringan SDN dengan *link aggregation*

4. METODE PENELITIAN

4.1. Perancangan Sistem

Penelitian ini menggunakan jaringan berbasis SDN, yang di-emulasi-kan menggunakan *Mininet* yang berjalan pada sistem operasi *Linux Ubuntu*. *Controller* yang digunakan pada penelitian ini adalah *Ryu*. Topologi jaringan yang digunakan adalah topologi jaringan *data center* sederhana, yang terdiri atas 3 buah *host* yang bertindak sebagai *client*, 1 buah *host* yang bertindak sebagai *server* layanan *real-time video streaming*, buah *switch* dengan *OpenFlow v1.3.0*. Seluruh *link* yang digunakan memiliki kapasitas 1000 Mb/s, kecuali pada *link* antara server video dengan *switch* yang memiliki kapasitas 100 Mb/s. Trafik *video streaming* memiliki teknik kompresi video H.264 yang dialirkan secara *real-time* ke *host* yang mengaksesnya.

Terdapat dua jenis topologi jaringan SDN yang digunakan, yakni pada Gambar 4 skenario jaringan SDN tanpa menggunakan *link aggregation* sehingga hanya ada *link* tunggal antara *server* layanan *video streaming* dengan *OpenFlow switch*, dan jaringan SDN Gambar 5 dengan menggunakan *link aggregation* yang terdiri atas 4 buah *link* fisik yang diikat menjadi satu buah *link* logika antara *server* layanan *video streaming* dengan *OpenFlow switch*, yang kemudian akan dibandingkan performansinya. Gambar 4 dan 5 menunjukkan topologi jaringan SDN tanpa *link aggregation* dan dengan *link aggregation* antara *server* dan *OpenFlow switch*.

4.2. Skenario Penelitian

Penelitian berupa perbandingan performansi antara jaringan SDN yang menerapkan konsep *link aggregation* dengan yang terhubung secara konvensional tanpa menggunakan *link aggregation*. Parameter performansi yang diukur diantaranya adalah *throughput, jitter* dan *packet loss*. Terdapat dua buah skenario, skenario pertama adalah variasi penambahan ukuran *background traffic*, yakni mulai tanpa adanya *background traffic* dalam jaringan, 200 Mb, 400 Mb, 600 Mb, hingga 800 Mb untuk dilihat performansinya terhadap nilai *throughput, jitter* dan *packetloss* dari *client* yang mengakses layanan *video streaming* secara *real-time*. Skenario kedua adalah pemutusan salah satu *link* fisik untuk dilihat pengaruh *link aggregation* terhadap redundansi jaringan terkait

network availability. Standar yang digunakan sebagai acuan untuk mengetahui baik atau tidaknya performansi yang dihasilkan adalah Rekomendasi ITU-T G.1080, dimana *throughput* minimal yang direkomendasikan untuk kompresi H.264 adalah 1.75 Mbps dan *jitter* minimalnya adalah 50 ms. Untuk *packet loss*, standar yang digunakan sebagai acuan adalah rekomendasi dari Cisco untuk QoS pada LAN, WAN, dan VPN [15], dimana nilai rekomendasi untuk layanan video streaming yang baik adalah diatas dibawah 5 persen.

5. HASIL DAN PEMBAHASAN

Bagian ini menampilkan data hasil emulasi menggunakan *Mininet* dan dijelaskan pengaruh ukuran *background traffic* terhadap beberapa parameter performansi yang diteliti *on* dan pemutusan salah satu *link* fisik sebagai bentuk uji redundansi dan *availability* dari jaringan.

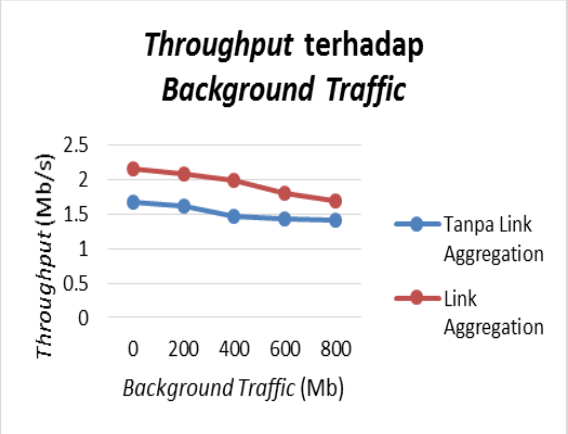
5.1. Variasi ukuran *background traffic* terhadap *throughput*

Pada Tabel 1 dan Gambar 6 ditunjukan hubungan antara ukuran *background traffic* terhadap *throughput* dari sisi *client* yang melakukan *video streaming* secara *real-time.*

Tabel 1. *Throughput* Terhadap Ukuran *Background Traffic*

Background Traffic (Mb)	Throughput (Mbit/sec)	
	Tanpa *Link aggregation*	Menggunakan *Link aggregation*
0	1.668	2.153
200	1.615	2.087
400	1.469	1.98
600	1.422	1.799
800	1.416	1.697

Dari Gambar 6 terlihat bahwa *throughput* mengalami penurunan seiring dengan bertambahnya ukuran *background traffic*. Kondisi tersebut karena kapasitas jalur yang digunakan semakin besar, yang membuat *throughput* untuk streaming layanan video mengecil. Terlihat, *throughput* mengalami penurunan hingga sebesar 45.43% di jaringan tanpa *link aggregation* serta penurunan sebesar 10.77% pada jaringan dengan menggunakan *link aggregation* di sisi *server* dan *OpenFlow switch* dari *throughput* sebelum diberi *background traffic* hingga diberi *background traffic* sebesar 800 Mb.



Gambar 6. *Throughput* terhadap ukuran *background traffic*

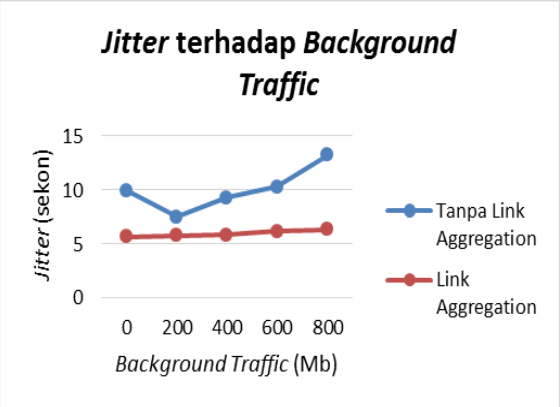
Pada Tabel 1 dan Gambar 6 terlihat bahwa *throughput* pada jaringan dengan menggunakan *link aggregation* lebih tinggi sebesar 22.53%, 22.61%, 25.81%, 20.95%, dan 16.56% dibandingkan dengan jaringan tanpa menggunakan *link aggregation.* Hal ini disebabkan karena kapasitas *link* pada jaringan dengan menggunakan *link aggregation* bertambah besar. Terdapat empat buah *link* fisik yang masing-masing memiliki kapasitas *link* 100 Mbps yang diikat menjadi satu membentuk suatu *link* logika yang berkapasitas 400 Mbps. Sehingga, *throughput* yang diterima akan menjadi lebih tinggi akibat kapasitas *link* dari *server* ke *OpenFlow switch* yang besar dibandingkan tanpa *link aggregation*. Sesuai dengan standar ITU-T G.1080, maka pada jaringan tanpa *link aggregation* untuk seluruh ukuran *background traffic*, nilai *throughput* yang dihasilkan dibawah standar, yakni 1.75 Mbps, sementara layanan video pada jaringan dengan *link aggregation* saat tidak ada *background traffic* dan saat ada *background traffic* dengan ukuran 200 Mb, 400 Mb,dan 600 Mb menghasilkan *throughput* yang diatas nilai standar rekomendasi ITU-T G.1080. Namun, saat ukuran *background traffic* meningkat menjadi 800 Mb, nilai *throughput* yang dihasilkan dibawah standar akibat kapasitas *link* yang padat oleh trafik sehingga mempengaruhi kualitas video yang ditampilkan di sisi *client*.

5.2. Variasi ukuran *background traffic* terhadap *jitter*

Pada Tabel 2 dan Gambar 7 ditunjukkan hubungan antara ukuran *background traffic* terhadap *jitter* dari sisi *client* yang melakukan *video streaming* secara *real-time*.

Tabel 2. *Jitter* Terhadap Ukuran *Background Traffic*

Background Traffic (Mb)	Jitter (ms)	
	Tanpa *Link aggregation*	Menggunakan *Link aggregation*
0	9,96	5,67
200	7,52	5,75
400	9,29	5,83
600	10,27	6,15
800	13,24	6,31



Gambar 7. *Jitter* terhadap ukuran *background traffic*

Dari Gambar 7 terlihat bahwa *jitter* secara keseluruhan mengalami kenaikan seiring dengan bertambahnya ukuran *background traffic*. Kondisi ini dilatarbelakangi semakin tingginya *background traffic* maka trafik menjadi padat saat melewati suatu *link*, dimana saat trafik sedang padat, maka jalur paket akan terhambat yang memunculkan *delay* yang nilanya bervariasi. *Variasi delay* ini yang menghasilkan nilai *jitter*. Untuk aplikasi seperti video real-time maka *jitter* harus diperhatikan.

Jitter di SDN tanpa *link aggregation* dan dengan *link aggregation* masing-masing meningkat sebesar 32.93% dan 11.28% dari keadaan *tanpa background traffic* sampai pada penambahan *background traffic hingga* sebesar 800 Mb.

Pada Tabel 2 dan Gambar 7 terlihat bahwa nilai *jitter* yang berasal dari lingkungan SDN dengan *link aggregation* lebih rendah dibandingkan dengan nilai jitter pada jaringan SDN tanpa metode *link aggregation*, yakni lebih rendah sebesar 43.07%, 23.53%, 37.24%, 40.11%, dan 52.34% pada masing-masing keadaan jaringan tanpa *background traffic*, dengan adanya *background traffic* berukuran 200 Mb, 400 Mb, 600 Mb, dan 800 Mb. Hal ini disebabkan karena kapasitas *link* setelah mengalami agregasi meningkat menjadi empat kali lipat kapasitas sebelumnya sehingga penambahan *background traffic* yang mengalir dari *server* ke *client* tidak menghasilkan jitter sebesar kondisi jaringan tanpa adanya *link aggregation* antara *server* dan *OpenFlow switch* yang membuat *link* fisik tersebut kewalahan menangani trafik data yang semakin besar. Mengacu pada Rekomendasi Standar ITU-T G.1080, nilai *jitter* untuk keadaan jaringan SDN tanpa *link aggregation* maupun dengan adanya *link aggregation* masih bisa ditoleransi oleh jaringan karena memiliki nilai jitter dibawah 50 milisekon.

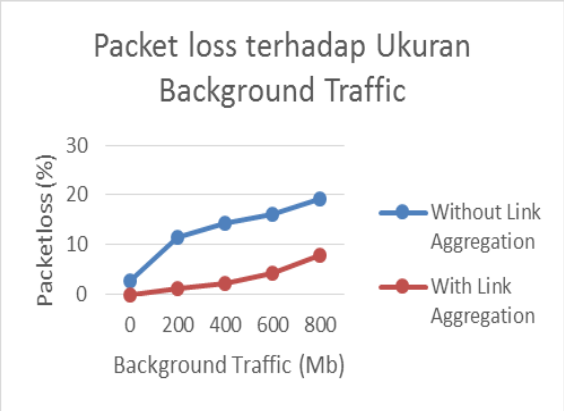
5.3. Variasi ukuran *background traffic* terhadap *packet loss*

Pada Tabel 3 dan Gambar 8 nilai *packet loss* di jaringan SDN dengan skenario *link aggregation* lebih rendah dibandingkan dengan nilai *packet loss* pada jaringan SDN tanpa *link aggregation*, yakni lebih rendah sebesar 100%, 89.06%, 84.79%, 73.93%, dan 58.93% pada masing-masing keadaan jaringan tanpa *background traffic*, dengan adanya *background traffic* berukuran 200 Mb, 400 Mb, 600 Mb, dan 800 Mb. Hal ini disebabkan karena kapasitas *link* setelah mengalami agregasi meningkat menjadi empat kali lipat kapasitas sebelumnya yang membuat *link* tidak sepadat seperti pada jaringan tanpa adanya *link aggregation*, sehingga penambahan *background traffic* yang mengalir dari *server* ke *client* tidak menghasilkan *packet loss* sebesar kondisi jaringan tanpa adanya *link aggregation* antara *server* dan *OpenFlow switch* yang membuat *link* fisik tersebut kewalahan menangani trafik data yang semakin besar.

Tabel 3. *Packet Loss* Terhadap Ukuran *Background Traffic*

Background Traffic (Mb)	*Packet Loss* (ms)	
	Tanpa *Link aggregation*	Menggunakan *Link aggregation*
0	2,6	0
200	11,52	1,26
400	14,27	2,17
600	16,19	4,22
800	19,24	7,9

Mengacu pada standar Cisco terhadap layanan *video streaming* yang digunakan, nilai *packet loss* pada kondisi tanpa *link* aggregation ketika diberikan *background traffic* tidak memenuhi standar sehingga video yang dihasilkan kurang baik. Untuk kondisi dengan *link aggregation*, nilai *packet loss* nya memenuhi standar minimum yang ditetapkan pada ukuran *background traffic* sebesar 200 Mb , 400 Mb, dan 600 Mb.



Gambar 8. *Packet Loss* terhadap ukuran *background traffic*

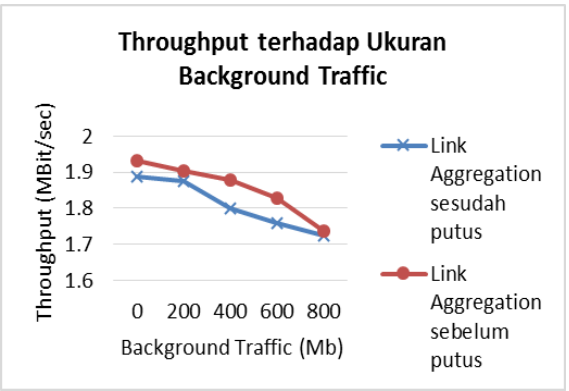
5.4. Uji redudansi dan *availability* jaringan

Uji redundansi dan *availability* jaringan dilakukan untuk mengetahui jaminan suatu jaringan mampu menghadirkan komunikasi yang dapat diandalkan dalam segala kondisi, bahkan di saat terjadi gangguan sekalipun. Pada skenario ini, gangguan komunikasi dihadirkan dalam bentuk pemutusan *link* fisik antara *server* layanan *video streaming* dengan *OpenFlow switch*.

Hasilnya, layanan video masih bisa dapat diterima dengan baik di sisi *client* walaupun salah satu *link* fisik putus pada jaringan SDN dengan *link aggregation*. Namun, akibat pemutusan satu *link* fisik yang sebelumnya terikat dengan *link* fisik lain membentuk suatu kanal logika menyebabkan *throughput* dan *jitter* dari sisi client yang mengakses layanan *video streaming* menurun. Tabel 4 dan Gambar 8 menunjukkan *throughput* untuk variasi ukuran *background* traffic pada saat salah satu *link* fisik yang tergabung dalam kanal logika yang teragregasi diputus.

Tabel 4. *Throughput* Terhadap Ukuran *Background Traffic* pada kondisi salah satu *Link* fisik terputus

Background Traffic (Mb)	Throughput (Mbit/sec)	
	Link aggregation sesudah putus	*Link aggregation* Sebelum putus
0	1,889	1,932
200	1,876	1,905
400	1,799	1,878
600	1,759	1,829
800	1,723	1,736



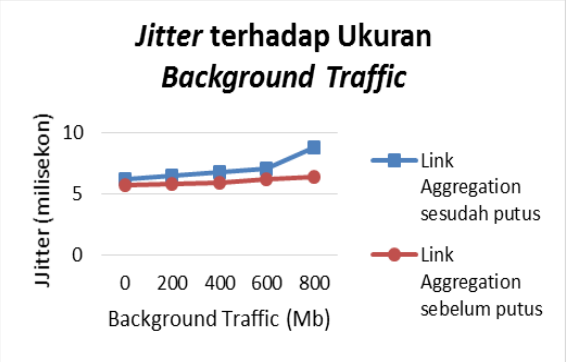
Gambar 9. *Throughput* terhadap ukuran *background traffic* pada jaringan SDN dengan kondisi salah satu *link* fisik putus

Pada Tabel 4 dan Gambar 9, nilai *throughput* pada jaringan SDN dengan menggunakan *link aggregation* saat terjadi pemutusan salah satu *link* fisik lebih kecil sebesar 2.22%, 1.52%, 4.20%, 3.82%, dan 0.74 % dari nilai *throughput* pada jaringan SDN dengan menggunakan *link aggregation* tanpa pemutusan salah satu *link* fisik. Hal ini disebabkan karena kapasitas *link* menurun akibat terjadinya pemutusan *link* fisik yang sebelumnya

teragregasi dan terikat dengan *link* fisik lain membentuk satu kanal logika, sehingga kapasitas *link* setelah terjadi pemutusan adalah sebesar 300 Mbps. Dari nilai *throughput* yang didapat saat terjadi pemutusan, terlihat bahwa *throuhput* yang memenuhi rekomendasi standar ITU-T G.1080 hanya tercapai pada saat tanpa adanya *background traffic*.

Tabel 5. *Jitter* Terhadap Ukuran *Background Traffic* pada kondisi salahsatu *Link* fisik terputus

***Background Traffic* (Mb)**	***Jitter* (ms)**	
	***Link aggregation* sesudah putus**	***Link aggregation* Sebelum putus**
0	6,12	5,67
200	6,48	5,75
400	6,73	5,83
600	7,04	6,15
800	8,77	6,31

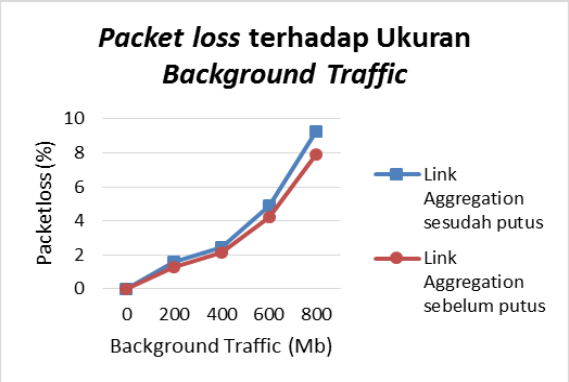


Gambar 10. *Jitter* terhadap ukuran *background traffic* pada jaringan SDN dengan kondisi salah satu *link* fisik putus

Pada Tabel 5 dan Gambar 10, nilai *jitter* pada jaringan SDN dengan menggunakan *link aggregation* saat terjadi pemutusan salah satu *link* fisik lebih besar sebesar 7.93%, 12.69%, 15.43%, 14.47%, dan 38.98% dari nilai *jitter* pada jaringan SDN dengan menggunakan *link aggregation* tanpa pemutusan salah satu *link* fisik. Hal ini disebabkan karena kapasitas *link* menurun akibat terjadinya pemutusan *link* fisik yang sebelumnya teragregasi dan terikat dengan *link* fisik lain membentuk satu kanal logika, semakin tinggi nilai *background traffic* maka semakin padat trafik yang melewati suatu *link*, oleh sebab itu kondisi trafik padat, maka perjalanan paket tersendat. Dari nilai *jitter* yang didapat saat terjadi pemutusan, terlihat bahwa seluruh *jitter* pada seluruh ukuran *background traffic* memenuhi rekomendasi standar ITU-T G.1080 *background traffic*.

Tabel 6. *Packet Loss* Terhadap Ukuran *Background Traffic* pada kondisi salahsatu *Link* fisik terputus

***Background Traffic* (Mb)**	***Packet loss* (%)**	
	***Link aggregation* sesudah putus**	***Link aggregation* Sebelum putus**
0	0	0
200	1,58	1,26
400	2,42	2,17
600	4,88	4,22
800	9,23	7,9



Gambar 11. *Packet Loss* terhadap ukuran *background traffic* pada jaringan SDN dengan kondisi salah satu *link* fisik putus

Pada Tabel 6 dan Gambar 11, nilai *packetloss* pada jaringan SDN dengan menggunakan *link aggregation* saat terjadi pemutusan salah satu *link* fisik lebih besar sebesar 0%, 25.40%, 11.52%, 15.64%, dan 16.83% dari nilai *packet loss* pada jaringan SDN dengan menggunakan *link aggregation* tanpa pemutusan salah satu *link* fisik. Hal ini disebabkan karena kapasitas *link* menurun akibat terjadinya pemutusan *link* fisik yang sebelumnya teragregasi dan terikat dengan *link* fisik lain membentuk satu kanal logika, semakin tinggi nilai *background traffic* maka semakin padat trafik yang melewati suatu *link*, dimana saat trafik yang lewat banyak, maka akan berdampak

ke terhambatnya perjalanan paket akan terhambat dan mengalami kehilangan paket. Dari nilai *packet loss* yang didapat saat terjadi pemutusan, terlihat bahwa seluruh *packet loss* pada seluruh ukuran *background traffic* telah memenuhi standar minimumnya yaitu dibawah 5% pada skenario ada *background traffic* dan skenario melewatkan *background traffic* sebesar 200 Mb, 400 Mb, dan 600 Mb.

Tabel 7. Waktu pemulihan jaringan saat salah satu *link* terputus

***Background Traffic* (Mb)**	***Waktu Pemulihan Jaringan (sekon)***
0	66,04
200	74,74
400	79,52
600	91,07
800	101,23

Tabel 7 menunjukkan waktu yang dibutuhkan untuk melakukan pemulihan jaringan bila salah satu *link* diputus untuk setiap *background traffic* yang diberikan. Dengan nilai *background traffic* yang diberikan sebesar 200 Mb, 400 Mb, 600 Mb, dan 800 Mb. Terlihat bahwa semakin tinggi *background traffic* yang dilewatkan maka akan membutuhkan waktu lama untuk melakukan pemulihan jaringan. Dari data yang diperoleh, waktu pemulihan jaringan telah mengalami kenaikan sekitar 34.76% dari sebelum diberikan *background traffic* sampai diberikan *background traffic* sebesar 800 Mb.

6. KESIMPULAN

Berdasarkan penelitian yang dilakukan, diperoleh kesimpulan bahwa jaringan SDN yang memanfaatkan konsep *link aggregation* menghadirkan komunikasi data yang cepat dari sisi throughput, menghasilkan jitter yang rendah serta menghasilkan packetloss yang kecil dibandingkan dengan jaringan SDN tanpa memanfaatkan konsep *link aggregation*. Hal ini dilakukan untuk menambah kualitas layanan yang disediakan dan dinikmati oleh client, sehingga akan menambah nilai lebih dari sisi *Quality of Experience* pelanggan. Selain itu, jaringan dengan *link aggregation* menjamin keberlangsungan proses komunikasi meskipun terdapat salah satu *link* yang putus, yaitu dengan melihat waktu pemulihan jaringan yang cepat, sehingga meningkatkan redundansi dan kehandalan jaringan yang tentunya akan menjadi nilai lebih bagi setiap penyelenggara jaringan untuk melayani konsumennya.

DAFTAR PUSTAKA

[1] McKeown, N., T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker et al. 2008. OpenFlow: Enabling innovation in campus networks. SIGCOMM Comput. Commun. Rev. 38(2):69–74.

[2] Steinbacher. M and Bredel. M, "LACP Meets OpenFlow – Seamless *Link*," University of Applied Sciences, Kufstein Tirol, Austria.

[3] Jeffree. T, "Strawman/D2½: Changes/Additions to 802.3 required in order to specify *Link* Aggregation," Contribution to the November '98 Link Aggregation Meeting, IEEE, 1998.

[4] C. Balakrishnan och M. Manikandan, "*Link* Aggregation Strategies," International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 1, nr 10, pp. 199-203, 2012.

[5] "Link Aggregation Control Protocol (LACP) (802.3ad) for Gigabit Interfaces," Cisco, March 2007. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/ios/12_2sb/feature/guide/gigeth.html

[6] B. Zdrnja, "Malicious JavaScript Insertion through ARP Poisoning Attacks," IEEE Security & Privacy, vol. 7, nr 3, 2009.

[7] SysKonnect GmbH, White paper. "*Link* Aggregation according to IEEE Standard 802.3ad", Okt 2002.

[8] D. Pemberton, A. Linton S. and Russel, "Ryu OpenFlow Controller," University of Oregon.

[9] Benson. T, Akella. A, and Maltz. D, "Unraveling the complexity of network management," eProceedings of the 6th USENIX Symposium on Networked systems design and implementation, Berkeley, 2009.

[10] Ghodsi, A., S. Shenker, T. Koponen, A. Singla, B. Raghavan and J. Wilcox, "Intelligent design enables architectural

evolution," Proceedings of the 10th ACM Workshop on Hot Topics in Networks, New York, pp. 3:1–3:6, 2011.

[11] N. Feamster and H. Balakrishnan, "Detecting BGP configuration faults with static analysis," Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2, Berkeley, California, 2005.

[12] K. Butler, T. Farley, P. McDaniel och a. J. Rexford, "A survey of BGP security issues and solutions," Proceedings of the IEEE, 2010.

[13] D. Kreutz, F. M. Ramos, P. Verissimo, C. Rothenberg, S. Azdolmolky and S. Uhlig, "Software Defined Networking: A Comprehensive Survey," Proceedings of the IEEE, vol. 103, nr 1, pp. 14-76, 2015.

[14] ONF, "Open networking foundation," 2014. [Online]. Available: https://www.opennetworking.org/.

[15] T. Szigeti and H. Christina, "QoS Requirements of Video," End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs, Cisco Press, 2004.

Biodata Penulis

Rohmat Tulloh, S.T., M.T, staf pengajar pada Program Studi D3 Teknik Telekomunikasi, Fakultas Ilmu Terapan, Universitas Telkom. Ketertarikan dalam bidang Jaringan Komputer khususnya *Software Defined Network* dan *Internet of Things*.