Running Network of Unikernel Based Middleboxes on Low-Cost Commodity Computers

Nataša Maksić, *Member, IEEE*

***Abstract* - This paper examines the potential of low-cost commodity computers for hosting network of middleboxes and switches, and providing flexible inexpensive platform for packet processing and filtering. The performance of one such solution is reviewed.**

***Index Terms* - middlebox, unikernel, packet processing.**

I. Introduction

ADVANCES in processing power of modern computers have opened new application possibilities. One of those is packet processing. Important functionality of packet network is processing of packet streams in order to perform actions such as packet filtering, NAT, load balancing, intrusion detection and prevention, or function of proxy. Middlebox functionality is often overlooked by observing the network as the set of routers and switches but it is obvious that middleboxes are integral part of modern networks.

Middlebox functionality could be performed by dedicated devices as well as commodity computers. Some of the dedicated devices implement their functionality in specialized hardware chips and provide high throughputs. However, those higher throughputs come at higher price and reduced flexibility, in terms of adaptation to new algorithms and applications. In cases when high throughput is not required, middlebox functionality can also be implemented in software and executed on commodity computers.

Recent advances in virtualization technologies have additionally increased the flexibility of software solutions by enabling deployment and migration of entire virtual operating systems. Virtualization enables execution of multiple isolated operating systems on single server, thus providing the basis for fast and automatic addition, configuration and removal of middleboxes implemented as virtual machines, without the need for any changes in hardware configuration.

Unikernel based virtual machines [1] provide additional advances in this area, by reducing memory and computational requirements in comparison with virtual machines based on multitasking operating systems. Unikernel virtual machines execute only one application. Hence, they do not require multitasking kernel. They contain drivers and other needed system components in libraries which are used by executing applications. Unikernel virtual machines need only drivers for underlying virtualization platform, and with them, they can execute on any computer that has an installation of this virtualization platform. Unikernel virtual machines require memory and processing power needed by application contained in them and have very small additional memory and processing requirements. That enables execution of many unikernel virtual machines even on low-cost commodity computers.

Section 2 of this paper discusses the implementation of middleboxes on commodity computers. Measurement setup is introduced in section 3. Results of measurements with one active middlebox are presented in section 4. Section 5 presents results of measurements for multiple virtual middleboxes. Section 6 concludes the paper.

Nataša Maksić is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: maksicn@etf.rs).

II. Implementation of Middleboxes Using Commodity Computers

Goal of this paper is to evaluate the possibility of using inexpensive, older computers as a platform for running virtual network of middleboxes and bridges. Such network can be dynamically reconfigured and updated without the need for any hardware changes. For that purpose, measurements are performed on PC configuration with processor i7 920. The price of this configuration is low. Since the price of 10Gb/s cards is still high, this PC uses integrated 1Gb/s Ethernet port and one low-cost 1Gb/s network card. The price of 10Gb/s network card alone can be higher than the price of entire older computer.

Virtual middleboxes are implemented as ClickOS [1] virtual machines. Open vSwitch [3] instances are used to connect the network of middleboxes, and to connect middleboxes with network interfaces. The goal of ClickOS project was to create unikernel virtual machine with integrated Click router [4]. ClickOS virtual machine is based on MiniOS virtual machine which is part of Xen [5] virtualization software.

ClickOS is intended to be used as a middlebox virtual machine with low memory and processing requirements. ClickOS project incorporated usage of netmap [6], Vale switch [7] and additional optimizations of Linux kernel in order to provide 10Gb/s processing speeds. Those kernel optimizations will not be used in this paper since they are not

part of official Linux kernel, and some system instabilities with them were detected during the measurements for this paper. Two third-party components that will be used in the measurements are virtualization software Xen and software switch Open vSwitch. Both Xen and Open vSwitch have large user base and continuous development.

The rest of the networking software on the server is part of official Linux kernel. This brings the advantages of stability, easy installation and setup, as well as the possibility of using popular applications on the server. However, official Linux kernel network stack was built for modularity and not for maximal packet throughput, and throughputs that can be achieved are limited. Measurement results for these throughputs are presented in sections IV and V.

DPDK [8] and netmap projects introduce optimizations for faster packet processing. These optimizations will not be used for measurements in this paper since they are not intended for low-cost network cards. DPDK does not support Realtek network cards used in these measurements, and netmap has supported them in earlier versions of Linux kernel, but does not support them anymore.

Various solutions for execution of virtual middleboxes were proposed in literature. Flowstream was a proposal for the middlebox platform which consists of more than one commodity computer [9]. Flowstram proposes configuration in which switches distribute flows across a group of commodity computers, each running a set of virtual machines. Each virtual machine would run instance of Click router. Computer running ClickOS virtual machines whose performance is evaluated in this paper could generally fit into Flowstram architecture, with the difference that virtual machines in Flowstram are not connected within the host computer. Instead, their inputs and outputs are connected to switches external to the computer that hosts virtual machines. Configurable connections between virtual machines within the host server are achieved by using Open vSwitch instances.

Alternative approach to executing middlebox functionality on commodity computers is FlowOS platform [10]. Instead of using virtual machines, FlowOS integrates functionality within the kernel of the host computer. FlowOS paper demonstrates that this approach introduces very small processing overhead and provides good performance. On the other hand, dependence on the kernel of the host machine reduces some flexibility introduced by virtual machines, such as possibility to exchange virtual machine with its complete software, or the possibility of creating virtual networks using virtual bridges and virtual machines.

Recent work on software middlebox performance presents profiling tool and proposal for performance improvement [11]. This paper recognizes performance problems introduced by executing multiple middleboxes on one computer. The profiling of high performance commodity computers has detected two possible improvements. One is reduction of the number of system calls during packet processing. The other is modification of Linux process scheduler to provide longer execution times for middlebox processes. That work is part of the ongoing effort to provide best possible performance for middlebox software executing on high performance general purpose computers. Such computers require significant investment, contrary to low-cost configurations evaluated in this paper.

III. MEASUREMENT SETUP

Measurements were performed on two computers connected with two gigabit Ethernet links. In the following text, these computers will be referred to as middlebox server and test computer. These computers are connected using network port integrated on computer motherboard and one additional low-cost network card. Both integrated interfaces and network cards use Realtek chips.

The test computer generates packet stream and sends it to the middlebox server using the first gigabit Ethernet link. The stream goes through the virtual middleboxes and returns to the test computer via second gigabit Ethernet link. Packet throughput is measured at the reception on the test computer. Measurement configurations are illustrated sections IV and V.

Each measurement was run for 60 seconds, and results for packet rates and throughputs are averaged over this time period.

The test computer has netmap installed and generates packet stream using netmap utility program pkt-gen. Received packet rate is also measured using program pkt-gen. Netmap was used with Realtek driver support for older versions of Linux kernel.

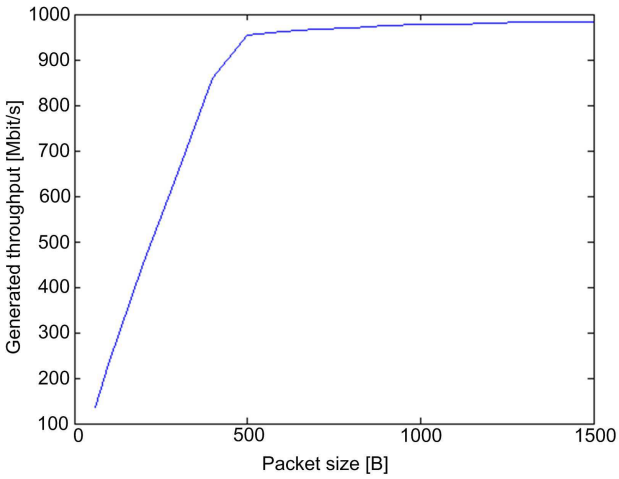

Fig. 1. Throughput generated on test computer

Figure 1 shows generated throughputs on the test computer. For packet sizes below 500 bytes, throughput is lower than maximal link speed of 1Gb/s. This is the consequence of limited packet rates that can be achieved on the test computer. Table 1 presented in Appendix shows that generated packet rates are limited to around 280kpkt/s. Since throughput is proportional to packet rate, the throughputs for small packet sizes are small. However, this does not affect the measurement since measured throughputs are smaller than generated throughputs.

For packet sizes of 500 bytes and above, generated throughput is equal to link throughput. For these packet sizes

generated packet rates are limited by link throughput.

The middlebox server contains a number of ClickOS virtual machines and Open vSwitch instances which connects them. Configuration of ClickOS virtual machine consists of Xen configuration file and configuration file for Click router. Xen configuration file contains name of virtual machine file, the name of virtual machine, and number of processors, amount of RAM memory and configuration of virtual network interfaces. Each virtual machine was configured with one processor and 100MB of RAM.

ClickOS instances are created using Xen program xl, and started using program cosmos, which is the part of ClickOS project. Program xl creates virtual machine based on Xen configuration file. Program Click inside the middlebox will execute according to supplied configuration file. Click enables different kinds of packet processing and forwarding, and in the measurements, it will simply transfer packets from one port of virtual machine to the other port.

IV. Evaluation of Performance with Single Middlebox

First set of measurement is performed with one ClickOS virtual machine on middlebox server. Both the throughput of packet streams generated by the test computer and the throughput of the virtual middlebox were measured in this set of measurements.

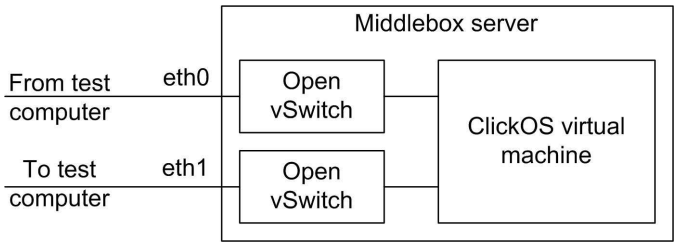

Fig. 2. Measurement configuration with one virtual middlebox

The measurement setup is illustrated in Figure 2. On middlebox server, one Open vSwitch connects port eth0 of the server and first port of ClickOS virtual machine. Another Open vSwitch connects second port of the ClickOS virtual machine, and port eth1 of the server. Packet stream is received from the test computer on port eth0, than it passes through bridges and ClickOS virtual machine, and returns towards the test computer through port eth1 of the server. ClickOS virtual machine has simple configuration in which all incoming packets are forwarded from one port to the other port. On the middlebox server, port eth0 is integrated on the computer motherboard, and port eth1 is low-cost network card. Both eth0 and eth1 are 1Gb/s Ethernet ports.

Table 1 shows throughputs of streams with different packet sizes. Generated packet rate does not change significantly with increase of packet size until the link is saturated for bigger packet sizes. This maximal packet rate is determined by the software and hardware setup of the test computer. However, because of this maximal packet rate, generated throughputs are smaller for small packet sizes. This is not critical, since all received throughputs are smaller than generated throughputs, and measurements can be performed with available packet rates of the test computer.

In practice, packet streams with such mean packet below 500 bytes are not expected. For higher packet sizes, generated throughput achieves link limit of 1Gb/s.

Table 1 shows the packet rates and throughputs with one ClickOS virtual machine. The packet rates decrease with the increase of packet size, which can be attributed to packet copying while traversing through the network drivers, bridges and ClickOS virtual machine on the middlebox server. However, as the packet size increases, the throughput also increases, and reaches around 500Mb/s for typical mean packet sizes. This throughput is sufficient for most local networks, and such configuration can be used for implementation of middleboxes for such networks.

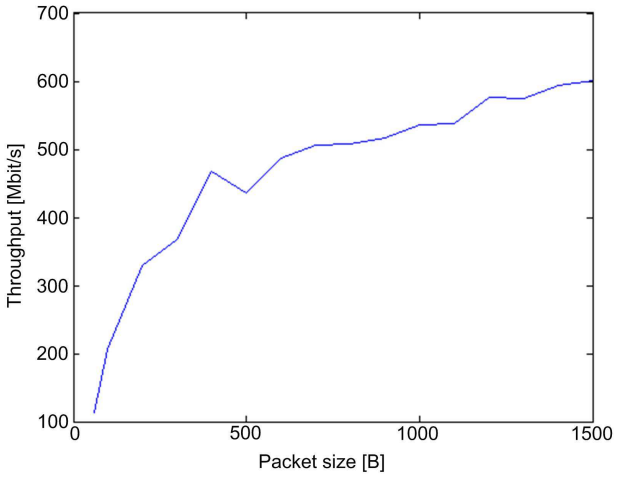

Fig. 3. Received throughput with one virtual middlebox

Figure 3 shows the graph of throughput of one middlebox. Relatively small inconsistencies can be observed in the increase of throughput. General purpose operating systems on PC computers may have variations in packet processing rate due do operation of kernel subsystems such as process scheduling and memory management. These variations may affect the measurements. However, those variations are limited.

V. Evaluation of Multiple Middleboxes

This section presents results of measurements for different number of virtual machines executing on middlebox server. The goal of these measurements is to check how many middleboxes can be run on middlebox server, while keeping satisfactory throughput.

Figure 4 shows the measurement setup for this case. ClickOS virtual machines are connected in linear topology, with one Open vSwitch connecting a pair of ports on neighboring virtual machines.

Measurement results are presented in Table 2 in the Appendix. Table 2 shows how throughput decreases as the number of virtual middleboxes increase.

Figure 5 shows values of throughput measured for packet

size of 800 bytes, which is realistic assumption for mean packet size. The throughput drops significantly for more than four linearly connected middleboxes, and stays approximately constant for four or less middleboxes. This coincides with i7 processor architecture which has four processing cores, and each core supports two processing threads. This indicates that throughput decreases after there are more middleboxes than processor cores, and more than one middlebox needs to be executed on one core.

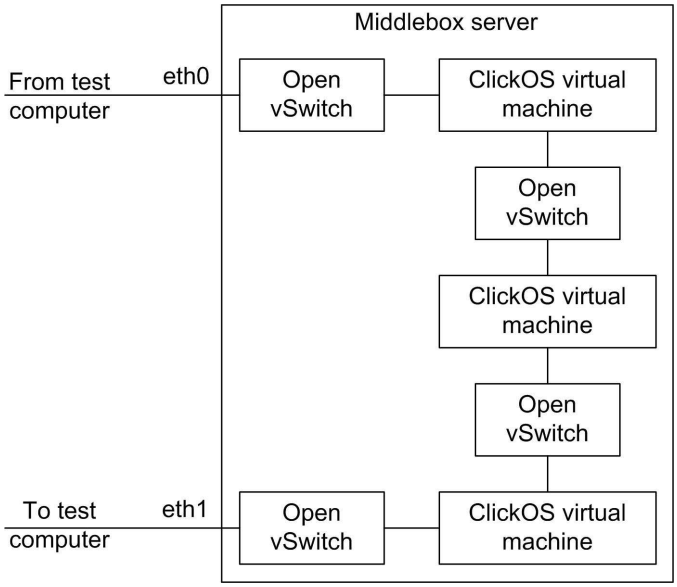

Fig. 4. Measurement configuration with multiple virtual middleboxes

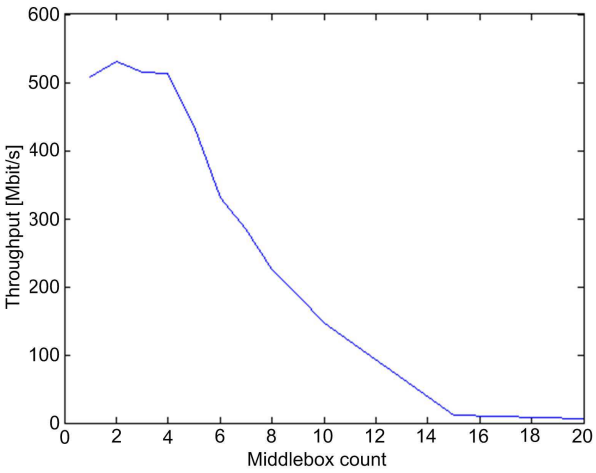

Fig. 5. Received throughput with multiple virtual middleboxes and 800 byte packet size

VI. Conclusion

This paper presents results of evaluation of ClickOS virtual machines with official network stack and Open vSwitch bridging between ports of middleboxes and host computer. Such configuration can be easily installed on some outdated computer without any additional cost. It can provide the network administrators the possibility to evaluate the benefits of the approach with virtual middlebox network. These benefits include completely software-based reconfiguration, without the need for any operation on network cabling or some other hardware operation. Software reconfiguration introduces simpler updates with less down-time and fast configuration of complicated virtual topologies.

The paper shows that one computer without optimized software can execute smaller number of virtual middleboxes and achieve performance which can be sufficient for smaller networks.

This results show throughputs that can be achieved with low-cost computer configurations, and Linux kernel without unofficial optimizations aimed at increase of packet processing speeds. Such setup is stable and affordable and may provide introduction into using platforms with virtual middleboxes.

As packet processing optimizations on the path between Ethernet port and virtual machine mature, and the price of 10Gbit/s network cards fall, the performance of affordable platforms for virtual middleboxes will significantly improve.

Acknowledgment

This work was supported within the project TR-32022 by the Serbian Ministry of Science and Education, and by companies Telekom Srbija and Informatika.

References

[1] A. Madhavapeddy and D. J. Scott, “Unikernels: The Rise of the Virtual Library Operating System,” Communications of the ACM, Vol. 57, No. 1, January 2014

[2] J. Martins, M. Ahmed, C. Raciu, V. Olteanu, M. Honda, R. Bufulco, and F. Huici, “ClickOS and the Art of Network Function Virtualization,” USENIX NSDI ‘14, Seattle, WA, USA, April 2014

[3] B. Pfaff, J. Pettit, T. Koponen, E. Jackson, A. Zhou, J. Rajahalme, J. Gross, A. Wang, J. Stringer, P. Shelar, K. Amidon, and M. Casado, “The Design and Implementation of Open vSwitch,” USENIX NSDI ‘15, Oakland, CA, USA, May 2015

[4] R. Morris, E. Kohler, J. Jannotti, and M.F. Kaashoek, “The Click modular router,” SOSP ’99, Kiavah Island, SC, USA, December 1999.

[5] P. Bargam, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, A. Warfield, “Xen and the Art of Virtualization,” SOSP ’03, Boldon Landing, NY, USA, October 2003

[6] L. Rizzo, “netmap: a novel framework for fast packet I/O,” Usenix ATC'12, Boston, MA, USA, June 2012

[7] L. Rizzo, G. Lettieri, “VALE, a Switched Ethernet for Virtual Machines,” CoNEXT '12, Nice, France, December 2012

[8] Intel Data Plane Devopment Kit (Intel DPDK), http://www.intel.com/content/www/us/en/communications/data-plane-development-kit.html

[9] A. Greenhalgh, M. Handley, M. Hoerdt, F. Huici, L. Mathy, and P. Papadimitrou, “Flow Processing and the Rise of Commodity Network Hardware,” ACM SIGCOMM Computer Communication Review, Vol. 39, No. 2, April 2009

[10] A. Alim, M. Bezahaf, and L. Mathy, “FlowOS: A Programmable Platform for Commodity Hardware Middleboxes,” CFI ’13, Being, China, June 2013

[11] G. P. Katsikas, G. Q. Maquire Jr., Dejan Kostić, “Profiling and accelerating commodity NFV service chains with SCC,” The Journal of Systems and Software 127, 12-27, January 2017.

APPENDIX: TABLE I
Performance of single virtual middlebox

Packet Size [B]	Generated Packet Rate [kpkt/s]	Generated Throughput [Mb/s]	Received Packet Rate [kpkt/s]	Received Throughput [Mb/s]
60	284.99	136.80 (raw 191.51)	237.52	114.01
100	286.82	229.46 (raw 284.53)	256.63	205.30
200	281.48	450.36 (raw 504.41)	205.72	329.15
300	271.19	650.85 (raw 702.92)	152.84	366.82
400	268.75	860.00 (raw 911.60)	146.26	468.03
500	238.53	954.11 (raw 999.91)	108.82	435.28
600	200.30	961.46 (raw 999.91)	101.33	486.38
700	172.64	966.76 (raw 999.91)	90.47	506.63
800	151.69	970.79 (raw 999.92)	79.47	508.61
900	135.26	973.88 (raw 999.85)	71.70	516.24
1000	122.06	976.48 (raw 999.92)	66.90	535.20
1100	111.20	978.57 (raw 999.92)	61.12	537.86
1200	102.12	980.31 (raw 999.92)	59.98	575.81
1300	94.40	981.79 (raw 999.92)	55.29	575.02
1400	87.77	983.06 (raw 999.92)	53.10	594.72
1500	82.01	984.17 (raw 999.92)	50.13	601.56

APPENDIX: TABLE II
Throughput of multiple virtual middleboxes [Mb/s]

VM count / Packet size	1	2	3	4	5	6	7	8	9	10	15	20
60	114.01	98.90	67.91	55.23	43.61	30.57	25.05	19.65	10.30	11.05	0.82	0.5
100	205.30	156.89	127.68	91.76	71.79	57.62	48.84	45.68	27.25	18.22	1.37	0.87
200	329.15	263.46	200.90	154.94	108.05	103.63	64.53	60.62	52.08	43.52	2.59	1.71
300	366.82	369.41	295.49	205.08	169.68	148.42	128.59	112.30	59.50	49.66	3.89	2.52
400	468.03	426.88	260.96	280.26	213.57	188.86	121.18	93.50	22.24	23.52	4.93	3.42
500	435.28	476.48	346.48	278.08	266.28	231.52	202.24	135.2	26.36	33.84	6.28	4.12
600	486.38	508.18	464.69	353.66	301.63	281.81	162.43	207.70	83.66	92.59	7.78	4.99
700	506.63	525.0	506.30	478.07	387.07	323.23	232.18	173.32	139.55	104.38	10.58	5.77
800	508.61	530.88	515.39	512.51	436.8	331.33	284.28	226.24	186.05	145.98	11.33	6.51
900	516.24	527.54	525.67	514.08	425.66	389.88	289.37	256.25	211.39	131.11	13.39	7.49
1000	535.20	543.36	544.8	540.8	478.08	428.0	353.68	245.44	235.2	148.72	15.76	8.0
1100	537.86	547.62	549.65	544.46	500.02	400.84	334.84	250.36	255.90	262.24	16.90	8.98
1200	575.81	564.38	566.78	566.11	551.62	510.91	413.57	294.62	283.2	170.59	15.94	10.27
1300	575.02	575.02	574.18	574.81	561.50	500.76	357.24	283.92	293.49	235.87	23.4	11.44
1400	594.72	593.71	588.56	586.88	583.52	541.74	386.85	336.90	323.34	196.0	21.95	11.87
1500	601.56	602.52	598.92	599.16	518.76	579.96	377.28	333.24	348.84	207.36	22.56	12.96