

Performansi *High Availability* pada *Software Defined Network-Internet Protocol* untuk Topologi Jaringan Inti

Oktapani Panca Jaya ¹⁾, Ridha Muldina Negara ²⁾, Danu Dwi Sanjoyo ³⁾

^{1),2),3)}Fakultas Teknik Elektro, Universitas Telkom
Jl. Telekomunikasi No. 1, Bandung
Email : oktapanipj@gmail.com

Abstrak. *Software Defined Network-Internet Protocol (SDN-IP)* merupakan suatu teknologi untuk membantu peralihan jaringan tradisional ke jaringan berbasis *Software Defined Network (SDN)*. Jaringan berbasis SDN yang dikendalikan oleh kontroler bekerja sama dengan sebuah perangkat jaringan (*router*) untuk menerjemahkan paket routing *Border Gateway Protocol (BGP)* akan bertindak sebagai jaringan inti (*core network*) yang mampu menghubungkan beberapa *autonomous system (AS)* jaringan IP tradisional. Konektifitas pada jaringan inti harus selalu tersedia, dengan kata lain jaringan inti haruslah mampu terus berfungsi meski terjadi kegagalan pada salah satu komponennya. Kemampuan ini disebut dengan *high availability (HA)*. Pemilihan topologi sangat penting untuk menentukan ketersediaan link cadangan ketika terjadi kegagalan. Maka pada penelitian ini dilakukan pengujian dan analisis performansi HA pada simulasi jaringan SDN-IP kontroler ONOS dengan topologi jaringan inti 2-D Mesh dan Full Mesh. Tujuannya untuk melihat pengaruh pemilihan topologi jaringan inti terhadap performansi HA jaringan. Berdasarkan hasil pengujian dan analisis dapat disimpulkan bahwa pemilihan topologi berpengaruh pada performansi HA jaringan. SDN-IP pada topologi 2-D Mesh menghasilkan performansi HA yang lebih optimal ketimbang topologi Full Mesh berdasarkan parameter *Failover Delay* dan *Failover Overhead Size*.

Kata kunci: *Software Defined Networking, ONOS, SDN-IP, High Availability, Autonomous System*

1. Pendahuluan

Software-Defined Networking (SDN)[1] adalah sebuah konsep arsitektur jaringan yang sedang berkembang saat ini, dimana *control plane* dan *data plane* dipisahkan secara fisik. Saat ini, untuk mencapai implementasi SDN secara keseluruhan pada jaringan IP yang sudah ada dalam waktu singkat merupakan hal yang mustahil. Oleh karena itu, penerapan SDN dilakukan secara bertahap (inkremental) dengan cara menambahkan SDN pada jaringan IP yang telah ada [2]. Jaringan SDN tentu harus mampu berkomunikasi dan bertukar informasi *routing* dengan jaringan *native*. Salah satu solusi untuk masalah ini adalah dengan menggunakan aplikasi *peer SDN-IP*. SDN-IP adalah sebuah aplikasi yang sedang dikembangkan oleh *Open Network Operating System (ONOS)*, yang memungkinkan jaringan berbasis SDN untuk terkoneksi dan bertukar informasi *routing* dengan *Autonomous System (AS)* jaringan *native* menggunakan *Border Gateway Protocol (BGP)* [3].

Pada penelitian ini jaringan SDN berperan sebagai jaringan inti yang menghubungkan beberapa AS. Jaringan inti merupakan sentral dimana beberapa jaringan yang berbeda saling terhubung satu sama lain, sehingga seluruh pertukaran data antar jaringan akan melewati jaringan inti terlebih dahulu [4]. Oleh karena itu jaringan inti diharapkan memiliki *high availability (HA)* sehingga trafik data antar AS selalu mengalir tanpa masalah. HA adalah kemampuan jaringan untuk selalu berfungsi dan dapat diakses ketika dibutuhkan meskipun salah satu komponennya mengalami kegagalan [5]. Ketika kegagalan *link* terjadi pengiriman informasi yang biasa melewati *link* tersebut harus mencari jalur alternatif. Pemilihan topologi sangat berpengaruh pada ketersediaan *link* redundant dan proses penentuan *routing path* yang baru dapat memakan waktu (*delay*) sehingga dapat mempengaruhi kinerja jaringan. Oleh karena itu pada penelitian ini akan dianalisis performansi dan HA pada simulasi SDN-IP pada jaringan inti yang memiliki topologi yang berbeda. Tujuan dari penelitian ini adalah mampu merancang dan menganalisis performansi *high availability (HA)* dari sistem SDN-IP sebagai topologi jaringan inti yang menghubungkan sejumlah AS. Serta mengetahui apakah pemilihan topologi jaringan inti memberikan pengaruh yang signifikan terhadap performansi *availability* jaringan dan topologi manakah yang menghasilkan performansi HA terbaik. Diharapkan hasil penelitian ini

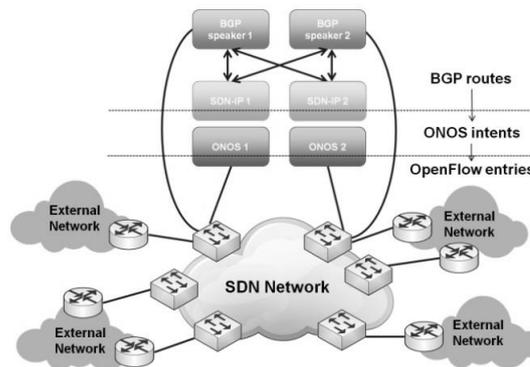
dapat menjadi salah satu referensi dalam pengembangan dan implementasi dari jaringan inti yang menerapkan SDN-IP sebagai metode peralihan jaringan tradisional menuju jaringan berbasis SDN.

2. Pembahasan

Pada bagian ini akan dibahas perancangan sistem serta penggambaran topologi yang diujikan. Selain itu akan dibahas cara kerja dari sistem yang dibuat dan hasil analisis keluaran sistem yang diujikan.

2.1. Analisis Kinerja Sistem SDN-IP

SDN-IP adalah salah satu aplikasi ONOS yang memungkinkan interkoneksi antara jaringan *Software Defined Network* dengan jaringan native menggunakan *Border Gateway Protocol* (BGP) [3]. Aplikasi ini memungkinkan beberapa AS untuk saling bertukar informasi *routing* dan mengirimkan trafik menuju dan melalui jaringan SDN.



Gambar 1. Arsitektur Jaringan SDN-IP [3]

Jika ditinjau berdasarkan arsitektur jaringan SDN-IP pada gambar 1 bahwa jaringan SDN terdiri atas beberapa komponen, yaitu Kontroler ONOS, *Router BGP Speaker* dan Open vSwitch sebagai infrastruktur data plane. Kontroler ONOS terhubung secara langsung dengan *router BGP Speaker* melalui hubungan iBGP. *Router BGP Speaker* ini juga terhubung ke salah satu *switch* di *data plane* untuk menerima paket routing dari *external BGP router* dan menerjemahkannya menjadi *intents request* ke kontroler ONOS melalui aplikasi SDN-IP.

Sehingga secara eksternal, melalui sudut pandang BGP, jaringan berbasis SDN pada penelitian ini dianggap sebagai sebuah *Autonomous System* (AS) yang berperilaku seperti jaringan inti (*core network*) yang meneruskan trafik dari berbagai jaringan eksternal yang terhubung melaluinya. Untuk melakukan fungsi ini SDN-IP menyediakan mekanisme integrasi antara BGP dan ONOS. Pada tingkat protokol, SDN-IP berperilaku sebagai *BGP Speaker* biasa. Dari sudut pandang ONOS, SDN-IP hanyalah sebuah aplikasi yang menggunakan layanannya untuk menginstall dan memperbarui *forwarding state* pada *SDN data plane*.

Pada jaringan SDN-IP, terdapat satu atau lebih *BGP Speaker* internal. *BGP Speaker* dapat berupa *BGP routers* atau *software* yang mampu mengimplementasikan BGP. *Speaker* menggunakan eBGP untuk bertukar informasi *routing* BGP dengan *border routers* dari jaringan eksternal terdekat, dan iBGP untuk menyebarluaskan informasi tersebut antara sesama *Speaker* dan dengan *SDN-IP application instances*.

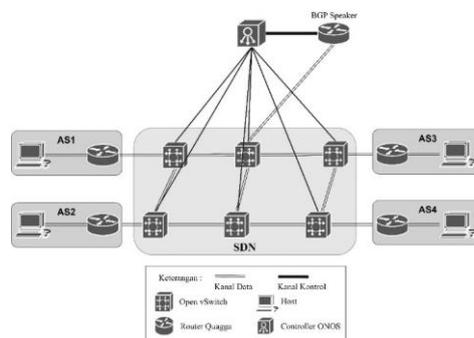
Rute yang ditampilkan oleh *border routers* eksternal berasal dari jaringan eksternal lalu diterima oleh *BGP Speakers* pada jaringan SDN-IP, diproses dan ditampilkan kembali pada jaringan eksternal lainnya. Dengan cara yang sama, rute tersebut juga ditampilkan kepada *SDN-IP application instances* yang bertindak sebagai *iBGP peers*. Rute terbaik untuk setiap destinasi ditentukan oleh SDN-IP application berdasarkan aturan iBGP, dan diterjemahkan kedalam sebuah *ONOS Application Intent Request*. ONOS menerjemahkan *Application Intent Request* menjadi aturan *forwarding* pada *data plane*. Aturan ini digunakan untuk meneruskan trafik transit sata yang terjadi antara jaringan IP yang terkoneksi.

Selain itu, SDN-IP menciptakan ONOS *Intents* yang memungkinkan router BGP eksternal untuk terhubung dengan BGP *Speakers* melalui *data plane*. Sehingga, BGP dengan sendirinya juga dapat mendeteksi *failures* yang terjadi di *data plane* yang mempengaruhi trafik transit data dan mengambil tindakan yang sesuai atas hal tersebut.

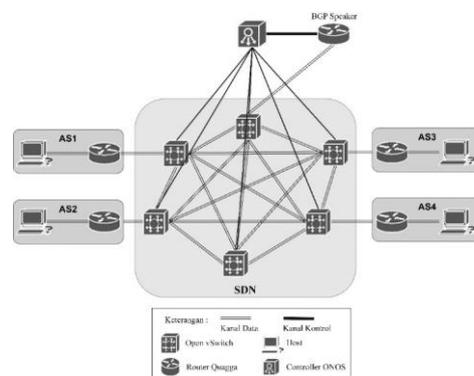
2.1. Pemilihan topologi dan perancangan sistem

Sistem akan dibangun diatas sebuah *Virtual Machine (VM) Guest* dengan sistem operasi Ubuntu Desktop 16.04 LTS yang diinstal pada *hypervisor* Oracle VM VirtualBox diatas komputer *Host*. Komputer host menggunakan sistem operasi Windows 10 Home. Pada VM Guest inilah diinstal controller ONOS, emulator Mininet, Open vSwitch dan *routing machine* Quagga.

Penelitian ini akan menggunakan simulasi dua jaringan dengan dua topologi jaringan inti yang berbeda sebagai perbandingan, yaitu topologi *2-D Mesh* dan *Full Mesh*. Pada penelitian yang dilakukan oleh Jonathan [6] mengenai SDN-IP, topologi jaringan inti SDN yang digunakan adalah topologi *2-D Mesh*. Namun menurut penelitian yang dilakukan oleh David S. Lee yang berjudul *Network Topology Analysis* [7], topologi *2-D Mesh* memiliki beberapa kekurangan, yaitu: Ketidaksetaraan jumlah *link* antar node, distribusi trafik tidak merata dan meskipun jumlah *link* redundant pada topologi ini terbilang cukup, kegagalan dua *link* dapat mengisolasi node yang terletak di sudut. Kekurangan yang dimiliki oleh topologi *2-D Mesh* ini tidak baik untuk availabilitas jaringan, terutama jaringan inti. Maka penulis melakukan penelitian untuk menguji availabilitas jaringan ketika menggunakan topologi *2-D Mesh* pada jaringan inti SDN-IP. Sebagai pembandingan, penulis memilih topologi *Full Mesh*. Alasan dipilihnya topologi ini dikarenakan topologi *Full Mesh* memiliki jumlah *link* yang setara pada setiap nodenya dan merupakan topologi dengan redundansi terbaik. Jaringan yang dibentuk akan terdiri dari 4 host, 6 Open vSwitch sebagai perangkat jaringan inti, 4 *router* Quagga dan sebuah kontroler ONOS. Untuk lebih jelas, *link* antar node pada kedua topologi dapat dilihat pada gambar 2 untuk topologi *2-D Mesh* dan gambar 3 untuk topologi *Full Mesh*.



Gambar 2. Topologi *2-D Mesh*



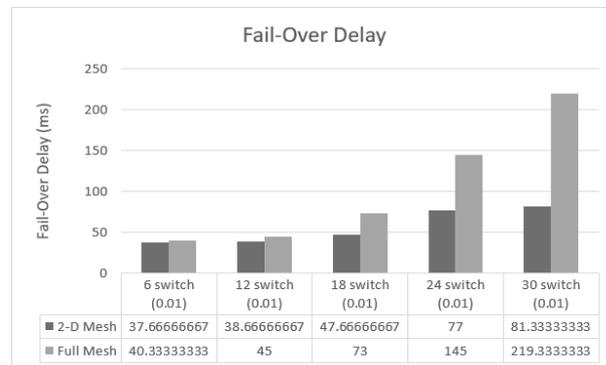
Gambar 3. Topologi *Full Mesh*

Node dengan nama “BGP *Speaker*” adalah *router* yang berfungsi sebagai *Internal BGP Speaker*. *Internal BGP Speaker* terletak didalam jaringan SDN dan terhubung dengan salah satu Open vSwitch di dalam jaringan SDN. *Internal BGP Speaker* akan mempelajari paket *routing* BGP yang masuk ke jaringan SDN dan menerjemahkan paket *routing* tersebut kepada kontroler ONOS sebagai *intents*

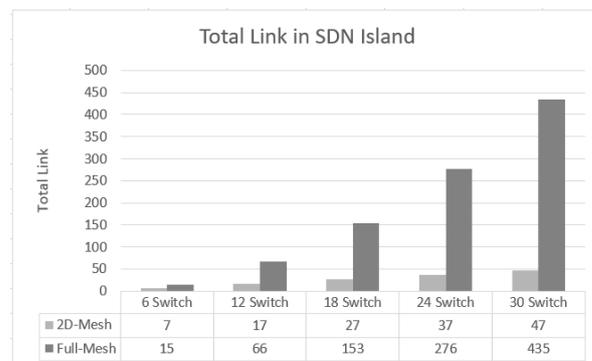
request melalui aplikasi SDN-IP. Empat node *router* lainnya (R1 sampai R4) adalah External BGP *Routers*. Merupakan *router* tepi yang terletak didalam suatu *autonomous system* (AS) yang terhubung dengan jaringan SDN menggunakan protokol eBGP dan akan bertukar trafik data dengan jaringan SDN yang berperan sebagai jaringan inti. Dibalik setiap *router* tepi terdapat satu komputer host yang dinamai “h1” sampai “h4”.

2.3. Pengujian dan Analisis *Failover Delay*

Failover Delay merupakan waktu jeda pengiriman sebuah paket akibat kegagalan *link* yang terjadi pada rute yang dilewati oleh aliran data, hingga pengiriman data kembali berlanjut ketika proses *reroute* telah berhasil. Pengukuran dilakukan pada host1 di AS1 yang bertindak sebagai pengirim kepada host2 di AS2. Pengukuran ini dilakukan untuk mengetahui *delay* sejak laporan terjadinya *link failure* oleh *switch* kepada kontroller hingga *flow* baru terpasang pada *switch* dan pengiriman kembali berlangsung. Pengukuran parameter ini dilakukan pada kedua topologi jaringan inti dengan jumlah *switch* yang berbeda-beda. Dari hasil pengujian pada gambar 4, kinerja SDN-IP ONOS dalam menjalankan fungsi redundancy masih kurang stabil dan belum memiliki skalabilitas yang baik. Hal ini dapat terlihat dari kenaikan *failover delay* pada topologi *Full Mesh* yang cukup signifikan. Ketika jumlah *switch* naik dari 18 *switch* ke 24 *switch*, terjadi kenaikan *delay* sebesar 53.1% dari waktu *delay* sebelumnya. Namun lain halnya dengan topologi *2-D Mesh*, kenaikan *delay* cukup stabil meskipun jumlah *switch* naik hingga lima kali lipat dari semula. Hal ini terjadi akibat jumlah *alternative link* yang meningkat cukup signifikan, dua kali lipat lebih banyak, pada topologi *Full Mesh* setiap penambahan *switch* terjadi seperti yang dapat dilihat pada gambar 5. Proses *reroute* pada kontroller ONOS harus mempertimbangkan lebih banyak *alternative link* pada proses komputasi untuk menentukan path yang baru [8]. Sehingga waktu pembentukan *path* baru untuk sebuah *intents* yang mengalami *link failure* menjadi lebih lama. Sedangkan pada topologi *2-D Mesh* kenaikan jumlah *link* stabil, sepuluh *link* per-enam *switch*. Sehingga besar kenaikan *delay* cukup stabil.



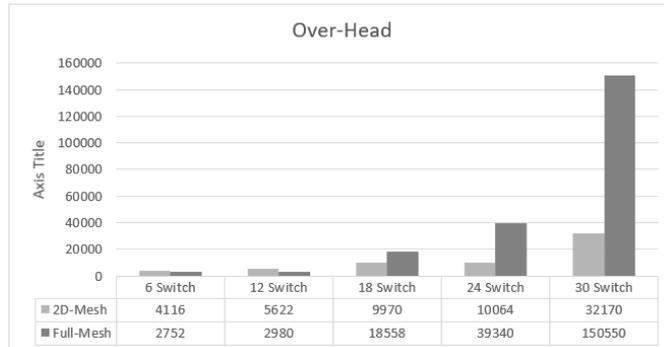
Gambar 4. Pengukuran *Fail-Over Delay*



Gambar 5. Perbandingan Total *Link*

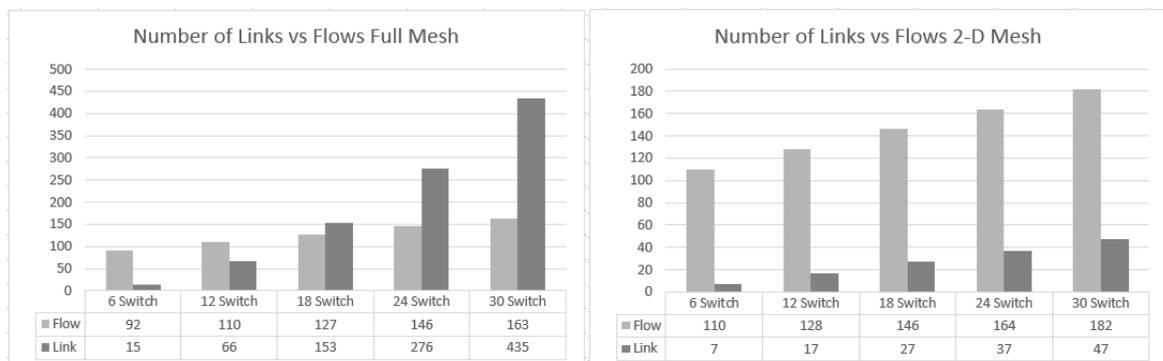
2.4. Pengujian dan Analisis *Failover Overhead*

Overhead size pada penelitian ini merupakan jumlah paket kontrol yang dikirimkan oleh kontroller ke *data-plane* dalam proses mengatasi *failure* dan melakukan *path modification* pada *intents* setelah terjadi pemutusan *link*. Pengukuran dilakukan dengan membangkitkan trafik data selama beberapa waktu antara dua host pada AS yang berbeda. Besar ukuran paket *overhead* dihitung dari laporan *switch* bahwa ada *link* yang putus berupa paket *openflow* PORT_STATUS, hingga paket modifikasi jalur berupa paket *openflow* FLOW_MOD, yang terakhir.



Gambar 6. *Failover Overhead Size*

Hasil pengukuran dapat dilihat pada gambar 6. terlihat bahwa semakin banyak jumlah *switch* dan *link* pada kedua topologi jaringan inti, maka semakin besar ukuran trafik kontrol. Namun kenaikan trafik *overhead* yang dialami pada topologi *Full Mesh* dan topologi *2-D Mesh* sangat berbeda. Terlihat pada gambar 6 bahwa saat jumlah *switch* pada jaringan inti antara 6 sampai 12 *switch*, besar trafik *overhead* pada topologi *2-D Mesh* lebih besar ketimbang topologi *Full Mesh*. Hal ini terjadi karena perbedaan banyak jumlah *link* dengan jumlah *flow* ketika jumlah *switch* masih diantara 6-12 *switch* stabil. Dimana jumlah *flow* lebih banyak dibanding jumlah *link*nya. Namun ketika jumlah *switch* naik menjadi 18 hingga 30 *switch*, trafik *overhead* pada topologi *Full Mesh* naik secara signifikan hingga 26% dari semula. Hal ini disebabkan karena kenaikan jumlah *link* yang lebih besar ketimbang jumlah *flow* seperti yang dapat dilihat pada gambar 7. Jika kita memeriksa *flow* table pada masing-masing *switch* didalam jaringan, kita dapat melihat bahwa ketika sebuah *switch* tidak memiliki jalur langsung menuju AS eksternal dan tidak terpilih dalam jalur pengiriman data, maka *flow* table pada *switch* tersebut akan kosong. Semakin besar jumlah *switch* di dalam jaringan maka semakin banyak *switch* yang tidak memiliki *flow entries* menuju jaringan AS eksternal, sehingga selisih antara jumlah *flow* dan jumlah *link* semakin besar. Selain itu, keadaan ini menggambarkan bahwa banyak jalur yang tidak terpakai dan kemungkinan perubahan trafik untuk berpindah jalur pada proses reroute fail-over ikut melibatkan perpindahan *switch* sehingga meningkatkan jumlah *packet_in* yang masuk ke kontroller



Gambar 7. Perbandingan *Link* Terhadap *Flow*

Pada topologi *2-D Mesh*, dapat dilihat pada gambar 7 bahwa kenaikan perbandingan jumlah *link* terhadap *flow* terlihat stabil seiring dengan bertambahnya jumlah *switch*. Tidak terjadi anomali seperti

pada topologi *Full Mesh*. Hal ini membuat kenaikan jumlah trafik *overhead* lebih stabil dan rendah dibandingkan dengan topologi *Full Mesh*. Perbedaan ukuran traffic *overhead* pada kedua topologi paling besar terjadi ketika jumlah *switch* sebanyak 30 buah, dimana besar trafik *overhead* pada topologi *2-D Mesh* 21.36% lebih kecil ketimbang topologi *Full Mesh*. Hal ini juga dikarenakan selisih antara jumlah *flow* dengan jumlah *link* yang sangat besar pada topologi *Full Mesh* sehingga trafik *overhead* naik dengan signifikan.

3. Kesimpulan

Berdasarkan hasil pengujian dan analisis dapat disimpulkan bahwa pemilihan topologi berpengaruh pada performansi HA jaringan. Hasil pengujian menunjukkan topologi *2-D Mesh* memiliki Failover *Delay* yang stabil dibawah 140 ms dibanding topologi *Full Mesh* yang melewati angka 200 ms. *Overhead Size* pada topologi *2-D Mesh* juga 21% lebih rendah ketimbang topologi *Full Mesh*. Sehingga berdasarkan hasil penelitian dapat disimpulkan bahwa pemilihan topologi berpengaruh pada performansi HA sistem SDN-IP kontroler ONOS dan penggunaan topologi *2-D Mesh* memberikan performansi HA yang lebih baik ketimbang topologi *Full Mesh*.

Ucapan Terima Kasih

Penulis mengucapkan puji syukur kepada Allah SWT karena atas rahmat dan karunia-Nya penulis dapat menyelesaikan jurnal penelitian ini tepat pada waktunya. Penulis mengucapkan terima kasih yang sebesar-besarnya kepada kedua orang tua penulis, atas doa dan dukungan yang tiada henti diberikan kepada penulis. Selain itu penulis juga berterima kasih kepada rekan sesama asisten Laboratorium Core Network dan *civitas academica* Universitas Telkom yang telah membantu penulis selama melaksanakan penelitian ini.

Daftar Pustaka

- [1]. ONF. Software-Defined Networking: The New Norm for Networks, <https://www.opennetworking.org>, diakses tgl 27 Januari 2018.
- [2]. Wei, Y., Zhang, X., Xei, L., & Leng, S. (2016). Energy-aware Traffic Engineering in Hybrid SDN/IP Backbone Network. *IEEE Journal of Communications and Networks*, 559-566.
- [3]. Hart, J. (2016). *ONOS Apps and Use Cases: SDN-IP*, Open Network Operating System (ONOS) Wiki: <https://wiki.onosproject.org>, diakses tgl 29 Januari 2018.
- [4]. IEEE Standards Coordinating Committee, IEEE Standard Glossary of Software Engineering Terminology, IEEE, 1990.
- [5]. S. Angelescue and A. Swerczek, "Network Design," in CCNA Certification All-In-One for Dummies, New Jersey, John Wiley & Sons, Inc., 2010, p. 116.
- [6]. P. Lin, J. Hart, U. Krishnaswamy, T. Murakami, M. Kobayashi, A. Al-Shabibi, K.-C. Wang and J. Bi, "Seamless Internetworking of SDN and IP," in SIGCOMM '13 Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM, New York, NY, USA, 2013.
- [7]. D. S. Lee and J. L. Kalb, "Network Topology Analysis," Sandia National Laboratories, Albuquerque, New Mexico, 2008.
- [8]. D. Sanvito, D. Moro, M. Gulli, I. Filippini and A. Capone, "ONOS Intent Monitor and Reroute service: enabling plug&play routing logic," in 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), Montreal, QC, Canada, 2018.