ANALISIS PERFORMANSI IN-BAND NETWORK TELEMETRY PADA INFRASTRUKTUR JARINGAN TERPROGRAM BERBASIS P4

PERFORMANCE ANALYSIS OF IN-BAND NETWORK TELEMETRY IN P4-BASED PROGRAMMABLE NETWORK INFRASTRUCTURE

M. Luthfi As Syafii [1], Sofia Naning Hertiana[2], Aris Cahyadi Risdianto[3]
[1,2] Universitas Telkom, Bandung, [3]National University of Singapore (NUS)
muhammadluthfias@student.telkomuniversity.ac.id[1], **sofiananing@telkomuniversity.ac.id[2]**, **aris@comp.nus.edu.sg[3]**

Abstrak

Streaming network telemetry adalah pendekatan baru untuk memantau kondisi jaringan, menggunakan metode pushing "push the data" dimana perangkat sebagai agent mengirimkan data terus menerus secara realtime ke platform terpusat dan menggunakan teknologi SDN dengan pendekatan bottom-up programming, tools yang biasa digunakan adalah NetFlow atau sFlow, namun karena streaming sehingga mengakibatkan size overhead yang tinggi dan metode bottom-up terbatas hanya pada fitur yang disediakan (fixes table, pipelines, match fields), sedangkan kebutuhan operator semakin kompleks, seperti end-to-end visualization. Untuk mengatasi masalah tersebut adalah dengan menggunakan In-band Network Telemetry (INT), dengan menyisipkan sejumlah kecil informasi (INT header) langsung ke dalam paket yang melewati perangkat jaringan berdasarkan flows, packet, protocols, sampai high-level names. Dengan begitu memungkinkan untuk end-to-end visualization. Metode top-down programming memungkinkan adanya INT dan menggunakan pemograman Bahasa P4. Berdasarkan hasil yang didapat pada penelitian ini, P4-INT dapat mengurangi storage overhead dibandingkan dengan sFlow-RT dan port-mirroring, namun jika hanya ada satu trafik lebih baik menggunakan sFlow-RT. Pada P4-INT setiap paket yang lewat perangkat jaringan akan ditambahkan header INT sehingga menimbulkan protocol overhead, walaupun pada P4-INT terdapat protocol overhead, hal tersebut memungkinkan adanya hop latency untuk end-to-end visualization

Kata kunci : ***SDN, P4, In-Band Network Telemetry, ONOS, P4Runtime***

Abstract

Streaming network telemetry is a new approach to monitoring network conditions, using the "push the data" method where the device as an agent sends data continuously in realtime to a centralized platform and using SDN technology with a bottom-up programming approach, the commonly used tool is NetFlow or sFlow. However, due to streaming, it results in high size overhead and the bottom up method is limited only to the features provided (fix tables, pipelines, matching fields), while operator requirements are increasingly complex, such as end-to-end visualization. To solve this problem is to use In-band Network Telemetry (INT), by inserting a small amount of information (INT header) directly into packets that pass through network devices based on flows, packets, protocols, to high-level names. This allows for end-to-end visualization. The top-down programming method allows for INT and use P4 Programming language. Based on the results obtained in this study, P4-INT can reduce storage overhead compared to sFlow-RT and port-mirroring, but if there is only one traffic it is better to use sFlow-RT. In P4-INT every packet passing through a network device is appended with an INT header, causing protocol overhead, although in P4-INT there is protocol overhead, allowing hop latency for end-to-end visualization.

Keywords: : SDN, P4, In-Band Network Telemetry, ONOS, P4Runtime

1. Pendahuluan

1.1.Latar Belakang

Pada saat ini cloud computing sangat berperan penting dalam penerapan layanan baru dan telah menjadi infrastruktur bagi masyarakat digital [1]. Seperti layanan *infrastructure as a service* (IaaS), *Platform as a Service* (PaaS), *Software as a Service* (SaaS), dan lain-lain [2]. Sehingga mendorong perusahaan untuk membangun data centernya sendiri dan mendorong dari segi ekonomi telah menyebabkan munculnya berbagai penyedia layanan data center (seperti *Amazon Web Services, Google Cloud Platform*, dan *Microsoft Azure*)

Streaming network telemetry adalah pendekatan baru untuk memantau kondisi jaringan, menggunakan metode *pushing push the data"* [3]. Dimana perangkat sebagai agent mengirimkan data terus menerus secara *realtime* ke *platform* terpusat dan menggunakan teknologi SDN dengan pendekatan *bottom-up programming* [4], tools yang biasa digunakan adalah NetFlow atau sFlow, namun karena *streaming* sehingga mengakibatkan *size overhead* yang tinggi dan metode *bottom-up* terbatas hanya pada fitur yang disediakan (*fixes table, pipelines, match fields*), sedangkan kebutuhan operator semakin kompleks, seperti *end-to-end visualization* [5][6][7].

Untuk mengatasi masalah tersebut adalah dengan menggunakan *In-band Network Telemetry* (INT), dengan menyisipkan sejumlah kecil informasi (INT *header*) langsung ke dalam paket yang melewati perangkat jaringan berdasarkan *flows, packet, protocols*, sampai *high-level names* [8]. Dengan begitu memungkinkan untuk *end-to-end visualization*. Metode *top-down programming* memungkinkan adanya INT, P4 *language* merupakan bahasa pemograman untuk mendukung hal ini [5]. Penelitian ini dilakukan untuk menganalisis performansi *In-band network telemetry* pada infrastruktur jaringan terprogram berbasis P4.

Berdasarkan hasil yang didapat pada penelitian ini, P4-INT dapat mengurangi *storage overhead* dibandingkan dengan sFlow-RT dan *port-mirroring*, namun jika hanya ada satu trafik lebih baik menggunakan sFlow-RT. Pada P4-INT setiap paket yang lewat perangkat jaringan akan ditambahkan *header* INT sehingga menimbulkan *protocol overhead*, walaupun pada P4-INT terdapat *protocol overhead*, hal tersebut memungkinkan adanya *hop latency* untuk *end-to-end visualization*

2. Dasar Teori

1. *Software Defined Networking* (SDN)

Software Defined Networking (SDN) adalah pendekatan jaringan yang dapat diprogram sehingga mendukung pemisahan antara control plane dan data plane melalui antarmuka standar [9]. SDN memungkinkan perilaku jaringan dapat dikontrol secara programmable dan centralized menggunakan software applications melalui open interface [10].

2. OpenFlow

OpenFlow merupakan salah satu jenis dari APIs dalam jaringan SDN yang digunakan untuk mengontrol/mengatur traffic flows pada switch dan disebut dengan southbound interfaces [11].

1. Programming Protocol-Independent Packet Processors (P4)

Programming Protocol-Independent Packet Processors (P4) merupakan domain-specific programming language yang dirancang untuk mendeskripsikan bagaimana perilaku (behavior) pemrosesan paket pada data plane [12].

2. P4Runtime

P4Runtime merupakan Application Programming Interface (API) berbasis gRPC dan Protocol Buffers (protobuf) yang menghubungkan antara P4 defined switch dengan control plane. [13]

3. Behavioral Model version 2 (BMv2)

BMv2 merupakan P4 software switch versi kedua. BMv2 ditulis menggunakan Bahasa pemrograman C++11. BMv2 menerima input dari file berformat JavaScript Object Notation (JSON) yang dihasilkan oleh P4 compiler. [14]

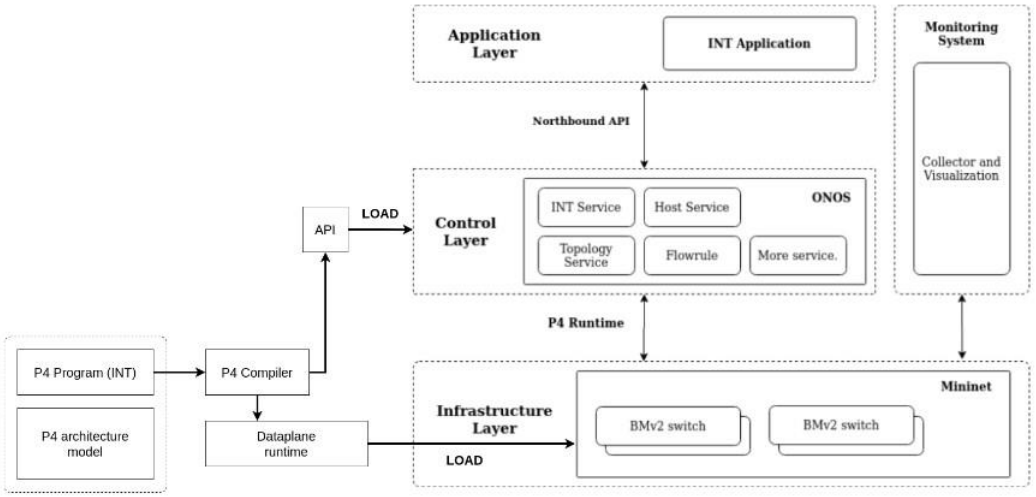
4. In-band Network Telemetry (INT)

In-band Network Telemetry (INT) merupakan framework atau kerangka kerja yang dirancang untuk memungkinkan pengumpulan dan pelaporan keadaan jaringan oleh data plane, tanpa memerlukan intervensi control plane karena semua update informasi akan dikirimkan dari data plane. In-band Network Telemetry dikembangkan oleh P4.org pada tahun 2015, hingga saat INT terus dikembangkan dan dibahas di IETF, komunitas, dan industri. [15]

3. Pembahasan

1. Desain Sistem

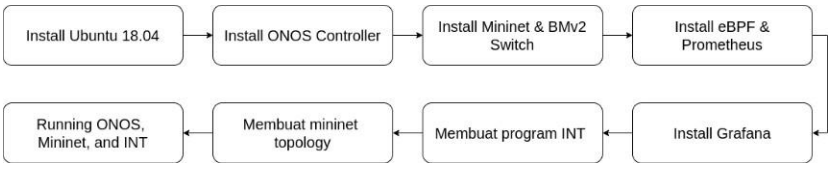
Sistem yang dirancang pada Tugas Akhir ini digunakan untuk menguji dan menganalisis performansi in-band network telemetry pada infrastruktur jaringan terprogram berbasis P4 language. Gambar sistem Tugas Akhir ini dapat dilihat pada Gambar 3.1, secara umum infrastructure layer menggunakan data plane BMv2 switch untuk dapat mendukung P4 language. Open Network Operating System (ONOS) digunakan sebagai SDN controller. Kemudian pada application layer terdapat INT application, dan untuk collector/visualization menggunakan Prometheus dan grafana. Secara umum komponen-komponen dalam desain sistem terbagi menjadi application, control plane, data plane, dan collector. Penjelasan tiap-tiap komponen sebagai berikut:



Gambar 3.1 Desain Sistem

2. Blok diagram

Untuk alur blok diagram pada gambar 3.2 langkah pertama adalah install ubuntu 18.04, kemudian install SDN Controller menggunakan ONOS, kemudian untuk infrastructure layer menggunakan menggunakan mininet dan BMv2 switch, install eBPF dan prometheus untuk data collector, dan install grafana untuk visualisasi data atau dashboard. Kemudian selanjutnya adalah pada bagian in-band network telemetry, dengan membuat program INT dengan P4 untuk BMv2 switch, kemudian membuat topology menggunakan mininet dengan switch BMv2 yang sudah dibuat sebelumnya, yang terakhir adalah kita perlu menjalankan scenario in-band network telemetry.



Gambar 3.2 Diagram Blok

3. Desain Perangkat Keras

Dalam Desain Perangkat Keras, perangkat menggunakan server KVM untuk menjalankan virtualisasi dengan spesifikasi perangkat keras yang digunakan untuk penelitian sesuai dengan tabel dibawah.

Tabel 3.1 Spesifikasi Virtual Machine

No.	komponen	spesifikasi
1.	*Processor*	12 vCPU
2.	RAM	64 GB
3.	*Disk*	SSD 50 GB
4.	Remote access	SSH

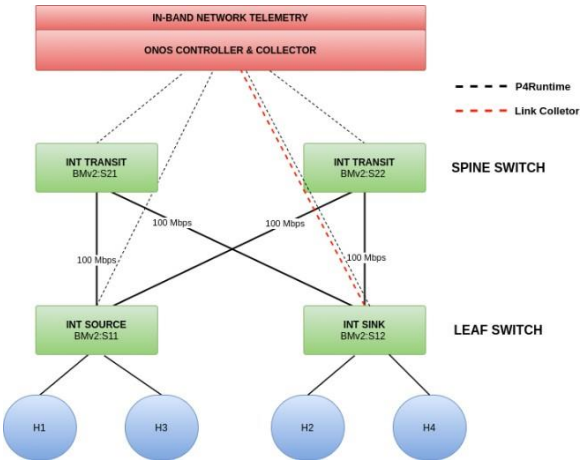
4. Desain Perangkat Lunak

Dalam Desain Perangkat lunak, perangkat menggunakan perangkat lunak sesuai pada Tabel 3.2 untuk menjalankan in-band network telemetry.

Tabel 3.2 Perangkat Lunak yang Digunakan

No.	*Software*	Nama
1.	*Controller*	ONOS v2.1.0
2.	*Network emulator*	*Mininet v2.2.1*
3.	*Packet processor*	P416
4.	*Software Switch (P4-based SDN)*	BMv2 *switch*
5.	*INT Parser*	XDP *Collector*
6.	*INT Collector*	*Prometheus*
7.	*Dashboard visualization*	*Grafana*

5. Implementasi Topologi Sistem

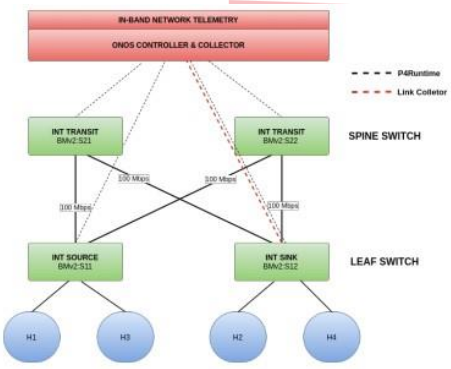

IN-BAND NETWORK TELEMETRY
ONOS CONTROLLER & COLLECTOR
P4Runtime
Link Colletor
INT TRANSIT
BMv2:S21
INT TRANSIT
BMv2:S22
SPINE SWITCH
100 Mbps
100 Mbps
100 Mbps
100 Mbps
INT SOURCE
BMv2:S11
INT SINK
BMv2:S12
LEAF SWITCH
H1
H3
H2
H4

Gambar 3.3 Desain topologi sistem

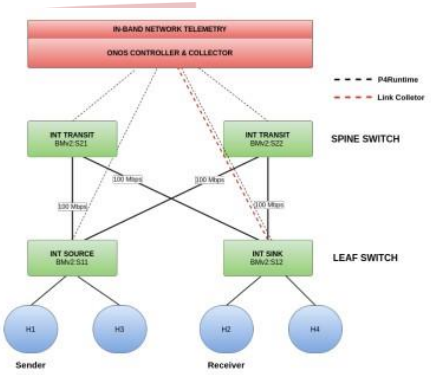
Pada penelitian Tugas Akhir ini, terdapat topologi yang akan digunakan untuk pengujian In-band Network Telemetry menggunakanP4 Language. Pada topologi yang digunakan terdapat 1 Controller, 4 data plane, 1 node collector & visual-ization, dan beberapahost end-point. Gambar topologi tertera sesuai Gambar 3.3 diatas. Topologi yang digunakan untuk pengujian adalah topologi spine-leaf.

1. Skenario Pengujian

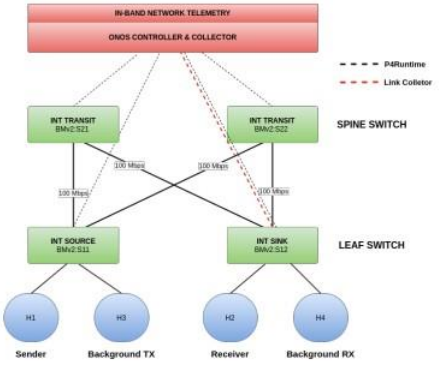
Pengambilan data yang akan dilakukan pada Tugas Akhir ini adalah dengan menguji analysis Protocol Overhead, Protocol Efficiency, Storage overhead idle, storage overhead active, dan storage overhead menggunakan background traffic.



Gambar 3.4 Skenario idle condition



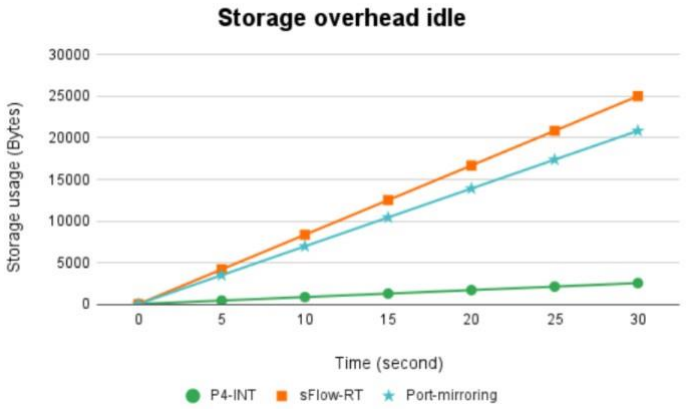
Gambar 3.5 Skenario active condition



***Gambar 3.6** Skenario 3* disk *server 9* disk *cluster*

7. Storage overhead idle

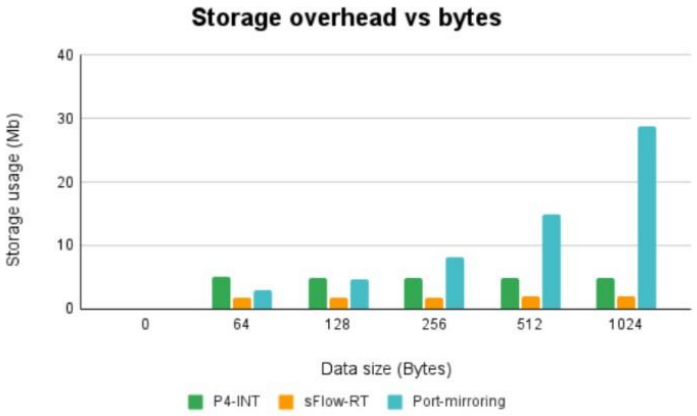
Hasil yang didapat pada saat tidak ada traffic antara client dan server dapat dilihat pada Gambar 3.7, storage overhead yang dihasilkan pada P4-INT selama 30 detik mencapai 2520 bytes (2.52 Kb), pada Port-mirroring mencapai 20859.96 bytes (20.85 Kb), dan pada sFlow-RT storage overhead nya mencapai 25027.65 bytes (25.02 Kb). Storage overhead pada sFlow sangat tinggi karena melakukan streaming telemetry terus menerus walaupun tidak ada trafik karena termasuk noisy protocol. Data streaming nya adalah protokol sflow yang berisi informasi status jaringan. Pada port-mirroring dan P4-INT secara metode hampir sama jika ada traffic, namun port-mirroring melakukan monitoring pada semua paket, sedangkan P4-INT hanya melakukan monitoring pada paket source dan destination tertentu, sehingga storage overhead pada P4-INT lebih kecil karena tidak semua data dimonitoring.



Gambar 3.7 Storage overhead idle

8. Storage Overhead vs bytes

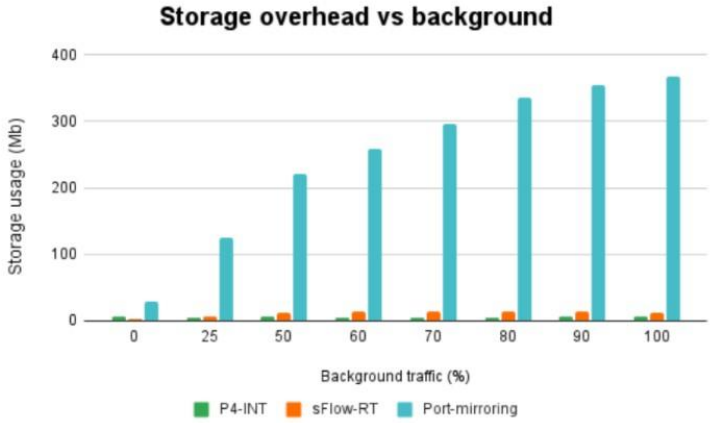
Hasil yang didapat jika ada traffic antara client dan server dapat dilihat pada Gambar 3.8, Paket yang dikirimkan adalah paket user datagram protocol (UDP) menggunakan masing-masing size sebesar 64, 128, 256, 512, dan 1024 bytes dengan rate 1000 packets/second. storage overhead yang dihasilkan pada sFlow-RT rata-rata sekitar 1.82766 Mb, dan P4-INT mencapai 4.90476 Mb, Hal tersebut dikarenakan sFlow-RT menyimpan data sampling setiap rata-rata 0.2 detik sekali, sedangkan P4-INT menyimpan semua informasi Header INT dari setiap paket yang ada dengan rate 1000 Header INT setiap detik, hal tersebut berdampak pada storage overhead lebih besar, namun informasi jaringan pada P4-INT sangat akurat karena realtime data dan bukan sample data. Port-mirroring memiliki storage overhead yang sangat tinggi, karena metode tersebut adalah melakukan copy semua paket yang ada dan disimpan pada sistem monitoring (collector)



Gambar 3.8 *Storage overhead active*

9. Storage overhead with background traffic

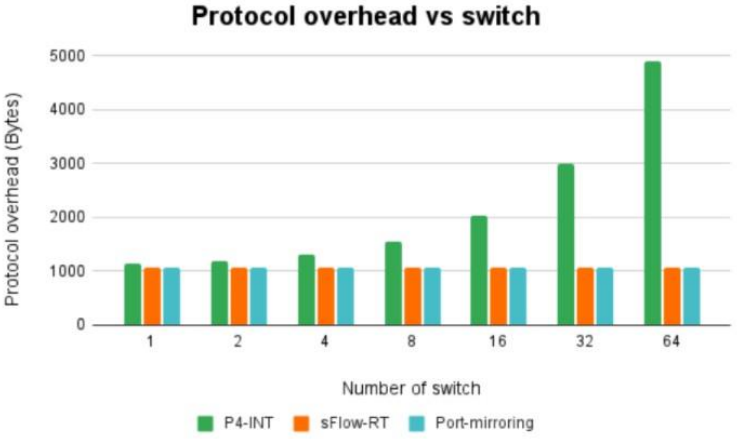
Hasil yang didapat jika ada traffic antara client dan server dapat dilihat pada Gambar 3.9, Paket yang dikirimkan adalah paket user datagram protocol (UDP) menggunakan size sebesar 1024 bytes dengan rate 1000 packets/second. Link bandwidth yang digunakan adalah 100 Mbps, dengan background traffic secara persentase eksponensial 0, 25, 50, 60, 70, 80, 90, dan 100 % dari 100 Mbps link bandwidth. Storage overhead yang paling kecil adalah menggunakan P4-INT rata-rata 4.82 Mb, hal tersebut karena P4-INT memungkinkan untuk monitoring pada specific port, protokol, source, dan destination. Dalam hal ini spesific yang dimonitoring adalah traffic client dan server dengan protokol UDP dan size 1024 Bytes. Kemudian s-Flow bergantung pada berapa banyaknya traffic pada jaringan, sehingga akan berdampak pada jumlah sampling yang semakin tinggi. Sedangkan port-mirroring memiliki storage overhead yang sangat tinggi, karena metode tersebut adalah melakukan copy semua paket yang ada dan disimpan pada sistem monitoring (collector).



Gambar 3.9 Storage overhead with background traffic

1. Protocol overhead

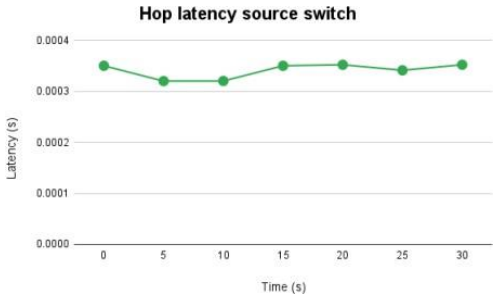
Hasil yang didapat pada Gambar 3.10 menunjukan bahwa semakin banyak switch yang dilewati pada metode INT maka overhead nya akan semakin bertambah, karena metode INT menyisipkan INT Header tambahan ke setiap paket yang lewat, dengan totalsize 60 bytessetiap 1 switch. Pengambilan data di bawah adalah dengan mengirimkan paket UDP sebesar 1024 bytes dan data rate 1 packet/second. Pada sFlow Port-mirrroring hanya menjadi 1066 bytes sampai pada 64 switch karena ada header UDP, IP dan MAC. Sedangkan pada switch 64, metode INT menjadi 4906 bytes karena ada header UDP, IP, MAC, dan INT, sehingga semakin banyak device yang dilewati maka akan berpengaruh terhadap bandwidth usage.



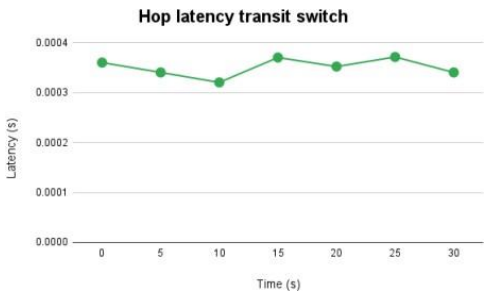
Gambar 3.10 Nilai *Throughput* pada *Random Read*

2. Hop Latency

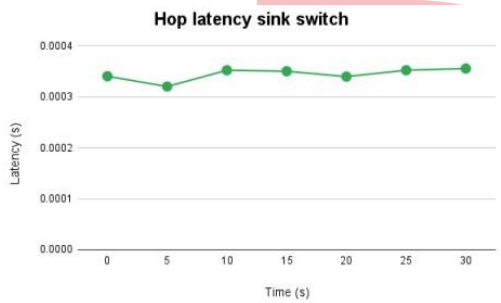
Hasil yang didapatkan pada Gambar 3.11, 3.12, dan 3.13 menunjukan hop-latency, hal ter Latency secara umum adalah waktu yang dibutuhkan dari client menuju server. Kemudian hop latency secara spesifik adalah waktu yang dibutuhkan untuk melewati setiap satu perangkat jaringan. Trafik yang dikirimkan adalah paket UDP port 5001 dengan size sebesar 1024 bytes dan rate 1000 packets/second. Untuk mencapai server, paket harus melewati source switch, transit switch, dan sink switch. Pada jaringan tradisional hanya bisa melihat total latency, namun dengan adanya INT kita bisa melihat latency setiap hop yang dilewati.



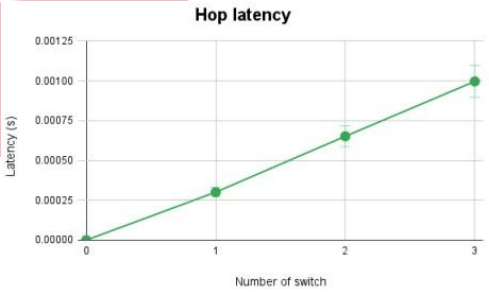
Gambar 3.11 *Hop latency source switch*



Gambar 3.12 *Hop latency transit switch*



Gambar 3.13 *Hop latency sink switch*



Gambar 3.14 *Hop latency summary*

4. Kesimpulan

Berdasarkan pengujian dan analisis In-band Network Telemetry pada infrastruktur jaringan terprogram berbasis P4, dapat diambil beberapa kesimpulan sebagai berikut :

1. Data plane programmability menggunakan metode top-down programming memungkinkan pengembang untuk menambahkan header protocol, mendefinisikan pipelines, dan membuat fitur yang sepenuhnya baru pada infrastruktur jaringan.
2. P4 adalah bahasa pemograman untuk Data plane programmability, P4 merupakan bahasa pemograman tingkat rendah yang dapat menentukan bagaimana switch bekerja dan mendeskripsikan bagaimana perilaku (behavior) pemrosesan paket pada data plane.
3. In-band network telemetry berbasis P4 based programmable infrastructure, memungkinkan untuk melihat end-to-end visualization, karena salah satu fitur yang saat ini bisa digunakan adalah hop-latency pada setiap paket.
4. P4-INT, sFlow, dan port-mirroring merupakan metode monitoring dimana masing-masing memiliki kelebihan dan kekurangan, berikut penjelasanya:
    a. Pada pengujian storage overhead idle, nilai yang dihasilkan oleh P4-INT sangat kecil, karena yang di monitoring adalah langsung paket, kemudian yang paling besar adalah sFlow-RT, karena setiap waktu mengirimkan paket sflow yang berisi informasi status jaringan. Sedangkan port-mirroring melakukan monitoring pada semua protocol.
    b. Pada pengujian storage overhead vs bytes, nilai storage overhead P4-INT dan sFlow-RT stabil, namun sFlow lebih kecil karena melakukan sampling data sflow bukan semua data yang disimpan, sedangkan pada P4-INT menyimpan semua informasi header INT. Storage overhead port-mirroring sangat besar karena, melakukan copy pada paket yang dikirimkan.
    c. Pada pengujian storage overhead dengan background traffic, nilai yang dihasilkan port-mirroring sangat besar dan berbeda jauh dengan P4-INT dan sFlow, karena port-mirroring melakukan copy pada semua paket yang ada. Kemudian nilai sFlow, semakin besar dan banyak traffic.
    d. Pada pengujian protocol overhead, P4-INT menunjukan apabila semakin banyak switch dilewati maka overhead nya semakin besar, hal itu terjadi karena setiap

paket yang lewat switch akan ditambahkan header informasi INT, sedangkan pada sFlow dan port-mirroring tidak ada modifikasi paket.

Referensi

[1] Zhao, Z., Hong, F., and Li, R. ”SDN Based VxLAN Optimization in CloudComputing Networks,” IEEE Access, vol. 5, pp. 23312–23319, 2017.

[2] Abdussalam Ali, and Igor Hawryszkiewycz, ”Cloud as infrastructure for managing complex scalable business networks, privacy perspective,” The Cloud Security Ecosystem, pp. 249-267, 2015.

[3] I. Ivanov, “Comparing the performance of SNMP to Network Telemetry streaming with gRPC / GPB,” Icest2018-40, pp. 175–178, 2018.

[4] John Edwards, ”Streaming telemetry challenges SNMP in large, complex networks”, [Online]. Tersedia di: https://www.networkworld.com/article/3575837/streaming-telemetry-gains- interest-as-snmp-reliance-fades.html, Diakses pada: 20/06/2021

[5] M. Yu, “Network telemetry: Towards a top-down approach,” Comput. Commun. Rev., vol. 49, no. 1, pp. 11–17, 2019, doi: 10.1145/3314212.3314215.

[6] Eriksson J, ”Evolution from OpenFlow to P4/P4Runtime COIN Meeting IRTF 105 2019”, [Online]. Tersedia di : www.noviflow.com, Diakses pada: 14/10/2020

[7] W. L. da Costa Cordeiro, J. A. Marques, and L. P. Gaspary, ”Data Plane Programmability Beyond OpenFlow: Opportunities and Challenges for Networkand Service Operations and Management,” J. Netw. Syst. Manag., vol. 25, no.4, pp. 784–818, Oct. 2017.

[8] Mauro Campanella (GARR), Tomas Martinek (CESNET), Damian Parniewicz (PSNC), Federico Pederzolli (FBK), Damu Ding (FBK) , “In-band Network Telemetry (INT),” [Online]. Tersedia di : www.geant.org.

[9] Internet Engineering Task Force (IETF), ”RFC 7426 - Software Defined Networking (SDN): Layers and Architecture Terminology.” https://tools.ietf.org/html/rfc7426 (accessed Nov. 23, 2020).

[10] Ciena, ”What is Software-Defined Networking (SDN)? - Ciena.” https://www.ciena.com/insights/what-is/What-Is-SDN.html (accessed Nov. 26, 2020).

[11] Open Networking Foundation, ”Software-Defined Networking: The New Norm for Networks [white paper],” ONF White Pap., pp. 1–12, 2012.

[12] Linux Foundation Course, ”Open Source and Software Defined Networking Landscape Book” https://www.linuxfoundation.org (accessed Nov. 26, 2020).

[13] The P4.org API Working Group, ”P4Runtime spesification version 1.3.0”, [online]. Tersedia di : https://p4lang.github.io/p4runtime/spec/main/P4Runtime-Spec.pdf

[14] P4lang github repository, ”behavior model”, [online]. Tersedia di : https://github.com/p4lang/behavioral-model

[15] Jonghwan Hyun, Tu Van Nguyen, James Won-Ki Hong , ”INT Management Architecture: ONOS INT Service and XDP”, (ONF, POSTECH)