PERANCANGAN DAN ANALISIS PERFORMANSI OPEN VSWITCH UNTUK JARINGAN VIRTUAL UNIVERSITAS TELKOM

DESIGN AND PERFORMANCE ANALYSIS OF OPEN VSWITCH FOR VIRTUAL NETWORK TELKOM UNIVERSITY

Muhammad Geo Unggul Putra Kusuma Utomo[1] Dr. Ir. Rendy Munadi, MT.[2] Leanna Vidya Yovita ST., MT.[3]

[1,2,3]Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

[1]mgeoup@gmail.com, [2]rendy_munadi@yahoo.co.id, [3]leanna.vidya@gmail.com

Abstrak

Cloud computing **telah menjadi primadona tren IT pada beberapa tahun terakhir. Baik perusahaan IT skala kecil maupun *data center* skala besar telah mengadopsi teknologi ini. Karena dapat meminimalkan operasi infrastruktur IT, teknologi ini juga terbukti meringankan cost yang harus dikeluarkan perusahaan pada umumnya [1]. Teknologi kunci dalam *cloud computing* adalah virtualisasi. Virtualisasi merupakan sebuah konsep di mana sebuah program atau *operating system* (OS) seakan-akan mempunyai perangkat keras sendiri. Sedangkan *virtual machine* (VM) bertanggung jawab untuk menjalankan OS tersebut seperti menggunakan mesin sesungguhnya.**

Virtualisasi jaringan (*network virtualization*) menghubungkan setiap VM yang dibangun tersebut ke sebuah port switch virtual. Namun, meski virtualisasi membuat implementasi sistem semakin sederhana, perlu ada administrasi jaringan di antara VM, karena *hypervisor* hanya berfungsi sebagai *bridge*. Open vSwitch sebagai alternatif virtual switch yang cukup populer belakangan ini di kalangan pengembang *cloud*, dapat menjadi solusi untuk mengelola trafik antar VM dengan komunikasi dunia luar.

Dari hasil pengujian dapat disimpulkan bahwa fungsi VLAN *isolation* dan QoS *rate-limiting* pada OVS dapat mengisolasi trafik antar VM serta membedakan *bitrate* antar *tenant*. OVS terbukti dapat memberi utilisasi prosesor yang lebih baik dibandingkan dengan tanpa menggunakan OVS. OVS memberi utilisasi prosesor lebih baik sebesar 18.86% pada saat TCP *Send* dan 11.09% pada TCP *Receive*. OVS juga dapat memberi utilisasi prosesor lebih baik dalam pengiriman paket UDP *Send* sebesar 27.27% dan 19.66% pada UDP *Receive*.

Kata Kunci : ***open vswitch, virtual switch , virtual network, virtualization, VLAN isolation, QoS, virtual machine***

Abstract

Cloud computing has become the belle of IT trend in recent years. Both small-scale enterprise IT and data center large scale have adopted this technology. Because it can minimize IT infrastructure operation, this technology also proved to ease the cost to be incurred by the company in general [1]. The key technology in cloud computing is virtualization. Virtualization is a concept in which a program or operating system (OS) as if it possessed the hardware itself. While the virtual machine (VM) is responsible for running the OS is like using a real machine.

Network virtualization connects each VM that is built into a virtual switch port. However, despite the implementation of virtualization make the system more simple, there needs to be between VM network administration, because the hypervisor only serves as a bridge. Open vSwitch virtual switch as an alternative

which is quite popular these days among developers the cloud, it can be a solution for managing inter-VM traffic with the outside world communications.

From the test results it can be concluded that the function of isolation VLAN and QoS rate-limiting with OVS can isolate the inter-VM traffic and differentiate bitrate between tenants. OVS is proven to provide better processor utilization than without using OVS. OVS give a better processor utilization amounted to 18.86% at the time of the TCP Send and Receive 11:09% on TCP. OVS also can provide a better processor utilization in the delivery of UDP packets send by 27.27% and 19.66% on UDP Receive.

Keywords : open vswitch, virtual switch , virtual network, virtualization, VLAN isolation, QoS, virtual machine

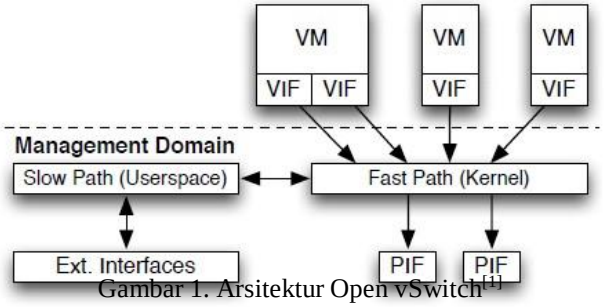
1. Pendahuluan

Teknologi kunci dalam *cloud computing* adalah virtualisasi. Virtualisasi merupakan sebuah konsep di mana sebuah program atau *operating system* (OS) seakan-akan mempunya perangkat keras sendiri. Sedangkan virtual machine (VM) bertanggung jawab untuk menjalankan OS tersebut seperti menggunakan mesin sesungguhnya. Umumnya setiap VM berada pada sebuah host, sehingga mereka berbagi sumber daya fisik termasuk koneksi jaringan. Virtualisasi jaringan (*network virtualization*) menghubungkan setiap VM yang dibangun tersebut ke sebuah port switch virtual. Namun, meski virtualisasi membuat implementasi sistem semakin sederhana, perlu ada administrasi jaringan di antara VM. Oleh karena itu, Open vSwitch sebagai *virtual switch* dapat melengkapi hypervisor yang hanya dapat berperan sebagai bridge antar VM.

2. Dasar Teori

2.1 Open vSwitch Overview[3]

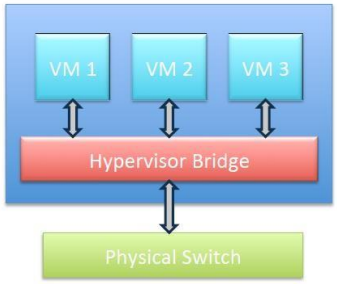
Teknologi ini diinisiasikan oleh perusahaan Nicira, sebuah perusahaan yang fokus dalam virtualisasi jaringan. Dalam artikel Open vSwitch In Your Network (Gross, J., 2013), dijelaskan bahwa Open vSwitch didesain berbasis Apache 2.0. Fitur utama dari Open vSwitch adalah IPsec tunneling, *port bonding* dan *VM trafficking policing*. Dalam *virtual switch*, diklasifikasikan menjadi dua jalur, *fast path* dan *slow path*. Pada fast path terjadi pemrosesan paket-paket data seperti: *packet forwarding*, *trafficking*, dan enkapsulasi paket VLAN. Sedangkan jalur lambat difokuskan pengelolaan dengan interface eksternal.



Gambar 1. Arsitektur Open vSwitch[1]

2.1.1 Virtualisasi

Virtualisasi adalah komponen penting dari *cloud computing*[1]. Pada *cloud environment*, sumber daya (umumnya berbentuk fisik/*hardware*) divirtualisasi sehingga dapat dikelola lebih efisien. Dari sudut pandang komputasi, virtualisasi dapat diasumsikan alih-alih menggunakan perangkat fisik (real environment), perangkat virtual (*virtual environment*) digunakan untuk menjalankan sebuah serangkaian program.Gambar 2.4 mewakili kasus paling dasar dari sebuah virtual environment di mana *virtual resources* digunakan untuk menyediakan layanan kepada masing-masing *user*.



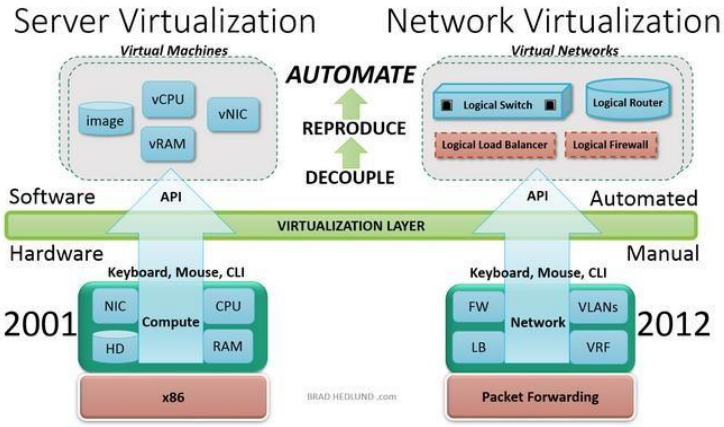
Gambar 2. Ilustrasi sederhana dari Virtualisasi[4]

2.2.1 *Hypervisor*

Hypervisor adalah sebuah perangkat yang memungkinkan untuk membuat dan menjalankan VM di atasnya.[6]

2.2.2 Network Virtualization (Virtualisasi Jaringan)

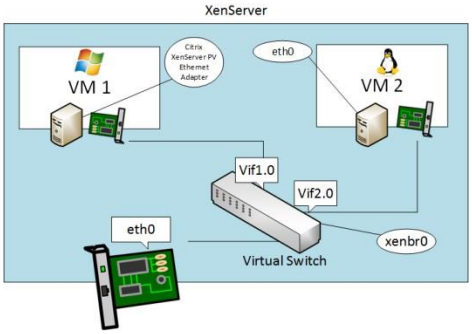
Network virtualization (virtualisasi jaringan) adalah sebuah network environment di mana beberapa jaringan virtual dibangun di atas jaringan fisik sebenarnya. Setiap jaringan virtual pada virtualized environment adalah kumpulan dari virtual node dan virtual link. Pada jaringan tersebut kita dapat membuat dan mengelola beberapa jaringan virtual pada level software tanpa mengganggu satu sama lain. Jaringan inilah yang umumnya dipakai penyedia jasa untuk melayani user.[5]



Gambar 3. Perbedaan virtualisasi jaringan dengan virtualisasi server

2.3 Citrix XenServer

Hypervisor yang digunakan pada penelitian kali ini menggunakan XenServer.XenServer merupakan server *virtualization* platform dari Citrix, untuk mengoptimalkan Windows dan Linux virtual server, dimana semuanya memerlukan kemampuan membuat dan manage sebuah virtual infrastructure.



Gambar 4. *Virtual switch* pada XenServer

Pada dunia Unix, NIC (fisik atau virtual) dinamakan interface (antarmuka). XenServer mengenal interface fisik sedangkan interface virtual sebagai VIF. Saat sebuah OS sedang dijalankan pada VM, VIF beroperasi seperti halnya PIF. Jaringan XenServer pada dasarnya terdiri dari beberapa VIF milik VM yang terhubung ke switch virtual atau bridge. Pada implementasi jaringan sebenarnya switch digunakan untuk mengurangi jumlah collision antar domain di antara segmen jaringan. Demikian halnya pada jaringan XenServer, bridge digunakan untuk menghubungkan antar VM dan jaringan dunia luar.

2.4 Quality of Service Policing

Pada umumnya pelanggan jasa telekomunikasi berlangganan dengan besaran bitrate sesuai kesepakatan kontrak dengan penyedia jasa, sebagai contoh dengan besaran 5,10, atau 20 Mbit. Pada koneksi fiber mampu mengirimkan trafik

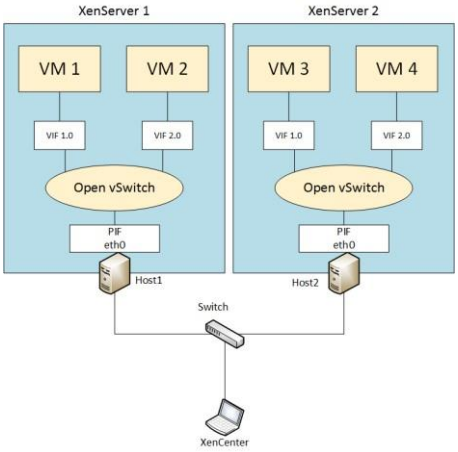
Kontrak yang dimiliki antara penyedia jasa dengan pelanggan umumnya disebut sebagai traffic contract. Besaran bitrate yang dibayarkan pada penyedia jasa (ISP) sering disebut sebagai CIR (Committed Information Rate). Membatasi bitrate koneksi dilakukan dengan teknik policing atau shaping.

3. Perancangan dan Implementasi

Tabel 1 Konfigurasi Perangkat Keras

Host	**Intel® Core™ i5 processor @3.2GHz | 8GB RAM | Harddisk 200 GB | 1000Mbps / Gigabit ethernet.**
Virtual Machine	**Intel® Core™ processor | 2GB RAM | Harddisk 26 GB | 1000Mbps / Gigabit ethernet.**
Remote	**AMD® Core™ processor @2.0Ghz | 4GB RAM | Harddisk 320 GB | 1000Mbps / Gigabit ethernet.**

Tabel 1 merupakan konfigurasi perangkat yang digunakan pada penelitian kali ini.Masing-masing VM menggunakan 1 *core* dari 4 *core* yang tersediap pada CPU *host.*



Gambar 5. Topologi perancangan sistem

Dalam penelitian ini, terdapat beberapa skenario untuk pengujian agar memudahkan analisis kinerja sistem yang telah dibuat.Skenario pertama akan diuji pengujian konektifitas implementasi VLAN *isolation* 802.1Q pada VM untuk kemudian dianalisis utilisasi cpu sebelum dan setelah implementasi.

Kedua, akan diuji fungsi QoS *policing* pada Open vSwitch. Fungsi QoS *policing* sangat berguna ketika cloud administrator ingin menyediakan kualitas yang berbeda untuk beberapa pelanggan.Untuk QoS policing dalam penelitian ini akan diimplementasikan *rate limiting* pada tiap interface VM untuk kemudian dilakukan pengukuran. VM1 dan VM2 akan mengirim paket TCP pada VM3 dengan waktu yang bergantian. Parameter yang diukur adalah *throughput* dari VM1 dan VM2 yang bertindak sebagai pengirim.

Skenario Ketiga, akan diuji performansi jaringan. Performansi dibandingkan antara tanpa OVS (Linux Bridge) dan menggunakan Open vSwitch. Parameter yang diukur adalah *CPU Usage*.

4. Pengujian dan Analisis

Pada bab ini membahas mengenai pengujian dan analisis dari hasil yang telah diimplementasikan. Pengujian dan analisis ini bertujan untuk mengetahui performansi OvS (Open vSwitch) pada host yang menggunakan hypervisor Citrix XenServer 6.5. Seperti yang telah di jelaskan pada bab sebelumnya, Tugas Akhir ini menganalisa performansi dari masing-masing virtual switch OvS tiap host. Beberapa software yang digunakan untuk pengujian dan analisis, yaitu XenCenter 6.5 untuk monitoring host XenServer.

Beberapa poin pengujian dan analisis di antaranya adalah:

- VLAN Isolation

  Pada pengujian ini akan dibuat dua buah tag VID (VLAN Identifier) yang berbeda di antara VM yang telah dibangun. Kemudian akan dilakukan pengujian ping apakah suatu VM dapat berhubungan dengan VM lainnya yang berbeda VLAN tag sehingga terbukti Open vSwitch mampu mengisolasi trafik antar VM menggunakan VLAN 802.1Q. Selain itu kedua VM akan diukur cpu usage sebelum dan setelah implementasi VLAN isolation.

- Quality of Service Policing

  Pada aspek ini akan diuji fitur QoS policing pada Open vSwitch apakah rate limiting tiap interface jaringan dapat diimplementasikan atau tidak. Masing-masing akan diberikan besaran rate limit yang berbeda untuk kemudian ditinjau besaran throughput-nya sebagai salah satu parameter performansi jaringan.

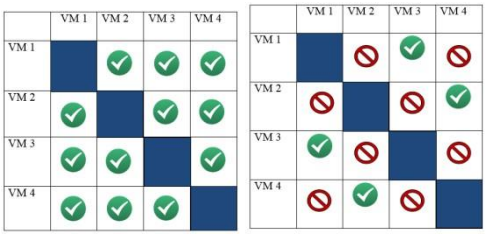
- Performansi non-OVS dan OVS

  Akan dianalisis dua *software switching* antara *Linux bridge* (non-OvS) dan Open vSwitch pada performansi sistem. Parameter yang diukur adalah cpu usage masing-masing VM dan host.

4.1 Pengujian Implementasi VLAN *isolation*

Pada pengujian ini akan dibuat dua buah tag VID (VLAN Identifier) yang berbeda di antara VM yang telah dibangun. Kemudian akan dilakukan pengujian ping apakah suatu VM dapat berhubungan dengan VM lainnya yang berbeda VLAN tag sehingga terbukti Open vSwitch mampu mengisolasi trafik antar VM menggunakan VLAN 802.1Q. Selain itu kedua VM akan diukur cpu usage sebelum dan setelah VLAN *isolation*.

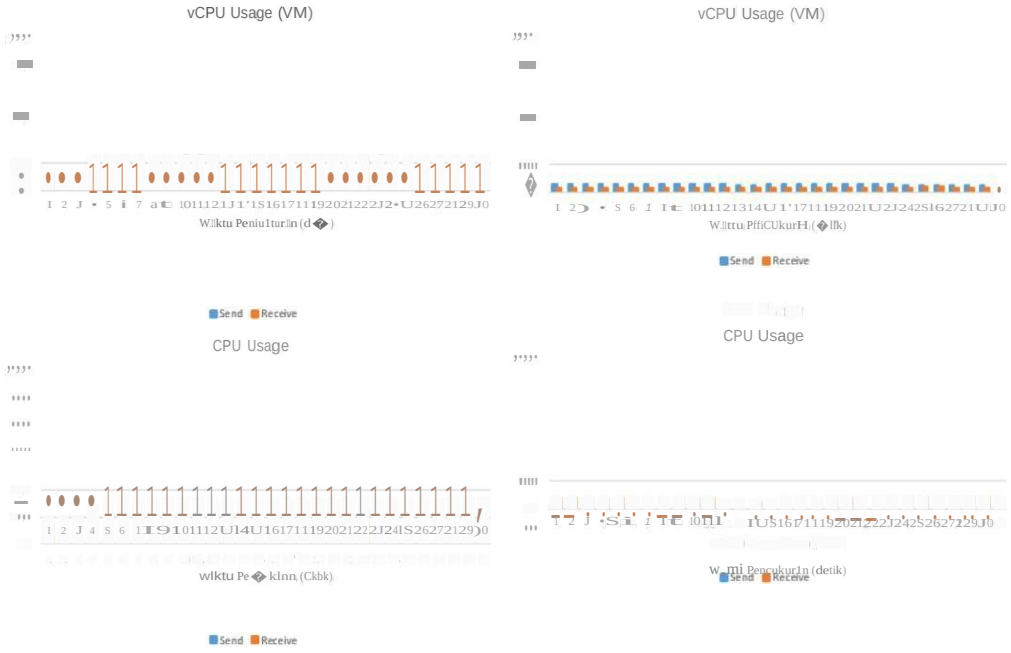
Tabel 2 Pengujian ping sebelum dan setelah VLAN *isolation*



a. Sistematika Pengukuran

Pengukuran CPU usage dilakukan saat diberikan trafik TCP dengan iperf oleh VM1 selama 30 detik. Pada Gambar 4.5 di bawah menggambarkan CPU usage pada masing-masing VM dan host sebelum diimplementasikan VLAN isolation.

b. Hasil Pengukuran



Gambar 6. CPU *Usage* sebelum dan setelah implementasi VLAN *isolation*

c. Analisis Hasil Pengukuran

Implementasi VLAN isolation terbukti dapat menekan utilisasi prosesor fisik (host) hingga 25.34% untuk send dan 17.74% untuk receive. Utilisasi prosesor virtual juga dapat ditekan hingga 7.2% untuk send dan untuk receive. Sistem networking OVS yang bekerja pada layer 2, lebih sederhana dibandingkan linux bridge (default) membuat kinerja beban CPU dalam pengiriman tidak terlalu terbebani.

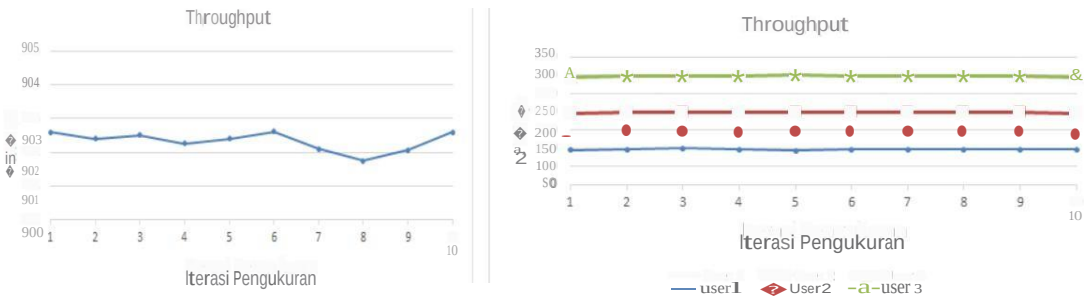
Sebagai catatan pada penelitian kali ini menggunakan prosesor i5-4460 sebagai host. Meski core cpu fisik menggunakan 4 core, tiap VM hanya diberikan 1 core untuk cpu virtual. Perlu diingat proses pengiriman paket TCP sendiri merupakan full duplex di mana sistem mengirim dan menerima paket secara simultan sehingga analisis lebih ditekankan pada penggunaan utilisasi cpu tiap host. Menarik disimak penggunaan VLAN isolation dapat menekan utilisasi cpu karena dengan sistem tagging sistem hanya perlu melakukan mapping pada elemen jaringan yang mempunyai header tag yang sama dengan paket yang dikirim. Ini berimbas pada proses pengiriman yang semakin ringkas dibandingkan sebelum dilakukan VLAN isolation.

4.2 Pengujian implementasi QoS *policing*

a. Sistematika Pengukuran

Dalam sub bab ini akan diuji fungsi QoS policing pada Open vSwitch dengan menggunakan rate limiting (dalam satuan Kb) pada masing-masing interface antara OvS dengan VM1 dan VM2. Pengukuran dilakukan dengan menggunakan iperf sebagai generator dan measurement tool untuk paket UDP.Untuk interface VM1 diberikan rate limit sebesar 102400 Kbit/s (100 mbps) dan untuk interface VM2 diberikan rate limit sebesar 50120 Kbit/s (50 mbps).

b. Hasil Pengukuran



Gambar 7. Grafik *throughput* sebelum dan setelah *rate-limiting*

c. Analisis Hasil Pengukuran

Rata-rata throughput yang diterima sebelum implementasi rate limiting pada interface VM1 adalah 902.32 Mbit/s. Besaran yang cukup bagus dalam koneksi 1 GigabitEthernet karena pada realitanya jarang ditemukan kasus hingga mencapai kapabilitas maksimum perangkat mengingat keterbatasan perangkat. Dapat diasumsikan juga pada jaringan virtual terdapat *trade-off* performansi sehingga tidak sebaik pada jaringan konvensional pada umumnya.

Rata-rata throughput yang diterima setelah implementasi rate limiting pada interface user 1 dan user 2 dan user 3 masing-masing adalah sebesar 96,83 Mbps, 198,15 Mbps dan 297,76 Mbps. Besaran ini cukup memuaskan karena mendekati limit yang diberikan pada interface yaitu masing-masing sebesar 100 Mbps, 200 Mbps dan 300 Mbps. Dengan demikian tenant dengan interface berbeda mendapat QoS yang berbeda pula sehingga implementasi QoS policy dengan teknik *rate-limiting* berhasil dilakukan.

4.3 Pengaruh OVS pada CPU *usage*

a. Sistematika Pengukuran

Dalam sub bab ini akan diuji performansi cpu usage. Performansi dibandingkan antara dua sistem virtual switch yaitu menggunakan *Linux Bridge* (tanpa OvS) dan Open vSwitch untuk kemudian dianalisis pengaruh pada cpu usage masing-masing sistem, baik VM maupun host-nya. Pengukuran menggunakan iperf dengan dua tipe paket, TCP dan UDP. Untuk TCP dan UDP masing-masing dikirimkan paket dengan *window size* dan *buffer size* =64 KB. Pengukuran diambil selama 30 detik dengan diambil besaran rata-rata per 5 detik.

b. Hasil Pengukuran

Tabel 3 Perbandingan CPU *usage* non-OVS dan OVS

VM CPU Usage	TCP		Gain	UDP		Gain
	TCP	TCP OVS		UDP	UDP OVS	
Send	14.55%	5.28%	9.27%	15.06%	5.06%	10%
Receive	24.06%	2.91%	21.15%	26.22%	11.41%	14,81%

Host CPU Usage	TCP		Gain	UDP		Gain
	TCP	TCP OVS		UDP	UDP OVS	
Send	33.6%	6.33%	27.27%	26.9%	8.07%	18.83%
Receive	27.18%	7.52%	19.66%	25.5%	14.41%	11.09%

c. Analisis Hasil Pengukuran

OVS terbukti dapat memberi utilisasi prosesor yang lebih baik sebesar 18.86% pada saat TCP *Send* dan 11.09% pada TCP *Receive*. Terbukti implementasi VLAN *isolation* membantu meningkatkan efisiensi utilisasi prosesor. Ini dikarenakan dalam VLAN *isolation* baik *sender* maupun *receiver* hanya perlu melakukan *network mapping* pada entitas yang hanya mempunyai header tag VLAN yang sama dalam paket .

OVS juga dapat memberi utilisasi prosesor yang lebih baik dalam pengiriman paket UDP *send* sebesar 27.27% dan 19.66% pada UDP *Receive*. Implementasi VLAN *isolation* juga membantu meningkatkan efisiensi utilisasi prosesor. karena baik *sender* maupun *receiver* hanya perlu melakukan network mapping pada entitas yang hanya mempunyai *header tag* VLAN yang sama dalam paket.

5. Kesimpulan

Implementasi Open vSwitch sebagai virtual switch dalam jaringan virtual berhasil dilakukan melalui uji konektifitas. Dengan VLAN *isolation* dapat membantu *provider* untuk memenuhi kebutuhan tenant dalam hal okupansi trafik terlepas dibangun dalam host yang sama atau tidak. Selain itu implementasi QoS *policing* berhasil dilakukan. Rata-rata throughput yang diterima setelah implementasi rate limiting pada interface user 1 dan user 2 dan user 3 masing-masing adalah sebesar 96,83 Mbps, 198,15 Mbps dan 297,76 Mbps. Besaran ini cukup memuaskan karena mendekati limit yang diberikan pada interface yaitu masing-masing sebesar 100 Mbps, 200 Mbps dan 300 Mbps. Dengan demikian tenant dengan interface berbeda mendapat QoS yang berbeda pula sehingga implementasi QoS policy dengan teknik rate-limiting berhasil dilakukan.

Performansi host OVS pada *hypervisor* juga lebih baik jika dilihat dari CPU usage dibandingkan sistem tanpa menggunakan OVS. Nilai CPU *usage hypervisor* dengan menggunakan OVS lebih rendah dibandingkan dengan tanpa menggunakan OVS. Dapat disimpulkan OVS terbukti dapat memberi utilisasi prosesor yang lebih baik sebesar 18.86% pada saat TCP *Send* dan 11.09% pada TCP *Receive*. OVS juga dapat memberi utilisasi prosesor yang lebih baik dalam pengiriman paket UDP *send* sebesar 27.27% dan 19.66% pada UDP *Receive*.

Daftar Pustaka

[1] Forbes, "KPMG's 2014 Cloud Computing Survey: Enterprises Quickly Moving Beyond Cost Reduction To Customer-DrivenResults"[Online], Available:http://www.forbes.com/sites/louiscolumbus/2014/12/26/kpmgs-2014-cloud-computing-survey-enterprises-quickly-moving-beyond-cost-reduction-to-customer-driven-results/ [Accessed 18 November 2014].

[2] H. Tseng, H. Lee, J. Hu, T. Liu, J. Chang, W. Huang," Network virtualization with cloud virtual switch," IEEE 17th International Conference on Parallel and Distributed Systems, 2011.

[3] Gross, J. " Open vSwitch In Your Network". http://www.vmware.com.

[4] R. Barga, J. Bernabeu-Auban, D. Gannon, and C. Poulain , "Cloud Computing Architecture and Application Programming," ACM SIGACT News vol. 40 issue 2, pp. 94-95, June 2009.

[5] Z. He and G. Liang, "Research and evaluation of network virtualization in cloud computing environment," in Networking and Distributed Computing (ICNDC), 2012 Third International Conference on. IEEE,2012, pp. 40–44.

[6] Popek, Gerald J.; Goldberg, Robert P. (1974). "Formal requirements for virtualizable third generation architectures". Communications of the ACM 17 (7): 412–421. doi:10.1145/361011.361073. Retrieved 2015-03-01.

[7] Graziano, Charles (2011). "A performance analysis of Xen and KVM hypervisors for hosting the Xen Worlds Project". Graduate Theses and Dissertations. Iowa State University. Retrieved 2013-03-01.

[8] M. Casado, T. Koponen, R. Ramanathan, and S. Shenker , "Extending Networking into the Virtualization Layer," PRESTO '10 Proceedings of the Workshop on Programmable Routers for Extensible Services of Tomorrow, Article No. 8 ACM New York, NY, USA ©2010

[9] OpenStack. "Open vSwitch Concept" [Online], Available:http://docs.openstack.org/trunk/install-guide/install/apt/content/concepts-neutron.openvswitch.html [Accessed 17 November 2014].

[10] Open vSwitch. "Open vSwitch Concept" [Online], Available:http://docs.openstack.org/trunk/install-guide/install/apt/content/concepts-neutron.openvswitch.html [Accessed 17 November 2014].

[11] Hedlund,B."WhatisNetworkVirtualization?".http://bradhedlund.com/2013/05/28/what-is-network-virtualization/