UDC 621.3

QUALITY OF SERVICE APPLICATION IN WINDOWS SERVER 2012

T. Vince, M. Gorbar
Technical University of Kosice
vul. Letnya, 9, Kosice, 042 00, Slovak Republic. E-mail: tibor.vince@tuke.sk, marian.gorbar@student.tuke.sk

Purpose. The use of Quality of Service is not a common practice for most of the companies. Although some of them use the Quality of Service policies, it's mostly from Cisco. Nevertheless, there is another option that maintains the similar results, but a lot easier to be implemented – Windows Server Quality of Service. The aim of the paper is to show that Quality of Service in Windows Server is satisfactory with its options even for administrators with less technical knowledge of Cisco configurations. Windows Server 2012 may bring the similar options as Cisco, when it comes to convergence of voice, video, and data onto a single IP network among the other business advantages. **Methodology.** A comparative analysis was applied to compare possibilities of Cisco and Windows Server 2012. A set of different test was applied to implement Quality of Service methods in Windows Server 2012 with the aim to detect its possibilities and advantages. **Results.** Two cases of Quality of Service performance, namely Policy-based Quality of Service and PowerShell Quality of Service, were tested. The pros and cons of both systems are further discussed with focus on how does the Quality of Service paper and what is it used for. Moreover, the providing results of running system approve the testing procedure and thus demonstrate the success of the procedure. All performed tests show that the basic principles of Quality of Service can be similarly integrated within Windows Server 2012, as well. From the testing point of view, the results are in the same agreement as from the application of specific rules or policies, as was predicted. Cisco brings more options and ways to match the traffic, set the classes and prefers traffic sensitive to delays, jitter and drops among the others, compare to Windows Server. However, for basic and intermediate purposes of Quality of Service application or in case of small knowledge of Cisco Quality of Service technology, the Windows Server is good alternative. **Originality.** For the first time it was carried out integrated research on possibilities of Quality of Service application in Windows Server 2012. **Practical value.** It was pointed out that Windows Server 2012 Quality of Service application offers almost the same options and ways as Quality of Service application in Cisco platform, can be drawn. Nevertheless, the Quality of Service is still service not as frequently used as it could be. And if it's used than is running mostly on Cisco platforms. However, Windows Quality of Service solutions could be a great alternative to Cisco Quality of Service solutions.

Key words: Quality of Service, Windows Server, computer network.

ЗАСТОСУВАННЯ QUALITY OF SERVICE ДЛЯ WINDOWS SERVER 2012

Т. Вінсе, М. Горбар
Технічний університет Кошице
Парк Коменського, 3, м. Кошіце, 042 00, Словаччина. E-mail: tibor.vince@tuke.sk, marian.gorbar@student.tuke.sk

На даний момент, коли мова заходить про мережі передачі даних в електромеханічних системах керування, безперечним світовим лідером на ринку є Cisco, включаючи Quality of Service та його застосування. Хоча опції Windows Server 2012 надають майже ті самі можливості для ведення бізнесу малим та середнім підприємствам, все більше користувачів віддають перевагу системам Cisco. Метою даної роботи є аналіз і виокремлення можливостей Quality of Service додатків у Windows Server 2012, здебільшого через їх нечасте застосування, на прикладі системи віддаленого керування лабораторним обладнанням університету. Було випробувано два випадки функціонування Quality of Service, а саме, систему якості обслуговування на основі прийнятої політики та на основі PowerShell. Проаналізовано переваги й недоліки обох систем із зосередженням уваги на документуванні Quality of Service та його призначенні. Більш того, надані результати роботи системи підтверджують результати тестів і, таким чином, демонструють успішність функціонування досліджуваної системи для розглянутих задач.

Ключові слова: Quality of Service, Windows Server, комп'ютерна мережа.

PROBLEM STATEMENT. If we have a look nowadays to computer networks, we might imagine it as a frequent highway full of cars. Some cars are faster, some are more expensive and each of them has a different manufacturer. The very same pattern can be applied to packets that flow through the network, carrying different information for end users. Except that in networks, when it comes to speed, the higher is better. As on the roads, the rules and driving etiquette is indispensable part, as well for the networks, the certain policies must be established to keep the system in order. For these purposes, the QoS (Quality of Service) is being used as a measure of several aspects, such as bandwidth, throughput, error rates, jitter, transmission delay etc.

In 21st century the use of QoS still not common practice for most of the companies. Although some of them use the QoS policies, it's mostly from Cisco. Nevertheless, there is another option that maintains the similar results, but a lot easier to be implemented Windows Server QoS. We would like to show that QoS in Windows Server is satisfactory with its options even for administrators with less technical knowledge of Cisco configurations. Windows Server 2012 may bring the similar options as Cisco, when it comes to convergence of voice, video, and data onto a single IP network among the other business advantages.

There are still no specific rules, what is the best way of how to enable the QoS. Many tutorials can be found for Cisco QoS application, but just to try to find a good reference with the possibilities and results of QoS appli-

cation in Windows Server is very difficult. Even though, if there is some reference, it definitely does not go in to details for each of its possibility. More importantly, there is no evidence of comparison with Cisco system up to now. Considering the lack of that information, as well as need of pointing out to Windows Server options, led to theoretical and practical research.

EXPERIMENTAL PART AND RESULTS OBTAINED. *QOS.* QoS is the abbreviation for Quality of Service, a set of standards widely used in computer networks. It controls the bandwidth given by provider to users in the network. The whole mechanism is based on providing different priority and bandwidth to the various types of network services such as VoIP, online gaming, real-time multimedia streaming and so on. Those, who require more bandwidth resources, get higher priority and network speed compared to the services, which are not so sensitive to network speed, as for instance mailing, chatting and browsing.

Quality of Service is affected by many factors, which can be divided into two main groups, depending on the character (e.g. human and/or technical). Availability of service, user information, delays and stability of service belong to group of human factors. Technical factors include scalability, reliability, effectiveness, grade of service and maintainability. [1]

QoS is one of the best ways of how to provide high quality services to the end users. Some of the scenarios, which can happen during the flow of packets in the network topology between a sender and receiver, are described below.

– *Low Throughput* – provided bit rate from the network resources for a data stream is too low.

– *Packet dropping* – occurs, when network devices fail to deliver some of the packets, due to the buffers already full or due to misconfiguration on the network device.

– *Latency* – it takes a long time for packets to reach their destination from the source device, even though the throughput is normal.

– *Errors* – packets sent across the wireless network, or network, where long cables are used, can be corrupted by noise or interference.

– *Jitter* – each of packets can have different delay due to the varying position in the buffers or can take different path to the destination. It can seriously affect the network streaming (e.g. video and/or audio).

– *Out-of-order delivery* – all packets are routed in the network by diverse paths, resulting in a different delivery time of the packets to the destination. When the data stream, which consists of these packets, reaches the destination, the packets can be reordered.

Network technicians use the QoS to guarantee a bandwidth for applications, so their transactions can be processed in amount of time, which is acceptable for users.

There are several benefits of using the QoS.

– The way to control network resources and to allow managing the network from a business rather than a technical perspective.

– Ensures that time-sensitive and mission-critical applications have all the resources they need, while allowing other applications to access the network.

– It improves whole user experience.

– Uses existing resources effectively, reduces costs, thereby delaying or reducing the need for expansion or upgrades [2].

The whole lifecycle of QoS is built of three main steps. Each of them is having its important place, to configure the QoS:

1. *Define QoS* – this step includes project planning, where system engineer's role is to better understand the need of QoS applied in whole organization in present and in future time. This step is important due to the fact that there are defined classes for traffic (Fig. 1), through which all the data will travel in the next step [3].

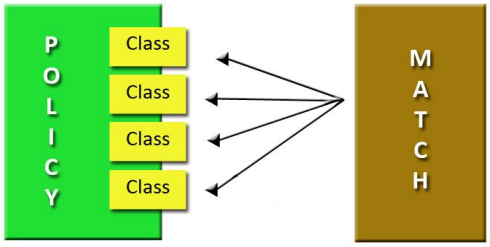

Figure 1 – Virtual laboratory environment

2. *Applying of QoS policy* – in this step administrator applies the selected policies to interface in inbound or outbound direction depending on the way where the QoS needs to take its place (Fig. 2).

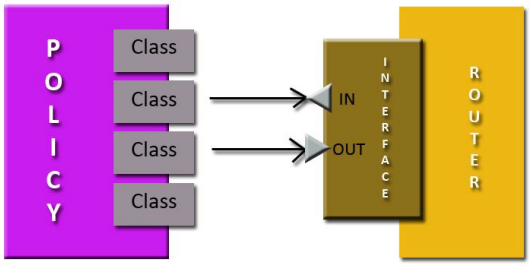

Figure 2 – Application of policies on the interface

3. *Prioritizing of traffic* – packets which travel through the interface are prioritized by QoS. It is done according to the priority set by QoS. Each packet takes a long way through the entire QoS process to get classified, marked and shaped. The flow process through the QoS-based policy is described in more details in Fig. 3 [4].

When packets flow in the network, which is using Quality of Service it has to be first classified according to some rules. This classification divides the flow of packets into multiple classes of service. After the packets are classified into the classes, each of them can be utilized according to the policies, including congestion control, allocation of bandwidth and delays. The system engineer defines the classes and each class defines some specific rules and practices. QoS classes for VoIP, media streaming and online gaming have usually the highest priority. Different priority can be also set to the devices that are in need to have faster access to the resources. Fig. 4 represents the typical QoS policy flow diagram.

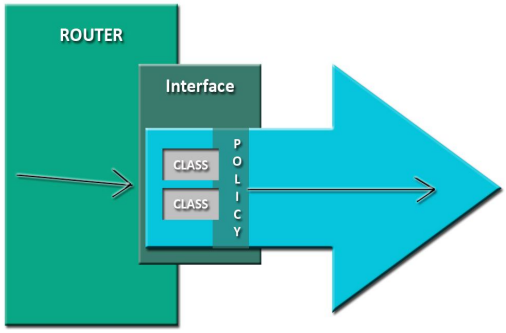

Figure 3 – The way of packet through the router out of interface

Packets are classified to the classes, from which they travel to the different queues:

– *Priority Queue* – is used for the most sensitive packets which demand low latency and low jitter. They will be delivered to the destination with the highest reliability.

– *Class-based Queue* – is used for traffic which is not sensitive to loss or time. Each queue can have specified bandwidth which guarantees availability.

– *Default queue* – is used for packets which are not specified for any class, but are pushed through the network by using the default queue.

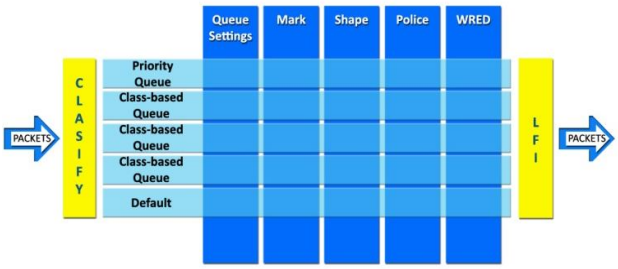

Figure 4 – Flow of packet through QoS

In the queues the packets reserve bandwidth and are prioritized by traffic which enters or leaves the network device. Since the bandwidth of the connection is not infinite it can drop packets in case of full buffer on the queue.

When the packets are identified in the network (using L2 or L3 header marking) and marked with priority, they travel further through the QoS model where all packets are shaped.

If traffic becomes higher than physical interface is able to handle, the shaping and policing techniques are being used to shape it. The ISP tries to avoid dropping the packets and thus it decreases the bandwidth available for end users.

After the packets pass through the Shaping and Policing, they further travel to Weighted Random Early Detection (WRED). WRED is a kind of mechanism which provides congestion avoidance. It prevents a queue from filling the capacity where no other packets could join the buffer and/or could drop down. The packets are dropped if the average queue size is greater than the maximum threshold.

The last stop of the packets flowing through the QoS mechanism is Link Fragmentation and Interleaving (LFI). The main goal of LFI is to decrease the serialization delays on links with slow speed. Serialization delay refers to time needed to serialize the packets into the wide area network (WAN) link [4].

APPLICATION OF QOS. During the evolution of Windows Server operating system quality of service brings new ways to implement QoS policies for the devices connected to the network.

This chapter delivers much more information and options of how the QoS works on Windows platform and how it can be implemented on the Windows Server operating system to meet all administrator and user requirements.

QWAVE. The first method of QoS policy setting that is covered in this paper is dedicated to Quality Windows Audio Video Experience, qWAVE. The set of QoS related software features that are mostly targeted to run in not-enterprise environments such as small office networks or home network brings high quality audio and video user experience. This add-in is typically disabled in environments as enterprise networks. QWAVE provides the best experience when both, the source and destination devices are qWAVE enabled and the traffic supports prioritization. On Windows Server platform, only rate of flow and prioritization services are provided. The feature qWAVE is available now in many wireless access points, wireless network adapters and any other devices like switches and hubs. QWAVE is not installed by default on Windows Server 2012. Administrator is able to install it as a plug-in in server manager. In add features field select a feature qWAVE and click install to proceed with the installation. It is required to restart the server to finish the installation.

Policy-based QoS. The very first step of QoS configuration in Windows Server 2012 using the Group Policy is to bring in the appropriate snap-in to mmc console. In the run window (Windows + R), type mmc and then press enter. Use the Add/Remove Snap-ins in the top File menu. Click on Group Policy Object Editor in the Add or Remove Snap-ins dialog box. The first step to create QoS policies in the Windows Server 2012 is to create policy-based QoS by pressing right-button on the Policy-based QoS in the left menu we can create a new QoS policy or see advanced QoS settings.

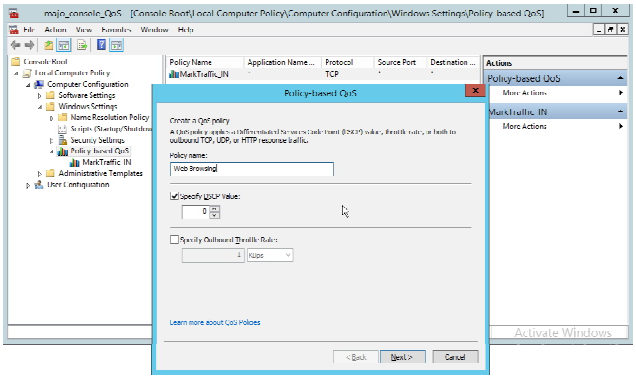

Figure 5 – Policy-based QoS

When creating QoS policies this way several inputs can be updated.

– *Policy name* – the name of policy, for example, Web policy, Lower the speed of link or anything else.

– *Specify DSCP Value* – DSCP value from 0 to 63.

– *Specify Outbound ThrottleRate* – the maximum rate of bandwidth, which can be consumed might be specified in kilobits per second or megabits per second.

The next step of creating this policy is to specify where to apply QoS policy. There are options as applying QoS to all applications, or we can select the applications we want add policy to. There is also a way of selecting HTTP server applications which respond to requests for some URLs.

When selecting all applications this policy matches all applications which are used on Windows Server and sets the policy.

The second option is to use only the applications with this executable name – we either specify the name of the application with its extension (exe) or the whole path of the appropriate application.

Third option is to use HTTP server, by specifying its URL name. Also enhances the feature in a way, to select not just one application file, but whole folder, which may include several executable files responding to the HTTP server.

The next step is to configure IP addresses where the QoS policy will be applied. QoS is applied to outbound traffic which is from a source or a destination IP address (IP address can be either IPv4 or IPv6 address).

When the option ,,Apply QoS to Any source IP address'' is selected, the QoS policy will match all IPs. All these IPs are touched from QoS policy and the action set to this policy is going to be in use for whole subnet which is provided from ISP.

Second way is to specify the source IP address or prefix. There is a way to set the policy which matches only to one host. This step also includes setting the source IP not only to one IP but to whole subnet. When we consider the destination of IP settings the whole step by step must be considered which is mentioned in the section, where we use source IP above.

Clicking the button next the last step shows when creating QoS policy. This step allows the administrator to wisely choose between protocols which match the QoS policy. As you might already know TCP works together with UDP on the 4th OSI layer. UDP is much faster but also less secure compared to TCP.

Specify the source port number is the option where we can prioritize the traffic based on the port number. Important is to give much more priority to real-time applications as to the surfing on the web or sending emails. We can also set the range of the ports not just the one port.

The option for specifying destination port number has the same properties as specifying the source port number. We can also set one port or port range as in case of specifying source ports in this section. After this step the policy-based QoS is created. There can be created many policies in this way which will be applied for whole traffic.

When the policy is created, by clicking the right mouse button, on the policy name in the details panel of the Group Policy Object Editor, several options to manage given policy exists.

Even if policy is created already, actions as delete, edit and view of the properties of the QoS policy are still able to execute by using the right click on the policy.

In the production where many of the Policy-based QoS policies are applied, it may happen that an application of IP address which belongs to several QoS policies will appear. In that case the policy with higher precedence or in the other words, the one which is most specific in matching conditions, will be matched and traffic will be shaped and sent out/in by using this policy.

PowerShell. The last QoS implementation method in Windows Server 2012 describes the method which uses PowerShell. PowerShell is a scripting language and command line especially developed for Windows Server system administration.

This method is much harder to implement because it requires administrator knowledge of PowerShell commands to configure QoS. Compared to the two options mentioned above this one has no GUI. Everything is configured in command line.

Those script cmdlets can be used to configure network Quality of Service:

– *New-NetQosPolicy* – creates a net network QoS policy.

– *Get-NetQosPolicy* – retrieves network QoS policies.

– *Remove-NetQosPolicy* – removes QoS policy.

– *Set-NetQosPolicy* – updates the policy settings [6].

For each of these cmdlets is the user able to get the help if not sure what the syntax of the commands should look like:

– *Get-help <cmdlet> -full* – command where the *<cmdlet>* is the name of the cmdlets, of which full output we want to get, for example, *Remove-NetQosPolicy*, this output shows all options how the cmdlets should look like.

– *Get-help <cmdlet> -detailed* – similarly as in previous command it gives detailed output for the requested command.

– *Get-help <cmdlet> -examples* – output, which shows practical example of implementing the command.

QOS POLICIES TESTING. As mentioned before, all QoS policies consist of match and set actions. Policies in Windows Server 2012 include several set parameter options, in this paper only throttle rate parameter (ThrottleRateActionBitsPerSecond) is used to implicitly show how the flow is influenced in the real traffic. Furthermore, the figures of traffic flow are used for better illustration of how the policy influences the flow of packets in the network. For testing purposes topology with only one router, one switch, one PC and a server. Of course, when using several routers or switches on site, the result is the same as when applying policies. Traffic flow without applied policy is the best-effort option – you get what you got guaranteed from ISP.

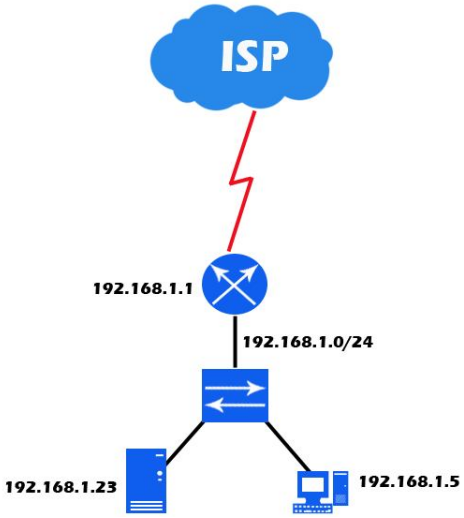

Figure 6 – Topology of testing phase

Policy-based QoS policies. This section brings the comparison which policy has higher priority compared to other because it may happen that administrator creates more policies which match in the same way. Also the graphs to show, that created policy is really in use. To find out the precedence in policies from highest to lowest multiple policies with different matching parameters are created:

– *Gorb_policy_srcIP_1KB* – matches source IP 192.168.1.0/24 and throttle rate is set to 1 KB;

– *Gorb_policy_dstIP_2KB* – matches destination IP 81.95.98.0/24 and throttle rate is set to 2 KB;

– *Gorb_policy_srcPort_3KB* – matches source port from 1 – 65000 and throttle rate is set to 3 KB;

– *Gorb_policy_dstPort_4KB* – matches destination port 80 and throttle rate is set to 4 KB;

– *Gorb_policy_app_5KB* – matches application chrome.exe and throttle rate is set to 5 KB.

After all these policies are created, the output from testing shows that highest priority is given to source IP, policy Gorb_policy_srcIP_1 KB is in use, throughput is decreased to only 1 KB (8 Kbps) as shows the Fig. 7.

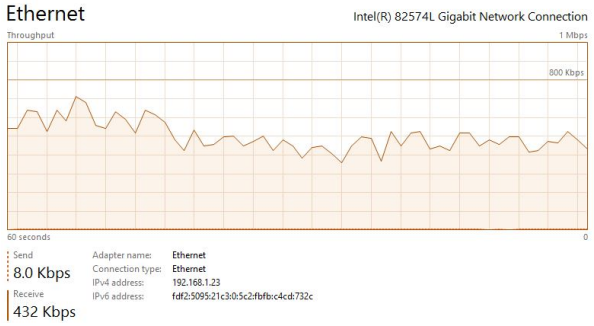

Figure 7 – Bandwidth decreased to 8 KBps

When the Gorb_policy_srcIP_1 KB is removed, the next one which has the highest priority is policy with destination IP parameter set. In this case it is Gorb_policy_dstIP_2 KB where actual throughput is raised to 2 KBps (16 Kbps).

After the IP parameter the next highest priority is given to source port, in this case the policy Gorb_policy_srcPort_3 KB which matches ports 1 – 65000 is in use, so the maximum throughput is 3 KBps (24 Kbps) as visible on the figure below.

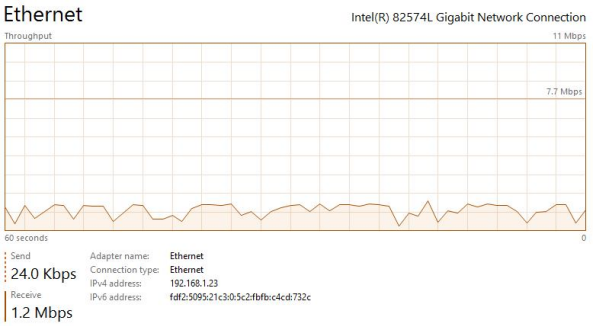

Figure 8 – Policy Gorb_policy_srcPort_3 KB in use

Once again, when the policy Gorb_policy_srcPort_3 KB is removed, the next one (Gorb_policy_dstPort_4 KB) will have the highest priority from the rest of policies and so the throughput is increased to 4 KBps (32 Kbps).

After the IP address and port parameter, the next parameter in policy which takes the highest priority is application name. In this case Gorb_policy_app_5 KB is in use and throughput is set to maximum of 5 KB (40 Kbps).

To sum it up, the precedence from highest priority to lowest is:

– Source IP
– Destination IP
– Source port
– Destination port
– Application
– Default policy (matching everything).

Testing of PowerShell policies. Similarly as in testing of Policy-based QoS policies, the set action is throttle rate where the throughput is changed to show that matched policy is in use and there is some difference. The QoS policies configured via PowerShell have one important difference compared to Policy-based QoS. When the policy is created in PowerShell using the parameter –ThrottleRate the throughput is in Kbps, not in KBps as it is configured in Policy-based QoS. So, when the administrator wants to set throttle rate to 5 KBs using PowerShell CLI the ThrottleRate of 40 Kb must be configured instead.

Similar to Policy-based QoS, the most specific policy wins among the policies. When more policies are created with the same parameters, policy using the smallest throughput has the highest priority.

QoS policy matching application name. The first testing shows the actual throughput which is decreased when policy matching the application is applied. For such a test following policies are created:

– *New-NetQosPolicy –Name def_policy –Default – ThrottleRate 40Kb* – this policy matches all packets which do not belong to any other QoS policy and sets the throughput to 40 KBps.

– New-Net Qos Policy – Name Gorb_policy_CHROME – App Path Name chrome.exe – ThrottleRateAcionBitsPerSecond 10 Mb **–** this policy only matches traffic coming from or to the application chrome.exe and sets the speed to 10 MBps.

The outputs of testing on Fig. 9 show that policies work fine because first half of the graph is the throughput of file downloading using the Explorer while the second part shows the downloading of the file via Chrome web browser. It's clearly visible that throughput is decreased down to 40 Kbps.

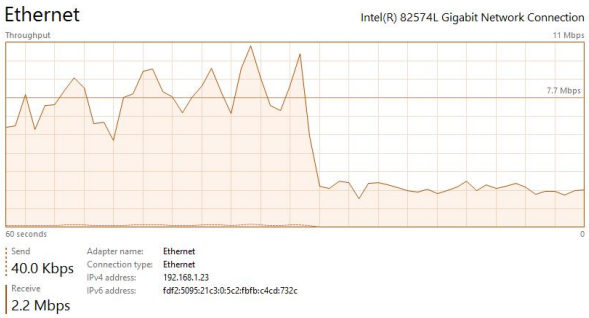

Figure 9 – Bandwidth change when application policy is in use

Using such a policy, the administrator is able to restrict some users of using more resources and provide more bandwidth to the others as well as to provide more of throughput to some service or application while the others have limited resources.

QoS policy matching IP address. Next testing is aimed to the policy with IP prefix parameter. Using the command

*New-Net Qos Policy – Name Gorbar_Policy_SourceIP – IpSrcPrefixMatchCondition 192.168.1.23 – ThrottleRateActionBitsPerSecond 20 Kb***,** all traffic coming from the source IP 192.168.1.23, what is the IP of server, flowing to the ISP is shaped to only 20 Kbps as shows the Fig. 10.

After that, using the command *Set-Net Qos Policy – Name Gorb_Policy_SrcIP – IpSrcPrefixMatchCondition 192.168.1.25*, what is the command to change the source IP to new IP 192.168.1.25, the traffic from server is not matched and the actual throughput is increased back to normal.

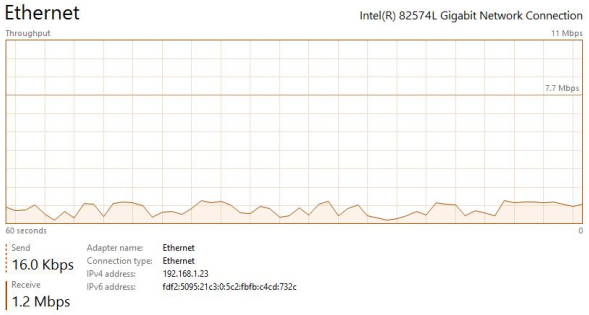

Figure 10 – Traffic from source IP shaped to 20 Kbps

Using *Ip Src Prefix Match Condition*, administrator can prioritize subnets, users or devices among the others. It works in the same way when *IpDstPrefixMatchCondition* parameter is set. For testing purposes, command *New-Net Qos Policy – Name Gorb_policy_dest IP – IpDstPrefixMatchCondition 10.83.0.0/16 – ThrottleRate 20 Kb* creates a new policy which matches all packets going to destination of 10.83.0.0/16 and throughput is decreased to only 20 Kb as shows the Fig. 11.

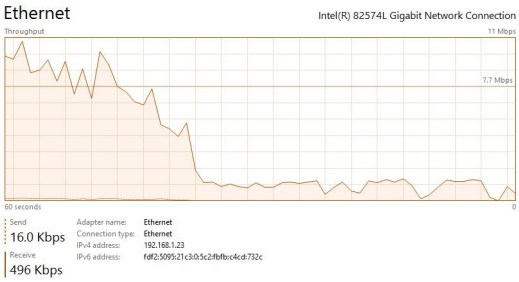

Figure 11 – Decrease of throughput due to policy match

QoS policy matching IP port. Testing of QoS policy configured in Power Shell with parameter *Ip Port Match Condition* seems to be also working. Command *New-Net Qos Policy – Name Gorb_policy_Port_80 – IpPortMatchCondition 80 ThrottleRate 40 Kb* is used which matches the communication on port 80 and sets the throughput to only 40 Kbps (Fig. 12).

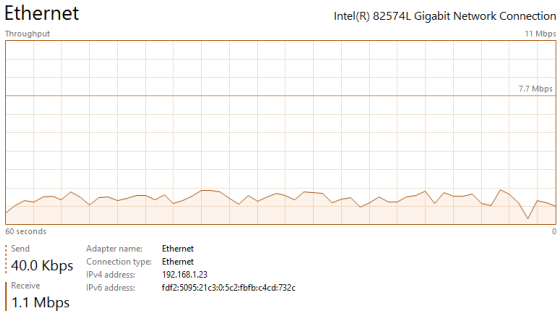

Figure 12 – Bandwidth decreased to 40 Kbps due to policy

Using the commands *IpSrcPortStartMatchCondition* to set the starting value of source port to match and *IpSrcPortEndMatchCondition* to select the ending value of source port or command *IpDstPortStartMatchCondition* to select the starting value of destination port to match and *IpDstPortEndMatchCondition* to select the ending value of destination port to match, administrator does not have to create policies for all ports, in case that more of the ports should be matched. Using the above commands, the range of ports can be applied.

CONCLUSIONS. The results from testing are concluded deeper in paper together with the ways of testing.

The application of QoS policies in Windows Server used in this paper has shown great potential as it could be a suitable service for small and medium sized com-

panies without any deeper knowledge and/or technical skill requirements.

All performed tests show that the basic principles of QoS can be similarly integrated within Windows Server 2012, as well. From the testing point of view, the results are in the same agreement as from the application of specific rules or policies, as was predicted. Cisco brings more options and ways to match the traffic, set the classes and prefers traffic sensitive to delays, jitter and drops among the others, compare to Windows Server. However, for basic and intermediate purposes of QoS application or in case of small knowledge of Cisco QoS technology, the Windows Server is good alternative.

Windows Server 2012 QoS application offers almost the same options and ways as QoS application in Cisco platform, can be drawn.

Nevertheless, the QoS is still service not as frequently used as it could be. And if it's used than is running mostly on Cisco platforms. However, Windows QoS solutions could be a great alternative to Cisco QoS solutions

ACKNOWLEDGEMENT

We support research activities in Slovakia / Project is co-financed from EU funds. This paper was developed within the Project "Centre of Excellence of the Integrated Research & Exploitation the Advanced Materials and Technologies in the Automotive Electronics ", ITMS 26220120055

REFERENCES

1. Peuhkuri, M. (1999), IP Quality of Service, Laboratory of Telecommunications Technology, University of Technology, Helsinki.

2. Microsoft: What is QoS? [online] (2015), < https://technet.microsoft.com/en-us/library/cc757120 (v=ws.10).aspx >, citated: 29.3.2015.

3. Zoeller, C. (2015), How Does QoS Work? [online] <http://packetpushers.net/how-does-qos-work/>, citated: 12.4.2015.

4. Hartmann, D. (2015), Cisco QoS: Link Fragmentation and Interleaving, [online] <http://www.networkworld.com/community/node/39221>, citated: 10.2.2015.

5. Molenaar, R. (2015), QoS Traffic Shaping Explained, [online] <http://www.networkworld.com/community/node/39221>, citated: 15.3.2015.

6. Microsoft (2015), Hyper-V Quality of Service (QoS) [online] < https://technet.microsoft.com/en-us/library/cc757120(v=ws.10).aspx>, citated: 20.3.2015.

7. Apiecionek, J., Czerniak, J.M. and Dobrosielski, W.T. (2014), Quality of services method as a DDoS protection tool, *In Intelligent Systems'* 2014, *Springer International Publishing,* pp. 225–234.

8. Cisco Systems, Inc. (2015), Cisco Nexus 1000 V Quality of Service Configuration Guide, Release 4.0(4)SV1(3), ? [online] <http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus1000/sw/4_0_4_s_v_1_3/qos/configuration/guide/n1000v_qos.pdf>, citated: 15.3.2015.

9. Szigeti, T. et al. (2013), End-to-End QoS Network Design, Quality of Service for Rich-Media and Cloud Networks, *Cisco Press.*

10. Apiecionek, J., Czerniak, J.M. and Zarzycki, H. (2014), Protection Tool for Distributed Denial of Services Attack, In Beyond Databases, Architectures, and Structures: 10th International Conference, BDAS 2014, Ustron, Poland, May 27–30, 2014, *Proceedings,* Vol. 424, p. 405.

11. Guzan, M. and Sobota, B. (2009), "Visualization of chaos", *Journal of Electrical and Electronics Engineering*, Vol. 2, no. 1, pp. 48–51.

12. Bucko, R. and Molnar, J. (2015), Programovanie priemyselnych aplikacii, Vol. 2, no. 1, *vyd. Kosice*, TU–2015, ISBN 978-80-553-1964-3.

ПРИМЕНЕНИЕ QUALITY OF SERVICE ДЛЯ WINDOWS SERVER 2012

Т. Винсе, М. Горбар

Технический университет Кошице

Парк Коменского, 3, г. Кошице, 042 000, Словакия. E-mail: tibor.vince@tuke.sk, marian.gorbar@student.tuke.sk

В настоящее время, если речь идёт о сетях передачи данных в электромеханических системах управления, безусловным лидером на рынке является Cisco, включая Quality of Service и его приложения. Несмотря на то, что опции Windows Server 2012 предоставляют практически те же возможности для ведения бизнеса малыми и средними предприятиями, всё больше пользователей отдают предпочтение системам Cisco. Целью данной работы является анализ и выделение возможностей Quality of Service приложений в Windows Server 2012, в основном ввиду их редкого применения, на основе системы удалённого управления лабораторным оборудованием университета. Были исследованы два случая функционирования Quality of Service, а именно, система качества обслуживания на основе принятой политики и на основе PowerShell. Выполнен анализ преимуществ и недостатков обеих систем с акцентом на документированные Quality of Service и его предназначение. Более того, предоставленные результаты работы системы подтверждают результаты тестов и, таким образом, демонстрируют успешность функционирования исследуемой системы для рассмотренных задач.

Ключевые слова: Quality of Service, Windows Server, компьютерная сеть.

Стаття надійшла 08.02.2016.