

Experimental Evaluation of Desktop Operating Systems Networking Performance

Original Scientific Paper

Krešimir Vdovjak

J.J. Strossmayer University of Osijek,
Faculty of Electrical Engineering, Computer Science and Information Technology
Kneza Trpimira 2b, Osijek, Croatia
kresimir.vdovjak@ferit.hr

Josip Balen

J. J. Strossmayer University of Osijek,
Faculty of Electrical Engineering, Computer Science and Information Technology
Kneza Trpimira 2b, Osijek, Croatia
josip.balen@ferit.hr

Krešimir Nenadić

J. J. Strossmayer University of Osijek,
Faculty of Electrical Engineering, Computer Science and Information Technology
Kneza Trpimira 2b, Osijek, Croatia
kresimir.nenadic@ferit.hr

Abstract – *The rapid advancement of network, communication and Internet technology resulted with always-on, always-connected, device-independent and remote online working, business, education and entertainment environment. Consequently, users are searching for solutions and technologies that enable fast and reliable wide area network connection and the typical solution is through using personal computers connected with ethernet cable to network equipment and infrastructure that supports gigabit ethernet connection. Besides the complex network infrastructure that can influence performance, the bottleneck can also be caused by insufficient hardware, operating system and software resources on clients' machines. Therefore, in this paper a networking performance evaluation of three globally most common and most used versions of Windows operating systems; namely Windows 7TM, Windows 8.1TM and Windows 10TM, on two identical computer systems, is conducted. Networking performance measurements are performed with three different benchmarks: namely iPerf, D-ITG and NetStress. Performance evaluation results showed that a newer versions of an operating system bring certain networking performance improvements but by sacrificing other performances.*

Keywords – *benchmark, network, operating systems, performance evaluation, Windows*

1. INTRODUCTION

In today's age of technology, a vast majority of computers and users are interconnected through wide area network (WAN) and a great number of services and applications require a high network quality of service (QoS) with low network latency and guaranteed up-time. Personal computers are using various operating systems designed to manage available resources efficiently. Since users are constantly present on the Internet one of the main requests is having a high bandwidth WAN access, and therefore network resources management became a crucial operating system component. More recently users tend to use wireless WAN connection, but security, speed and reliability can only be guaranteed by using a wired ethernet connection that is mainly used for business purposes. Today's local area networks (LAN)

typically support gigabit ethernet connection with most common bandwidths of 100 Mbps, 1 Gbps and 10 Gbps, but the WAN connection bandwidth is lower and depends on the territory, country, or local accessibility and wider network infrastructure quality. However, malfunctioning client hardware, software or operating system resources can dramatically affect networking performance, and therefore it is necessary to analyze available client resources in detail in order to minimize all possible failures and bottlenecks.

Network performance evaluation in a field of computer and operating systems is a challenging problem due to its complexity, dependency on numerous system and user parameters, as well as long implementation and duration. Moreover, a networking performance depends on numerous internal and external factors, such

as Internet service provider (ISP), quality of installation and equipment (connected routers, switches and servers), efficiency of different routing strategies and fault-tolerance schemes, as well as on operating system network resource management [1]. Performance degradation caused by operating system is often connected to underlying hardware, available firmware and drivers and can affect clients connected to any type of network, private or public, wireless or LAN. Furthermore, a testing environment can dramatically influence the performance since it is hard to achieve unique and repeatable working conditions. Therefore, performance evaluation of computer and operating systems is an interesting field but due to its complexity, limited work has been conducted by scientists, as well professionals who are publishing their research online [2] and [3].

Modern desktop operating systems are designed to accommodate multiple users, are multitasking and capable of handling large numbers of different processes at the same time. According to [4], Windows operating systems are the most common and most used desktop operating systems where Windows 10 holds 52.68%, Windows 7 29.34% and Windows 8.1 3.74% of the total market share. Therefore, those three latest versions of Windows operating system are used in our experiments. Microsoft has been dominating in the field of personal computer operating systems and Windows 7 was the major operating system until 2012 when Windows 8 was developed as an operating system for all devices. However, this logic was not successful, so Windows 10 was developed with different modes for tablets and personal computers which produced much greater success. Due to different time periods of development, system designing and integrating goals, as well as an architecture, purpose and application requirements, various versions of Windows operating systems have different characteristic, running requirements and performances. Due to different users and applications requests, it is necessary to choose an appropriate operating system to meet the performance requirements. However, it is unknown whether the latest operating system brings improved performance, and therefore it is necessary to study, test and compare operating systems and based on the results choose the best one. With the increased usage of WAN, Internet and computer networks, the networking performance analysis has become crucial.

This paper continues our work started in [5] where a networking performance of three versions of Windows operating systems; namely Windows XP, Windows Vista and Windows 7 were evaluated. However, two out of three versions of Windows operating system are outdated, all versions are based on older 32-bit architecture and the networking performance was primarily measured and evaluated based on the result from only one measurement tool. This paper presents an experimental research regarding networking performance evaluation of three latest versions of Windows operating system,

namely Windows 7, Windows 8.1 and Windows 10. Performance measurements were conducted by using three different measurement tools based on accurate, reliable and repeatable performance measurement methodology. Performance evaluation results show that Windows 7 had the lowest CPU usage during intense network traffic but with unstable performances. The opposite of that, Windows 8.1 is the most stable operating system during heavy network load but for the stability networking performance is sacrificed, resulting in increased delays and CPU usage. For Windows 10 the lowest possible networking delay was one of the major development goals, but it was succeeded with an increased CPU usage. Other networking performance metrics did not show any improvements when compared to other two tested operating systems.

The remainder of this paper is organized as follows. Section 2 presents related work, while Section 3 presents performance measurement tools, called benchmarks. Performance measurement setup, methodology and impact on results is detailed in Section 4. In Section 5 performance measurement results are presented and discussed. Section 6 concludes the paper.

2. RELATED WORK

Operating system networking performance evaluation is often complicated task due to complex network system structure, various available hardware resources, networking and usage scenarios, as well as numerous constructional, internal and external factors that influence the operating system performance. Therefore, very limited research in this field has been conducted resulting in low number of available related work and the latest is described below.

During the previous work conducted by the authors several different aspects of operating systems performance on various hardware system and in various environments, were studied. In [6], an influence of three different versions of Windows host operating system, namely Windows XP®, Windows Vista™ and Windows 7™ on virtual Windows Vista operating system performance was studied. In contrast to this work, in [7] performance evaluation of three different virtual Windows operating systems, namely Windows 7, Windows 8.1 and Windows 10 was conducted on a host computer system based on Linux Ubuntu operating system. In [8] a comprehensive performance evaluation based on CPU scheduling, memory management, graphic subsystem management, hard disk drive management and networking performance in two different environments of the three different versions of Windows operating system, namely Windows XP, Windows Vista and Windows 7, was performed. In [9], a performance evaluation of three popular virtual private server hosts based on Linux Debian operating system was conducted.

This paper continues our work described in [5] where networking performance of three versions of Windows

operating systems; namely Windows XP, Windows Vista and Windows 7, were evaluated. The networking performance was measured and evaluated on 32-bit operating systems with only one measurement tool Distributed Internet Traffic Generator (D-ITG) [10] in terms of network delay, throughput, jitter and CPU usage. Based on the performance evaluation results, it can be concluded that Windows Vista and Windows 7 improve network traffic of smaller packet sizes while Windows XP still showed the best networking traffic performance while transferring bigger packet sizes. Therefore, our motivation was to continue our work in the operating system networking performance evaluation field and conduct new research regarding improvements in the network management subsystem within latest versions of Windows operating systems. In this paper, our previous work is significantly expanded by conducting much more detailed research, performing a comprehensive performance measurement and evaluation with developed performance measurement methodology and by using three different performance measurement tools/benchmarks. In order to get more reliable and comparable results, performance measurement methodology is based on the process and experiments that is reliable, repeatable and can generate the same results or even be applied for the evaluation of similar systems and performances.

In [11] full performance evaluation of Linux server operating system was conducted based on six following metrics and performance: processor and processes, basic mathematical operations, context switching, communication latencies and bandwidth, file and system latencies, and memory latencies. Lmbench [12] was used to measure communication latencies of Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) protocols, AF_UNIX sock stream latency and a pipe latency. Performance measurement results showed huge differences among four tested distribution of Linux operating systems. In [13] similar measurement setup as in this paper was used. Two identical desktop computers were connected with ethernet cable through two identical Cisco routers. However, one computer had Windows 7 operating system while the other Windows Server 2012 operating system and the various networking performance scenarios were measured between them. D-ITG was used for performance measurements of Internet Protocol version 6 (IPv6) and version 4 (IPv4), followed by 4to6 transition mechanism configured with PPTP and 6to4 configured with IPsec without any VPN protocol. A comprehensive performance evaluation was conducted and obtained results showed that the 6to4 transition mechanism is reasonably faster and a more reliable protocol. However, in specific cases 4to6 such as VPN protocol configuration with IPsec as well as configured without PPTP showed better performance. In [14] performance analysis of three desktop operating systems, namely, Windows XP, Windows 7 and Linux Fedora 12, was conducted. Experiments were conducted through IEEE 802.11n peer-to-peer (P2) wireless LAN (WLAN)

network without implemented security for TCP and UDP, IPv6 and IPv4 protocols. The experimental setup is similar as in described paper since two identical desktop computers were used, and in each experiment different operating system was installed and tested. However, the difference is that computers were connected wirelessly through Cisco Linksys WAP4410N 802.11n access point. Netperf tool was used for the measurement of networking performance in terms of throughput and round-trip time. Performance evaluation results showed that Windows XP has the lowest throughput and highest delay while Linux Fedora provides higher throughput and lower delay than Windows 7. However, Windows 7 showed improved IPv6 performance when compared to Windows XP.

3. PERFORMANCE MEASUREMENT TOOLS - BENCHMARKS

Performance measurement tools specifically developed for measuring performance of a complete computer system or for measuring performance of specific component are called benchmark applications - benchmarks. In this paper, the emphasis is put on operating systems networking performance, and therefore special benchmark applications are required which enable network traffic to be generated and analyzed in terms of various networking performance metrics. Three different benchmark applications for measuring networking performance, namely iPerf, D-ITG and NetStress, were used in the paper and they are described below.

3.1. IPERF V3.1.3

iPerf [15] and [16] is a benchmark tool that enables measurements of various network parameters related to buffers, timing and protocols which include TCP, UDP, SCTP (IPv4 and IPv6). Major metrics includes bandwidth, jitter, packet loss and CPU usage. The main features of iPerf version 3 and above are the following:

- TCP and SCTP (bandwidth measuring, reporting MSU/MTU size),
- UDP (packet loss, jitter, delay, bandwidth streams, multicast),
- Cross-platform: Windows, MacOS X, Linux, FreeBSD, OpenBSD, NetBSD, VxWorks, Solaris, Android,
- Multiple simultaneous connections for client and server,
- Server handles multiple connections,
- Run time can be specified time,
- Measure periodic, intermediate bandwidth, loss reports and jitter,
- Run the server as a daemon,
- Set target bandwidth for UDP and TCP,
- Output in JSON format.

3.2. D-ITG V2.8.1

D-ITG is an open-source network benchmark tool [10] capable to reproduce a workload of Internet applications by generating network traffic at packet level at following layers: network, transport and application layer. For reproducing an application-level protocol behavior, two stochastic models are used: Inter Departure Time (IDT) and Packet Size (PS). At the transport layer the following protocols are supported: TCP, UDP, Datagram Congestion Control Protocol (DCCP) and ICMP (Internet Control Message Protocol). Furthermore, it enables measurement of the most common networking performance metrics such as delay, throughput, packet loss and jitter. Both internet protocols IPv4 and IPv6 are supported. The main features of D-ITG are the following:

- Customizable flow-level properties (duration, start delay, number of packets, number of Kbytes),
- Supported Layer-3 features (IPv4 and IPv6 protocols),
- Supported Layer-4 features (TCP, UDP, ICMP, DCCP, SCTP protocols),
- Supported Layer-7 features (redefined stochastic PS and IDT profiles, payload content, stochastic processes supported for both PS and IDT),
- Packet-level QoS metrics (bitrate, one way delay, round trip time, jitter, bitrate, packet loss),
- Cross-platform: Linux (Ubuntu, Debian, Fedora, CentOS, OpenWRT), Windows (XP, Vista, 7, 8.1, 10), OSX (Leopard), FreeBSD.

3.3. NETSTRESS V2.0.9686

NetStress [17] is capable to measure performance of both wired and wireless networks.

It is capable to produce high performance experiments by filling buffers in memory with data and then by using the TCP or UDP protocols repeatedly transmitting the data. Major measurement metrics is throughput that can be compared with the theoretical bandwidth between the transmitter and receiver or with a previous measurement result. By throughput comparison it is possible to identify variations and the possible network issues such as significant amount of other traffic present in a network or overloaded network equipment, that can cause loss of packets and communication errors. The main features of NetStress are the following:

- Supports Windows XP, Vista, 7, 8.1 and 10 operating systems,
- TCP and UDP data transfers are supported with variable segment sizes,
- It can be used as a server since there is only one instance of the application,

- Multiple data streams are supported,
- Multiple network adapters are supported,
- Supports Variable Maximum Transmission Unit,
- Two operating modes: uplink and downlink,
- Auto node discovery.

4. EXPERIMENTAL SETUP

4.1. PERFORMANCE MEASUREMENT SETUP

Networking performance measurement was performed on two identical personal computers, laptops with identical hardware and software resources and connected with 1 Gbps ethernet cable. Three latest versions of Windows operating system were used in the experiments since according to [4] they are most widely used operating system on personal computers. In order to have as much as possible similar operating environment three identical editions of 64-bit Windows operating systems were used, namely, Windows 7 Professional, Windows 8.1 Professional and Windows 10 Professional edition. Professional editions of Windows operating systems were selected since they are the most equivalent among three tested versions and also add additional features that are oriented towards business environments and advanced and professional users. Firstly, each operating system was installed with default settings and updated afterwards with the latest available updates. This is one of the crucial steps towards ensuring similar experimental environments since updates include enhancements that improve performance, security, and stability of operating system. Furthermore, the next crucial step is installing the newest device drivers since there are constantly new version which remove errors and issues and mostly improve performance of hardware components. Hardware and considered operating systems used in the experiments are shown in Table 1.

As mentioned before, networking performance evaluation was done on Windows operating systems in their default network configurations. Additional configuration of operating systems can also have huge impact on network performance but it is not considered in scope of this paper. Some of possible improvements for network subsystem are [18]:

- Enabling offload features,
- Enabling receive-side scaling,
- Increase network adapter resources,
- Reduce system management interrupts (SMI) to minimum,
- Performance tuning TCP (receive window auto-tuning),
- Using Windows Filtering Platform (WFP) to create packet processing filters,
- Using performance tools for network workloads.

Table 1. Hardware and considered operating systems

Hardware Components	
Motherboard	Hewlett Packard 15-p107nz
CPU	Intel Pentium N3540 2.16 GHz, 4 cores, 2 MB cache
RAM	8 GB 1333 MHz DDR3L SDRAM (1 x 8 GB)
HDD	1 TB 5400 rpm Serial ATA
GPU	Intel HD Graphics
NIC	10/100 BASE-T Ethernet LAN (Integrated)
Operating systems	
Microsoft Windows 7	Version: Professional SP1 64-bit
Microsoft Windows 8.1	Version: Professional 64-bit
Microsoft Windows 10	Version: Professional 64-bit

The benchmark applications were configured according to the following specifications detailed below. All other parameter that are not mentioned below were left in default settings:

Iperf3:

- The duration of each measurement was 10 seconds and it was repeated 5 times,
- In order to gain a wider range of results, measurements were conducted for 9 different packet sizes, ranging from 32 bytes to 896 bytes (with step of 128 bytes after 64 bytes),
- Performance of TCP and UDP protocols were measured,
- First laptop was configured as a server (receiver) and the other laptop was configured as a client (sender).

D-ITG:

- The duration of each measurement was 10 seconds and it was repeated 5 times,
- In order to gain a wider range of results, measurements were conducted for 9 different packet sizes, ranging from 32 bytes to 896 bytes (with step of 128 bytes after 64 bytes package)
- Inter-departure time was set as constant and number of packets was uniformly distributed between 30000 packets/s to 300000 packets/s,
- Performance of TCP and UDP protocols were measured,
- ITGSend component was used for sending packets to another laptop where ITGRecv component was used for receiving packets,
- ITGLog component was used for collecting all the measurement results.

NetStress:

- The duration of each measurement was 10 seconds and it was repeated 5 times

- In order to gain a wider range of results, measurements were conducted for 7 different packet sizes, ranging from 16 KB to 64 KB (with step of 8 KB),
- Performance of UDP protocol were measured,
- First laptop was configured as a server (receiver) and the other laptop was configured as a client (sender),
- Number of packets per second was set to 1024.

4.2. PERFORMANCE MEASUREMENT METHODOLOGY

Performance measurement methodology is oriented towards ensuring consistency, accuracy, repeatability and reliability of the conducted experiments and obtained measurement results. Performance measurement process shown in Fig. 1 was defined in order to ensure the same experimental conditions for all operating systems during performance measurements as well as to avoid possible measurement errors. The first three steps must be repeated for each tested operating system, while the last four steps must be repeated for each benchmark application, separately. Since other third-party applications and files can influence operating system performance, only benchmark applications should be installed, all other user files must be removed from the disk drives and during experiments user activity is not allowed. To improve the accuracy of the results each measurement was repeated five sequential times in the same working conditions. The final result for every metrics parameter is calculated as the arithmetic mean of those five repetitions, as shown in (1).

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \rightarrow N = 5 \quad (1)$$

$$x_{imin} \leq \bar{x} \leq x_{imax}$$

Furthermore, the final performance measurement results are reported with only significant digits, rounded on two decimal places. Comparison of performance measurement results was conducted by using Win-

dows 7 results as referent values and by calculating the percentage difference of other two operating systems with respect to Windows 7 values, as shown in (2).

$$Diff\% = \frac{Win8.1_or_Win10_val - Win7_val}{Win7_value} \times 100\% \quad (2)$$

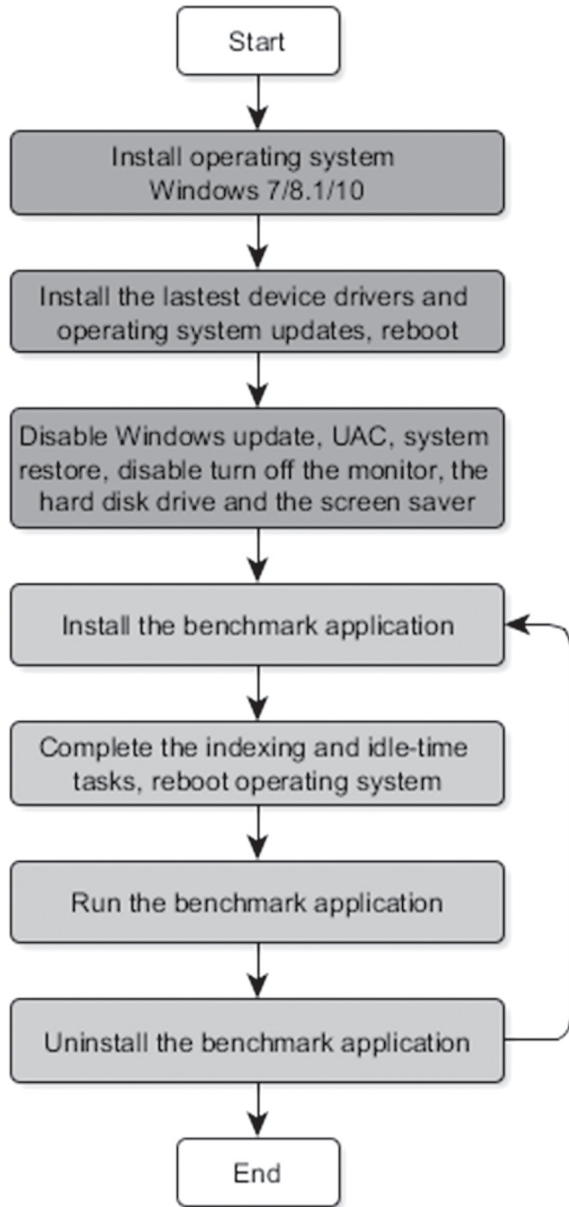


Fig. 1. Performance measurement process

Same performance measurement methodology can be used to evaluate network performance of other, non-Windows, operating systems. However, when comparing results of Windows and non-Windows operating systems it is necessary to use same versions of benchmark applications to ensure consistency. Different version of same benchmark application can use different algorithm or methodology to evaluate same parameter and thus, results obtained with different versions of same benchmark application cannot be compared.

Performance evaluation was performed by comparing performance measurement results of each test in every benchmark application. The results were evaluated through four main metrics:

- Bandwidth - represents the average bitrate and is measured in Kbps (kilobit per second). Higher bandwidth means better performance,
- Delay - calculated as the average of differences between receiving and sending times of packets. Unit is ms (millisecond) and lower delay means better performance,
- Jitter - is defined as the end-to-end transmission delay difference between selected packets in the same packets stream. Unit is ms and lower jitter means better performance [19],
- CPU usage - is the primary indicator of processor activity and is measured as the average percentage of busy time observed during the networking performance measurement with RTT meter. Lower CPU usage means that more CPU resources are available to other tasks in a computer system.

4.3. HARDWARE IMPACT ON PERFORMANCE MEASUREMENT RESULTS

Computer systems are very complex and consisted of various software and hardware components that are interconnected and have to work efficiently together in order to deliver satisfying performance. Complexity is rising with technology advancement and each new version of an operating system consists of numerous new features and capabilities when compared to the previous ones. However, although the complexity is growing, it is expected that each new operating system will deliver improved performances. This trend is also obvious in computer hardware development [20]. Beside the software itself, hardware has also a huge impact on networking performance. Main hardware components that can cause degradation in networking performance are explained below [21]:

- CPU - insufficient processing power can increase latency if a computer CPU is unable to keep up with network traffic and even a single node with an inadequate processing resources can impact the entire network and degrade performance due to the high degree of interdependence between the nodes in the modern network [22],
- RAM - memory is a computing resource that has requirements in both the data and control planes. The performance of the entire network can degrade when control plane processes fail, as is the case when routing convergence requires additional memory,
- Disk Drive - typically used as storage area. It contains operating system as well as used bench-

mark applications. It is important for hard drive to have sufficient read/write speed as well as data transfer speed to ensure that it is not a network bottleneck,

- Network interface controller (NIC) – network card is a direct connection between computer and a network, so it is crucial to have an adequate network card that support tested protocols and required network speed,
- Ethernet cable length - it can have a huge impact on networking performance, especially when the applications have not been optimized. It is important to use adequate cable and adequate length and today's mostly used standards Cat5e and Cat6 can guarantee data transfers up to 100 m without significant signal degradation.

5. RESULTS AND ANALYSIS

Networking performance of three different versions of Windows operating system in their default configurations was measured by using two identical personal computers, laptops with three network benchmark applications. For evaluation, four major networking metrics were used: bandwidth, delay, jitter and CPU usage. By comparison of the same performance measurement results on different operating systems, it can be determined which operating system provides the best networking performance in default configuration. All performance measurement results are shown in Fig. 2 and Fig. 3.

Iperf3 TCP Bandwidth test showed the best performance on Windows 8.1 for smaller packets, up to 84,47% on 128 bytes packet, while on larger packet size all three operating systems show similar performance, as shown on Fig. 2a. CPU usage during the test on client side was the lowest on Windows 8.1 for smaller packets, up to 58% less than Windows 10 and up to 38% less than Windows 10, as shown on Fig 2b. Again, on larger packets all operating systems show similar CPU usage. While measuring network bandwidth by using Iperf3 and UDP protocol, all operating system show almost identical performance, as shown on Fig. 2c. During that test, CPU on Windows 8.1 was the least utilized, as shown on Fig 2d. CPU usage on all operating systems was declining as packet size grew. Iperf3 UDP jitter measurement showed similar performance on Windows 8.1 and Windows 10 while Windows 7 jitter was up 132% larger, which indicates high packet delay variations in Windows 7, as shown on Fig. 2e. When using D-ITG benchmark, TCP bandwidth was very similar for all operating systems for smaller packet sizes, while on larger packets Windows 8.1 and Windows 10 showed up to 38.61% better performance, as shown on Fig. 2f. During that test, Windows 7 used the least CPU resources, as shown on Fig. 2g. When measuring TCP delay with D-ITG, Windows 10 showed significantly better performance (the smallest delay) than

both other operating systems, up to 269% smaller delay, as shown on Fig. 2h. D-ITG TCP jitter measurement resulted in best performance for Windows 8.1 for all packet sizes, as shown on Fig. 2i. D-ITG UDP bandwidth measurement showed similar results for all operating systems on smaller packet sizes while on larger packets Windows 7 and Windows 8.1 show better performance while Windows 10 falls behind, as shown on Fig. 2j. During that test, Windows 7 had the least CPU usage, while Windows 8.1 and Windows 10 had higher and similar CPU usage, as shown on Fig. 3b. D-ITG UDP delay was significantly higher on Windows 8.1 than on remaining operating systems, from 256% to 313% higher in average than on Windows 7 and Windows 10, as shown on Fig. 3a. UDP protocol bandwidth test performed with NetStress showed very similar performance for all operating systems. However, small performance degradation is noticed on Windows 10 during large packets transmission, as show on Fig. 3c.

When comparing results with our previous work in [5] and [8] it is obvious that Windows 8.1 and Windows 10 brought some improvements most regarding TCP and UDP bandwidth while CPU usage remained lower on Windows 7. Due to their older architecture, Windows XP and Windows Vista in most cases showed inferior performance. In [23] authors compared network performance of Windows XP, Vista, 7 and 8 which resulted in similar TCP and UDP throughput for all considered operating systems while average delay was lowest on Windows 8. Our results show decrease of delay on Windows 10. When considering network performance in wireless networks in [14] where authors compared Windows XP, Windows 7 and Fedora 12, highest throughput and lowest delay showed Fedora, while Windows XP showed highest delay and lowest throughput. Having stated the above, it would be interesting, for some future work, to evaluate network performance of latest Windows operating systems with latest Linux based operating systems.

6. CONCLUSION

The main purpose of majority of desktop operating systems is to enable an efficient usage of computer systems' software and hardware resources. Due to a reason that Windows operating systems are the most widely used operating systems on personal computers, they need to satisfy needs of various users as well work with various hardware architectures and components. As a result of technology advancement and globally accessible Internet, services, applications, business and users are shifted online and consequently networking performance has become crucial. Due to complex network infrastructure, numerous internal and external factors can influence the performance and, in this paper, operating system is recognized as one of the crucial components that influence networking performance. In this paper, we present an experimental evaluation and comparison of the networking performance of the three latest versions of Windows operating systems.

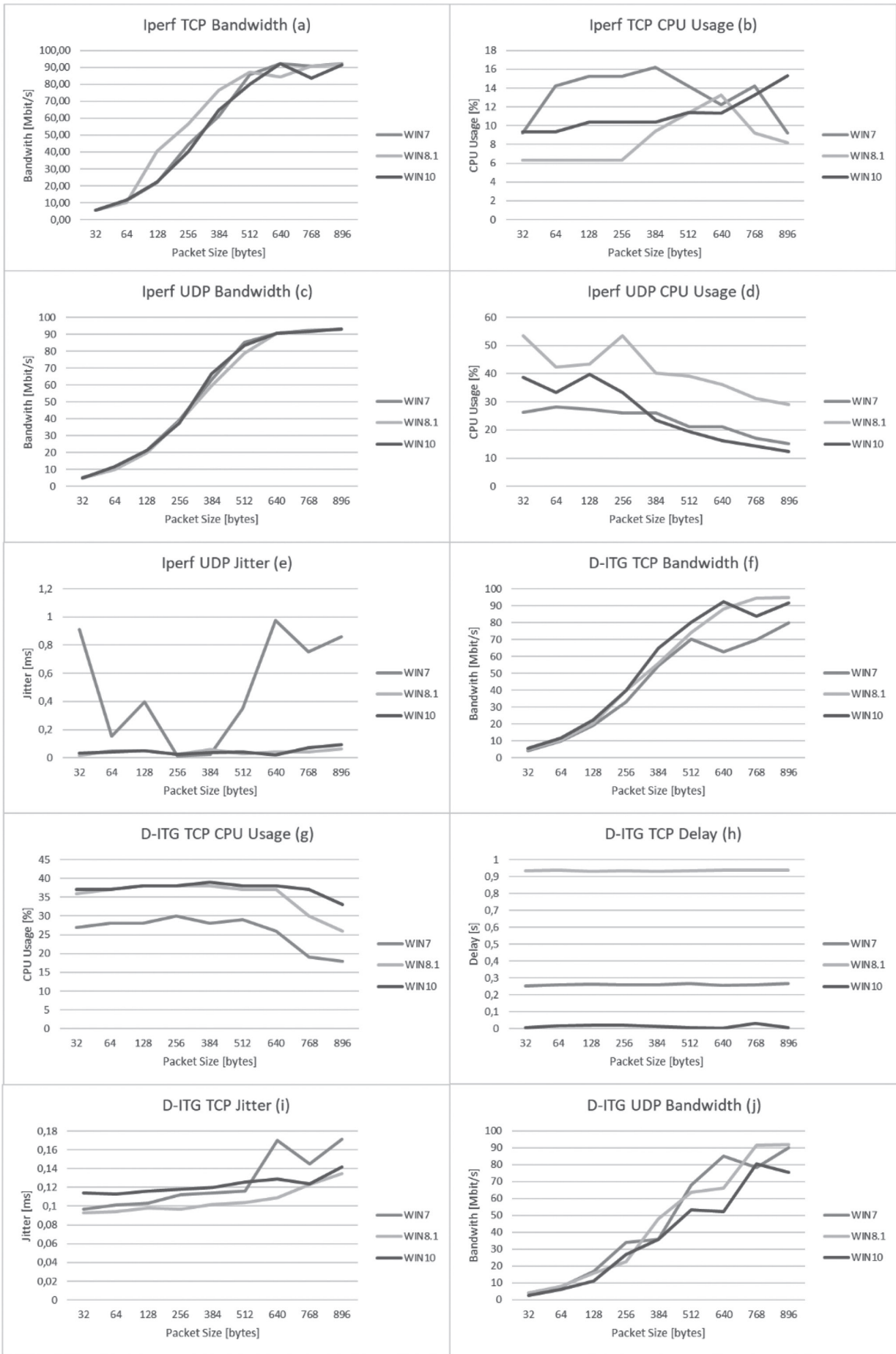


Fig. 2. Performance measurement results (1)

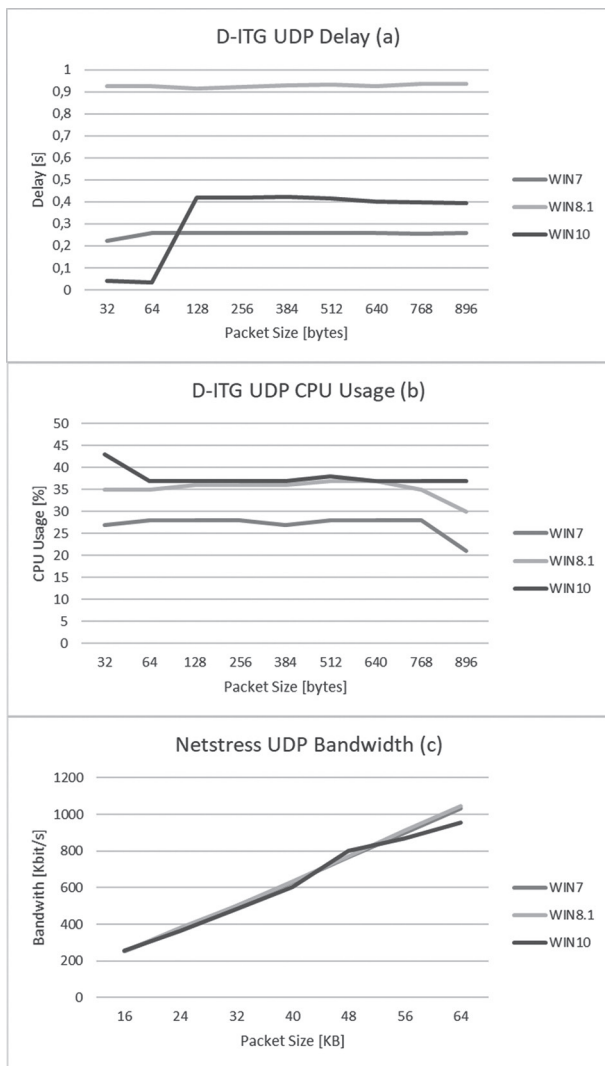


Fig. 3. Performance measurement results (2)

Performance evaluation results showed that Windows 10 brought some improvements in network management by lowering networking delays but with a price of increased CPU usage that however did not improve throughput or lower the jitter when compared to other operating systems. Windows 8.1 has proved to be the most stable operating system when it comes to networking performances since it has the lowest network jitter but at the cost of higher network delays and increased CPU usage. Windows 7 showed lower CPU usage that resulted with a bit higher delays but dramatically higher jitter which means that Windows 7 provides an unstable network connection resulting in a high fluctuation of latency within a short time frame. This could cause unstable packets and their arrival out of sequence that could negatively influence performance of real-time multimedia applications.

7. REFERENCES

- [1] O. Lemeshko, O. Yeremenko, N. Tariki, "Solution for the Default Gateway Protection within Fault-Tolerant Routing in an IP Network", International

Journal of Electrical and Computer Engineering Systems, Vol. 8, No. 1, 2017, pp. 19-26.

- [2] Small Tech News, "Performance comparisons between Windows, WSL and Linux", <https://www.smalltechnews.com/archives/40399> (accessed: 2020)
- [3] IT Pro team, "Windows 10 vs Windows 8.1 vs Windows 7 - Microsoft OS head-to-head", <https://www.itpro.co.uk/operating-systems/25101/windows-10-vs-windows-81-vs-windows-7-microsoft-os-head-to-head-1> (accessed: 2020)
- [4] Net Market Share, Operating System Market Share, <https://netmarketshare.com/operating-system-market-share.aspx> (accessed: 2020)
- [5] J. Balen, G. Martinovic, Z., Hocenski, "Network Performance Evaluation of Latest Windows Operating Systems", Proceedings of the: 20th International Conference on Software, Telecommunications and Computer Networks, SoftCOM, Split, Croatia, 11-13 September 2012, pp. 1-6.
- [6] G. Martinovic, J. Balen, S. Rimac-Drlje, "Impact of the Host Operating Systems on Virtual Machine Performance", Proceedings of the 33rd International Convention MIPRO, Opatija, Croatia, 24-28 May 2010, pp. 613-618.
- [7] J. Balen, K. Vdovjak, G. Martinovic, "Performance Evaluation of Windows Virtual Machines on a Linux Host", *Automatika*, Vol. 61., No. 3, 2020, pp. 425-435
- [8] G. Martinovic, J. Balen, B. Cukic, "Performance Evaluation of Recent Windows Operating Systems", *Journal of Universal Computer Science*, Vol. 18, No. 2, 2012, pp. 218-263.
- [9] J. Balen, D. Vajak, K. Salah, "Comparative Performance Evaluation of Popular Virtual Private Servers", *Journal of Internet Technology*, Vol. 21, No. 2, 2020, pp. 343-356.
- [10] A. Botta, A. Dainotti, A. Pescapè, "A Tool for the Generation of Realistic Network Workload for Emerging Networking Scenarios", *Computer Networks*, Vol. 56, No. 15, 2012, pp. 3531-3547.
- [11] Z. Jiang, "A Linux Server Operating System's Performance Comparison Using Lmbench", Proceedings of the International Conference on Network

- and Information Systems for Computers, Wuhan, China, 15-17 April 2016, pp. 160-164.
- [12] LMBench – Tools for Performance Analysis, Product Documentation, <http://lmbench.sourceforge.net/man/index.html> (accessed: 2020)
- [13] S. Narayan, S. Ishrar, A. Kumar, R. Gupta, Z. Khan, "Performance Analysis of 4to6 and 6to4 Transition Mechanisms over Point to Point and IPSec VPN Protocols", Proceedings of the 13th International Conference on Wireless and Optical Communications Networks, Hyderabad, India, 21-23 July 2016, pp. 1-7.
- [14] S. Kolahi, P. Li, "Performance Analysis of Some Operating Systems in Wireless 802.11n Networks", International Journal of Computing and Network Technology, Vol. 4, No. 1, 2016, pp. 33-40.
- [15] iPerf – The ultimate speed test tool for TCP, UDP and SCTP, User documentation, <https://iperf.fr/> (accessed: 2020)
- [16] C. H. Hsu, U. Kremer. "IPERF: A Framework for Automatic Construction of Performance Prediction Models", Workshop on Profile and Feedback-Directed Compilation, Paris, France, 12 October 1998, pp. 1-10.
- [17] NetStress, Network Benchmarking Tools, Product Documentation, <https://nutsaboutnets-documentation.s3.amazonaws.com/NetStress/NetStress.htm> (accessed: 2020)
- [18] Network Subsystem Performance Tuning, Microsoft Documentation, <https://docs.microsoft.com/en-us/windows-server/networking/technologies/network-subsystem/net-sub-performance-top> (accessed: 2020)
- [19] K. El Khadiri, O. Labouida, N. Elkamoun, R. Hilal, "Performance Evaluation of IPv4/IPv6 Transition Mechanisms for Real-Time Applications using OPNET Modeler", International Journal of Advanced Computer Science and Applications, Vol. 9, No. 4, 2018, pp. 387-392.
- [20] Z. Falt, M. Krulis, D. Bednarek, J. Yaghob, F. Zavoral, "Towards Efficient Parallel Data Stream Processing", Journal of Universal Computer Science, Vol. 21, No. 6, 2015, pp. 816-841.
- [21] Annese Team, "7 Factors that Can Impact Your Network Performance", <https://www.annese.com/blog/7-factors-that-can-impact-your-network-performance> (accessed: 2020)
- [22] C. Lu, C.-S. E. Yeh, Y.-C. Wang, C.-S. Yang, "Hybrid Clouds for Web Systems: Usability and Performance", Journal of Internet Technology, Vol. 19, No. 1, 2018, pp. 187-195.
- [23] S. Debbarma, A. Das, "Empirical Measuring IPv4/IPv6 Network Performance on Microsoft Windows Operating Systems", Proceedings of the 3rd International Conference on Advances in Computing and Communications, Cochin, India, 29-31 August 2013, pp. 393-395.