Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Information Technology

Master's Thesis

Implementation of QoS in a company network

Farhad Abbasi

2024 CZU Prague

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

Farhad Abbasi

Informatics

Thesis title

Implementation of QoS in a company network

Objectives of thesis

The main objectives of the thesis are to specify and analyze the typical network of an IT outsourcing company and propose the implementation of QoS methods. The implementation will be validated using a simulation environment.
Partial objectives are:
- Study and analyze available literature and online resources regarding company networks and quality of service
- Specify requirements for QoS implementation in a model scenario within a typical company network
- Propose a solution for QoS implementation and validate the approach using a simulation environment
- Analyze the suitability of the solution and formulate recommendations

Methodology

The methodology of the theoretical part consists of a review of available literature and online resources. The thesis addresses the implementation of QoS in an IT outsourcing company network with a specific focus on queuing theory. A typical corporate network will be described and analyzed, focusing on data infrastructure, VoIP, and video. Requirements for the implementation of QoS methods will be specified in a model situation. The proposed implementation will be experimentally verified using a suitable simulation environment. The conclusions of the thesis will utilize the results of both the theoretical and practical parts.

The proposed extent of the thesis

50-60

Keywords

ArubaOS, Best effort, CBWFQ, Cisco IOS, Delay, DiffServ, DSCP, EVE-NG, FIFO, IntServ, IP Network, IOL, Jitter, Packet loss, QoS, RSVP, WFQ

Recommended information sources

Allred, Miriam. 2018. Aruba Certified Switching Professional: Official Certification Study Guide (HPE6-A45). San Francisco: Hewlett Packard Enterprise Press, 2018. ISBN-10: 1942741812

Anthony Bruno, Steve Jordan. September 9, 2020. CCNP Enterprise Design 300-420 Official Cert Guide: Designing Cisco eEnterprise Networks. Hoboken, New Jersey: Cisco Press; 1st edition, September 9, 2020. ISBN-10: 0136575196

Aruba Education Development Team. August 2, 2018. Aruba Certified Design Professional: Official Certification Study Guide (HPE6-A47). San Francisco, CA 94107: HPE Press, August 2, 2018.

DGEWORTH, Brad, RIOS, Ramiro Garza, GOOLEY, Jason and HUCABY, Dave. CCNP and CCIE Enterprise Core Encore 350-401. San Jose, CA : Cisco Press, 2020. ISBN-10: 1587145235, ISBN-13: 978-1587145230

Expected date of thesis defence

2023/24 SS – PEF

The Diploma Thesis Supervisor

Ing. Jan Pavlík, Ph.D.

Supervising department

Department of Information Technologies

Advisor of thesis

Ing. Tomáš Vokoun

Electronic approval: 4. 7. 2023

doc. Ing. Jiří Vaněk, Ph.D.

Head of department

Electronic approval: 3. 11. 2023

doc. Ing. Tomáš Šubrt, Ph.D.

Dean

Prague on 06. 02. 2024

Declaration

I declare that I have worked on my master's thesis titled "Implementation of QoS in a company network" by myself and I have used only the sources mentioned at the end of the thesis. As the author of the master's thesis, I declare that the thesis does not break any copyrights.

In Prague on 28.3.2024 ______________________________

Acknowledgement

I am profoundly grateful to my father for his boundless love and unwavering support, which have been my driving force throughout the education journey. His encouragement has meant the world to me. Additionally, I extend my sincere appreciation to my supervisor Dr. Jan Pavlík and consultant Ing.Tomas Vokoun for their guidance and support, which played a pivotal role in the successful completion of this work.

Implementation of QoS in a company network

Abstract

This master's thesis studies the network structure of an IT outsourcing company and proposes Quality of Service (QoS) methods for improvement. It involves analyzing existing literature on corporate networks and QoS, specifying requirements for QoS implementation, suggesting a solution, and validating it through simulation. The focus is on queuing theory within the context of data infrastructure, VoIP, and video services. The proposed QoS implementation will be tested using simulation, and conclusions will inform recommendations for future implementations.

Keywords

ArubaOS, Best Effort, CBWFQ, Cisco IOS, Delay, DiffServ, DSCP, EVE-NG, FIFO, IntServ, IP Network, IOL, Jitter, MQC, Packet loss, QoS, RSVP, WFQ

Implementace QoS ve firemní síti

Abstrakt

Tato diplomová práce studuje síťovou strukturu společnosti outsourcingu IT a navrhuje metody pro zlepšení kvality služeb (QoS). Zahrnuje analýzu existující literatury o podnikových sítích a QoS, specifikaci požadavků na implementaci QoS, návrh řešení a jeho ověření pomocí simulace. Důraz je kladen na teorii front v kontextu datové infrastruktury, VoIP a video služeb. Navržená implementace QoS bude testována pomocí simulace a závěry poskytnou doporučení pro budoucí implementace.

Klíčová slova

ArubaOS, Best Effort, CBWFQ, Cisco IOS, Delay, DiffServ, DSCP, EVE-NG, FIFO, IntServ, IP IP síť, IOL, Jitter, MQC, Packet loss, QoS, RSVP, WFQ

Table of content

1. Introduction

This master thesis analyses and specifies the network of an IT outsourcing company, proposing a solution for Quality of Service (QoS) implementation. It validates the approach using a simulation environment and specifies QoS implementation requirements in a model scenario within a typical company network.

The first chapter focuses on the theory of Quality of Service (QoS) and how it's applied across various types of networks. It consists of 11 parts: 1) Traffic Characteristics, 2) Planning & Implementing QoS policies, 3) QoS Models, 4) Classification and Marking of packets, 5) Congestion management, 6) Congestion Avoidance, 7) Shaping & Policing, 8) QoS in LAN

The second chapter deals with practical part, it consists of three parts: 1) Network Simulation, 2) EVE-NG setup, 3) QoS scenarios. The first part distinguishes between the two terms simulation and emulation and describes the most popular emulators used to emulate our practical part. The second part describes how to prepare and set up the laboratory for the deployment of the third part. The third part describes typical design of company network and tests different scenarios of QoS implementation in a designed network.

2. Objective and Methodology

2.1 Objectives

The main objectives of the thesis are to specify and analyse the typical network of an IT outsourcing company and propose the implementation of QoS methods. The implementation will be validated using a simulation environment.

Partial objectives are:

- Study and analyse available literature and online resources regarding company networks and quality of service.
- Specify requirements for QoS implementation in a model scenario within a typical company network.
- Propose a solution for QoS implementation and validate the approach using a simulation environment.
- Analyses the suitability of the solution and formulate recommendations.

2.2 Methodology

The methodology of the theoretical part consists of a review of available literature and online resources. The thesis addresses the implementation of QoS in an IT outsourcing company network with a specific focus on queuing theory. A typical corporate network will be described and analysed, focusing on data infrastructure, VoIP, and video. Requirements for the implementation of QoS methods will be specified in a model situation. The proposed implementation will be experimentally verified using a suitable simulation environment. The conclusions of the thesis will utilize the results of both the theoretical and practical parts.

3. Literature Review

In a network, quality of service QoS) is a system or process that controls traffic to minimize packet loss, latency, and jitter and guarantee the smooth operation of vital applications, such real-time audio and video. By allocating precedence to particular kinds of data within the network, quality of service in computer networks regulates and maintains network resources. In reality, network traffic management, or network quality of service, is typically applied to networks that transport traffic for resource-intensive systems. Internet Protocol TV (IPTV), online gaming, media broadcasting, video conferencing, video on demand (VoD), and voice over IP (VoIP) are among the common services that demand it. Organizations can use the quality of service (QoS) of their network to maximize the performance of several applications running on it. They can also observe the bit rate, latency, jitter, and packet loss of their network using specialized tools and methodologies, and by balancing their traffic, they can achieve QoS. With the help of this function, you can make sure that network traffic is engineered and that packets are routed differently over the internet or other networks in order to prevent transmission delays. In order to maintain a specific degree of network performance and to guarantee that the business meets the anticipated quality of service for applications, many organizations incorporate QoS into their Service Level Agreement (SLA) with their network service provider. Organizations can objectively quantify network quality of service (QoS) using a variety of factors, including the following. Network quality of service parameters are one of the most essential aspects of the QoS discussion (1):

A. **Bandwidth:**

The bandwidth of a network communication link is the capacity to transfer the maximum amount of data from one point to another point in each period. By controlling bandwidth and giving priority to applications with higher performance requirements over other resources, Quality of Service (QoS) enhances network performance. In a network, bandwidth is the most crucial component of quality. Actually, bandwidth is where all quality problems stem from. We always have poor bandwidth, so we always require quality of service. If our network bandwidth is low, we need quality of service; if it is not low, we don't need quality of service (2).

B. Delay:

Delay is the time that takes for a packet to travel from source to destination and should be as close to zero as possible. Delay can often be affected by queuing latency, which occurs during times of congestion and a packet waits in a queue before being sent. QoS enables organizations to avoid this by creating priority queues for certain types of traffic. For example, if a Voice over IP (VoIP) call has high latency, users can experience echo and overlapping sound. In an IP network, there are six types of commonly identified delays (2):

- **Serialization delay**: This is the time taken to convert the (logical) packet into an electrical pulse on a link. Serialization delay depends on link speed and packet size.

$$S.D = \frac{\#\ of\ bits\ Sent}{Link\ Speed}$$

- **Propagation delay:** A bit sent by router A over the outgoing link needs to be transmitted to router B immediately. Propagation delay is the amount of time needed to send this bit from router B to the beginning of the link. Depending on the kind of physical media being used in the link, the speed of bit movement is really the speed at which electromagnetic waves propagate over the link (optical fiber, twisted pair, wireless, etc). The moment the last bit of the packet arrives at node B, the packet is stored in router B, which performs the same processes (processing and routing the packet to the appropriate output). Typically, the propagation delay of Wide Area Networks (WANs) is on the order of milliseconds. The propagation delay in a communication link is equal to the distance between two routers divided by the propagation speed of electromagnetic waves (2)**:**

$$P.D = \frac{Lengt\ of\ link\ (m)}{2.1x10^8}$$

- **Queuing delay:** Packets wait to be despatched until their turn in the output queue. This delay is what we call a "queuing delay." The amount of delay experienced by a particular packet depends on how many packets are in the transmission queue ahead of it. If there are no packets sent on the outgoing 12 link and the queue is empty, our packet queue delay will be 0. However, if there is a lot of router traffic and a large number of packets in it, our packet will have to wait a long time in the transmission queue.Queuing latency can, in fact, vary from milliseconds to microseconds (6).

- **Processing delay:** Part of the processing delay is the time the router takes to look over the packet header and figure out which outbound link is best for it. The amount of time the router takes to verify the integrity of the packet bits is one of the additional variables that may cause the processing delay. High-speed routers usually have processing delay of a few microseconds or less. Router A processes the packet and then forwards it to router B via the outgoing link queue (6).

- **Codec delay:** The codec's job is to take our voice over the IP phone or Videos and convert it into a packet that causes a delay.

C. Jitter:

Jitter means delay variation, i.e. the first packet arrives at the destination with 100ms and the second packet with 120ms, here we have a jitter of 20ms which can be important for us in the audio discussion which leads to vibration in the voice packet. Irregular speed of packets in the network which is created because of congestion and can lead to late arrival of packets and them out of sequence. The Jitter presents a significant challenge, particularly for real-time services. If changes occur in the network topology, such as connection failures or the discovery of more efficient routes for service packets, it also causes Jitter. This variability poses a substantial challenge for voice services, where packets must be transmitted and received every 20 milliseconds. To mitigate the impact of this variability, a Jitter Buffer is implemented on the receiving end. This buffer acts as a delay memory, stabilizing the delay from dynamic to consistent, thereby improving the reliability of real-time services. Excessive vibration can reduce the quality of audio and video communications and produce audio and video distortion or gaps in the transmission. The same instruments that lower latency also lower jitter: Increased LFI, traffic shaping, bandwidth, and queuing (9).

D. Packet Loss:

When network links get busy, routers and switches start dropping packets, which is known as packet loss. When packets are dropped during real-time audio or video conversations, there may be vibrations and speech pauses. Packets may be dropped when a queue or queue of packets waiting to be transmitted overflows. Packet loss usually occurs due to congestion in the network. QoS enables organizations to decide which packets to drop in this event. One of the reasons for packet loss is packet noise during routing, which results in FCS recognizing this packet as an error, which leads to dropping this packet. However, the bit error rate (BER) due to noise is now considerably lower; however, the Tail Drop is the more significant cause. A router or other network device buffers as many packets as it can under the standard tail drop technique, dropping new packets when the buffer is full. The tail drop technique divides and arbitrarily allocates buffer space amongst traffic flows. For tcp-based applications that are resiliency-aware, packet loss is negligible; however, for udp-based applications—particularly voice apps—loss is critical because even in the event of an application failure, the udp-based application resends the lost packets. What tools can help us reduce packet loss that occurs through tail drop? One of the tools is queuing, increasing the length of the queue. Actually increasing the queue length, we increase the queuing delay. This technique is suitable for Not delay-sensitive traffic (tcp-based). Another tool is also queuing but creating two queues, one with longer length and one with shorter length, where delay-sensitive traffic is placed in a smaller queue and loss-sensitive traffic is placed in a longer queue (9).

3.1 Traffic Characteristics

3.1.1 Voice traffic characteristics

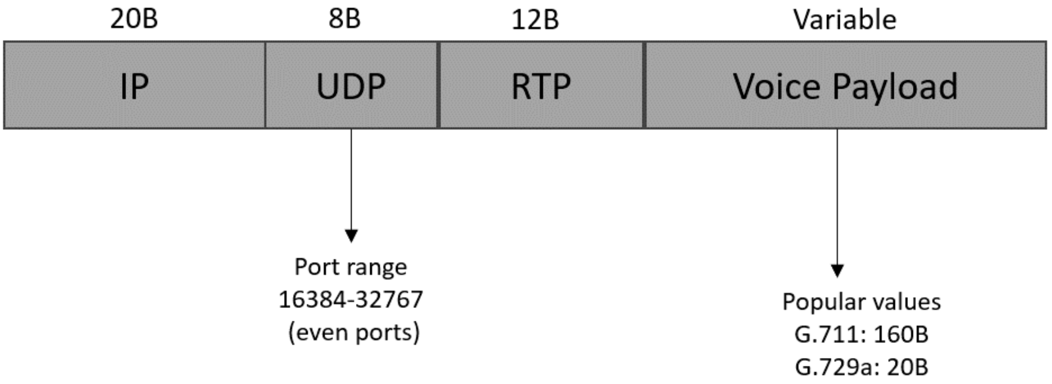


Figure 1: Original voice packet, source: (5)

When the Voice is converted into a packet (Voice Payload) by a codec, it is delivered to a protocol called RTP. RTP is based on udp. After adding its header, it gives the traffic to udp 15 and then the traffic is delivered to IP and then will be sent. The size of the payload depends on the codecs, the codec samples our sound puts these samples in the packet, and sends them. Naturally, the more samples of our sound are taken, the more they will be, finally the sound that is reconstructed is a better sound, and on the other side, the less the number of samples, the lower the quality of the reconstructed sound. For example, the G.711 codec provides a 160-byte payload, while the G.729a codec provides a 20-byte payload. This shows that G.729a has fewer samples, which has a lower quality, but the packet size is also smaller, and it should be considered if it will lead to poor quality doesn‘t mean It should not be used, but anyway, the quality parameters of G.711 are higher than G.729a. This shows that G.729a has fewer samples, which has a lower quality, but the packet size is also smaller, and it should be considered if it will lead to poor quality doesn‘t mean It should not be used, but anyway, the quality parameters of G.711 are higher than G.729a (8).

Codec	**Bit rate for payload (kbps)**	**Size of Payload (20ms- Default in Cisco IOS software)**
G.711 Pulse Code Modulation (PMP)	64	160 B
G.726 ADPCM	32	80 B

G.729a	8	20 B
G.723.1 ACELP	3.3	20 B

Table 1: Popular voice codecs and payload bandwidth requirement, source: (8)

As we can see in the Table 1 it should be kept in mind that if we use the G.711 codec, the bandwidth required to send its payload is equal to 64 kbps. Each of these codecs takes 20 milliseconds (ms) of our voice, that is, every 20 milliseconds of our voice is taken and converted into a packet (8).

$$\mathbf{64}\ kbps = \frac{64000}{8} = 8000B$$

$$8000B\ x\ 20ms = \mathbf{160}B$$

If the codec packets are large, it leads to a serialization delay, that is, those packets are no longer used, and accordingly, every 20 milliseconds, the voice is captured and converted into a packet (8).

	Bandwidth	**Delay**	**Jitter**	**Loss**
Voice payload	Low	Low	Low	Low
Voice signaling	Low	Low	Medium	Medium

Table 2: A comparison between Voice payload vs Voice signaling packets, source: (8)

Both Voice payload and Voice signaling do not require a lot of bandwidth, and if we allocate more bandwidth, it means that we have wasted it. Both are sensitive to Delay, but voice payload is more sensitive to Jitter and Loss (8).

The following Table 3 describes the standards for a one-way delay (The time it takes for the voice packet to reach from the source IP phone to the destination IP phone:

1-way Delay (ms)	**Description**
0-150	ITU G.114's recommended acceptable range
0-200	Cisco's recommended acceptable range
150-400	ITU G.114's recommended range for degraded services
+400	ITU G.411's range of unacceptable in all delay cases

Table 3: One-way Delay Budget Guideline, source: (8)

3.1.2 Video traffic characteristics

Video is usually available in networks in two ways: **a) Interactive video_** It is a two-way or multi-way interaction (video conferencing system), **b) non-Interactive video_** one-way interaction (IPTV, e-learning streams). H.323 protocol is used as the video signaling and RTP is used as the transport protocol that carries the video packet. Since interactive video is completely two-way, it is more sensitive to delay and jitter than non-interactive. We must keep in mind that whenever we have a video stream, this stream or data consists of two parts, one is audio, and the other is video. For audio traffic, the same audio codecs (G.729a, G.711) whose task is to convert the audio stream into a packet and video codecs are ITU H.261, MPEG (8):

Video Codec	Required Range
MPEG-1	500-1500 kbps
MPEG-2	1.5-10 Mbps
MPEG-4	28.8-400 kbps
H.261	100-400 kbps

Table 4: The most popular video codecs with their bandwidth requirements, source: (8)

As you can see in the below Table 5, In terms of video traffic requirements, especially interactive video traffic, the requirements are the same as for voice traffic, except that we don't need bandwidth in voice, but we do need bandwidth in video traffic same as Delay, Jitter and Loss (8).

	Bandwidth	Delay	Jitter	Loss
Voice Payload	Low	Low	Low	Low
Video Payload	High	Low	Low	Low
Voice Singalling	Low	Low	Medium	Medium
Video Signalling	Low	Low	Medium	Medium

Table 5: Comparison of quality requirements between video and audio, source: (8)

In the Table 6 below, Audio and video are compared in terms of bandwidth requirements, as mentioned, video consists of 2 flows in each direction, but audio has only one flow. The audio packet size is fixed based on codec, but video packet size is variable (8):

Feature	Voice	Video
Number of flows in each direction	1	2 (1:Voice, 1:Video)
Packet size	Static, based on codec	Variable
Packet rate	constant	Variable

Table 6: Comparison between audio and video from the point of view of bandwidth requirement, source: (8)

As for delay, the status of the video varies depending on whether the video is interactive or non-interactive. If the video is interactive, we should say that it is similar to voice in terms of delay sensitivities. But if we have non-interactive video, the amount of delay can be very high, even up to 30 seconds. Suppose a stream is playing from a computer. Computers usually have a de-jitter buffer to neutralize the jitter. They collect information and then broadcast it, it may even record for 30 seconds and then broadcast it. Packet loss in video traffic, in both interactive and non-interactive video modes, since there is no type of recovery in UDP, it results in loss of parts of the video and if this amount of loss is high, the video will be quite jerky. This problem is solved by the queue tool, queue with longer length or Random Early Detection (RED) (8).

	Bandwidth	Delay	Jitter	Loss
Voice Payload	low	low	low	low
Video Payload Interactive (2way)	High	low	low	low
Video Payload Streaming (1way)	High	High	High	low
Video Signalling	low	low	Medium	Medium
Voice Signalling	low	low	Medium	Medium

Table 7: A full Comparison of quality requirements between video and audio, source: (8)

3.1.3 Data traffic characteristics

Apps use TCP or UDP. TCP corrects errors, important for error-sensitive data like email. Traffic can be steady or sudden. Network control traffic is usually steady but may briefly surge. Modern networks handle this. Some TCP apps, like FTP, can use a lot of bandwidth, especially for large downloads. Data Traffic Characteristic (8) :

- Smooth
- bursty Benign/greedy
- Drop insensitive
- Delay insensitive
- TCP retransmits

For data traffic, despite being less sensitive to drops and delays compared to voice and video, network administrators prioritize user experience, called Quality of Experience (QoE). They focus on two main factors: A) Reliability: Ensuring data reaches its destination accurately. B) Performance: Enhancing speed and responsiveness. These factors are key for a good user experience in data transmission (8).

3.2 Planning & Implementing QoS Policies

When we want to implement a QoS policy in the network, what actions should we take? Cisco recommends three steps (3):

> **Step 1:** Identify traffic and its requirement. Network audit by using trace analysis tools, management tools, and Network-Based Application Recognition (NBAR). Business audit to determine the importance of the discovered traffic types to the business.
>
> **Step 2:** Divide traffic into classes. One class for each Voice payload, Video payload, Signalling, Mission Critical, Transactional, and Best-Effort.
>
> **Step 3:** Define QoS policy for each class.

Cisco Modular QoS CLI (MQC)

The main problem in the traditional way of configuring QoS features in Cisco IOS is that each feature is configured in its own method, which leads to many configurations in different ways and difficulty in QoS configuration. Cisco introduced a modular configuration method called Modular QoS CLI (MQC) that implements all the features in the simplest and fastest way (1):

• **Classification** Configuration	**class-map** *myclass1*
	class-map *myclass2*

• **Action/PHB** Configuration	**policy-map** *mypolicy*
	class *myclass1*
	class *myclass2*

• **Enable on Interface**	**interface** *interface_id* **service-policy output** *mypolicy*

3.3 QoS Model

Before we can work with QoS tools, we must first know what model of QoS we want to implement in our network. We have 3 QoS models to implement in the IP network (1):

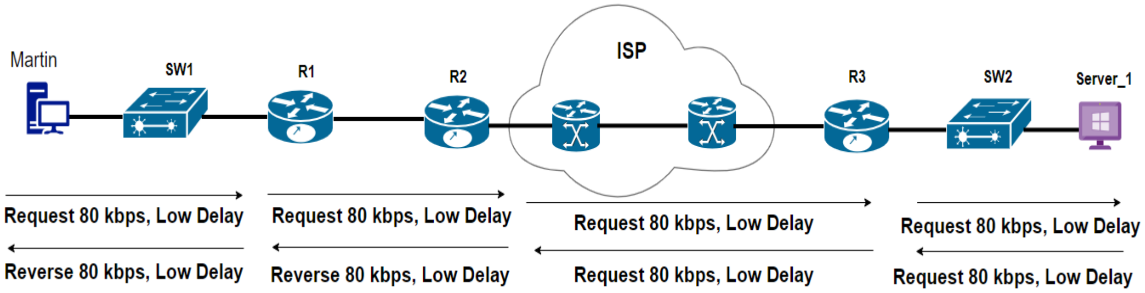
- Best-Effort [**BE**]
- Integrated Service [**IntServ**]
- Differentiated Service [**DiffServ**]

3.3.1 Best-Effort (BE)

The Best-Effort (BE) model of QoS in which all packets have the same priority and there is no guarantee that the packets will reach their destination. This method is used when there is no specific policy for QoS in the network and it doesn’t support QoS (2).

3.3.2 Integrated Service (IntServ)

QoS model in which entire end-to-end packet for a packet is ensured certain minimum QoS characteristics prior to packet transmission. Initial RFCs published by IETF in mid 1990s: RFC 1633, RFC 2211, and RFC 2212. Resource Reservation Protocol (RSVP) used as primary protocol to setup the path, requires every node along path to heed its reservations, to keep per-flow state. In this model, it is to provide a dedicated route/path for packets from the beginning to the end of the path. For this purpose, the is used. This model is older than the DiffServ and was originally proposed. The idea is that traffic in Figure 2 is sent from Martin's computer to server_1. In the IntServ model, network resources are reserved by the RSVP protocol. Martin's computer sends a request to the destination requesting 80 kbps of reserved bandwidth and low delay (2).



Martin
SW1
R1
R2
ISP
R3
SW2
Server_1
Request 80 kbps, Low Delay
Reverse 80 kbps, Low Delay
Request 80 kbps, Low Delay
Reverse 80 kbps, Low Delay
Request 80 kbps, Low Delay
Request 80 kbps, Low Delay
Request 80 kbps, Low Delay
Request 80 kbps, Low Delay

Figure 2: Resource Reservation Protocol (RSVP), , source: author

SW1 passes the bandwidth to the following device in the direction of the destination if it can reserve this quantity with minimal latency. When every device in the path between the source and the destination can reserve the necessary bandwidth and latency, traffic starts to flow from the source to the destination. Although this model works well, we are unable to configure it for every device due to its incompatibility. Thus, RSVP makes this reservation and reserves bandwidth and delay for traffic based on its needs before delivering the traffic; this is the IntServ concept. RSVP completes this after the path is completed and the traffic is rerouted. There are two primary parts to the Intserv model: a) Resource reservation, b) Admission Control. Resource Reservation is a component that reserves the required resources, Admission Control is a component that controls whether the required amount of bandwidth or delay can be allocated by the device or not. Since the IntServ model allocates bandwidth for each flow, it is very difficult and almost unreasonable to reserve for all flows.

3.3.3 Differentiated Service (DiffServ)

Differentiated Services designed to address challenges of Integrated Services (RFCs 2474, 2597, 2598, 3246, and 4594). The DiffServ model describes various behaviors to be adopted by each compliant node which is called Per-Hop Behaviors (PHB).The existing resources can be used more effectively. The user does not request a reservation in advance. It sends its data, the first router puts a value in the IP header that indicates the priority of the data. Each router that the data will visit, if the necessary QoS settings are made, reads the priority value and treats the data accordingly. If the router on the route has no QoS configuration, the device doesn‘t care at all, treats it like an ordinary packet, puts the packet in one of the default queues and forwards it sequentially. In the IntServ model, each router on the route must understand the priority, while the DiffServ model offers the opportunity to address a much wider area, namely scaling. To make an analogy, imagine that there are traffic police at every intersection. It is accompanied by the Prime Minister. All police officers on the route he will pass are instructed to evacuate the intersections from the beginning. But then the prime minister leaves. This approach is the Inserv approach. DiffServ, on the other hand, has an approach to why all junctions should be emptied from the beginning. When the traffic police see the escort, he evacuates the intersection. Thus, existing roads can be used more effectively (for everyone) (2).

3.4 Classification & Marking

Do you know what traffic is important for you...but how does the ROUTER know that? This is done by the Classification. Classification is a feature that identify traffic based distinctive differences. Prior to analysis, traffic needs to be separated into "classes." Each "class" of traffic will be subjected to the same QoS protocols and packet analysis will be used to distinguish between various flows. Packets are marked to ensure that analysis occurs just once, typically at the network's ingress edge. This starts as a business decision, by identifying business needs and getting executive buy-in. Most common ways of classifying traffic: 1) Marking, 2) Addressing, and 3) Application Signatures. All DiffServ-based QoS scenarios use classification and marking tools. Marking is a QoS tool that classifies packets based on their header content and then marks the packets by changing several bits in a series of specific fields in the headers (7).

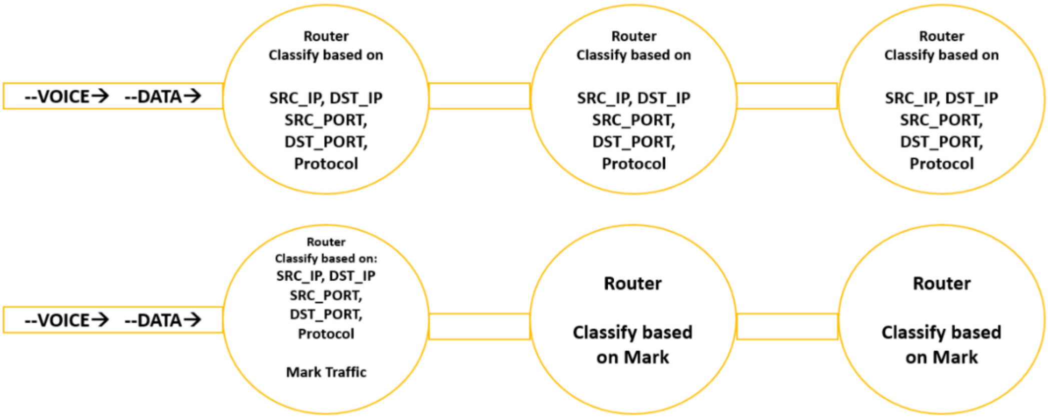
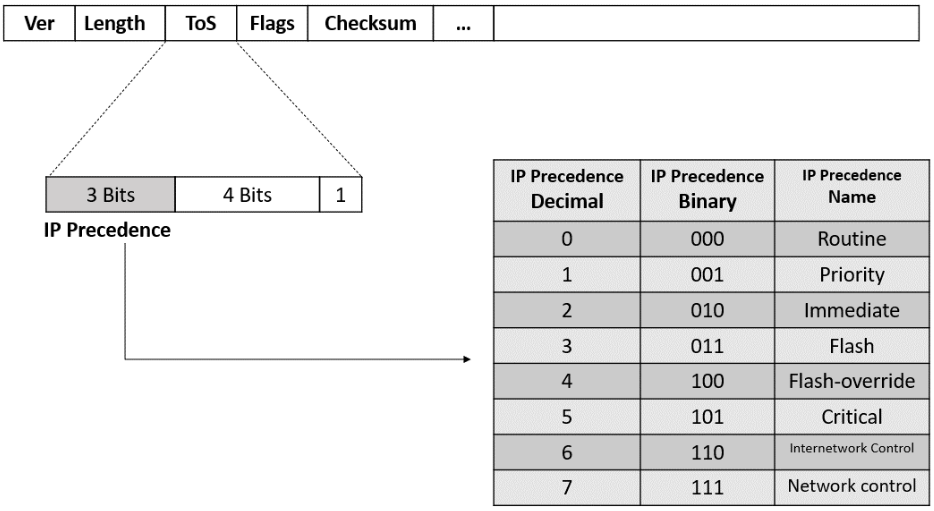


Figure 3: QoS Classification & Marking, , source: author

Three routers are shown in the scenario Figure 3 above, Voice and data traffic travels across these routers. Our work in QoS, particularly in the DiffServ discussion, is founded on the idea of giving each traffic its own quality by separating it based on its requirements. It is impossible to offer a high standard of service if traffic is not classified. Prior to implementing the policy, we must first identify which traffic requires QoS. Once that is done, we can classify the traffic. So, the first step is traffic classification. In the Figure 3, marking is not used in the first scenario, only classification is implemented. Voice and data traffic comes in, the router classify and separates the traffic based on traffic components (source IP, destination IP, source port, destination port, Protocol) and realizes that the first traffic is Data and the second is Voice. Later, it gives the desired quality of service (low delay, jitter and packet loss) to the Voice and gives the desired quality (high bandwidth besides low delay, jitter and packet loss) to the data. Therefore, we first classified the traffic based on the traffic components and finally applied the policy. Again, these traffics reach the second router, the second router classifies the traffic based on the same components and provides services. But if we do a one-to-one classification for each packet on each intermediate device based on the traffic components, this process is very heavy, so we have another suggestion, one device should classify the traffic based on its components. Then mark the traffic, other devices in the traffic path are classified based on the Mark. Classification & Marking doesn’t provide QoS for traffic by itself, we just classified the traffic and then marks them, this does not provide QoS, but they can be the first step to provide further services. In other words, if we want to increase the bandwidth in the upcoming routers, we may do so by looking at the marking right away and allocating the necessary bandwidth for the traffic that is marked. In this way, classification and marking are only prerequisites rather than sources of QoS. The

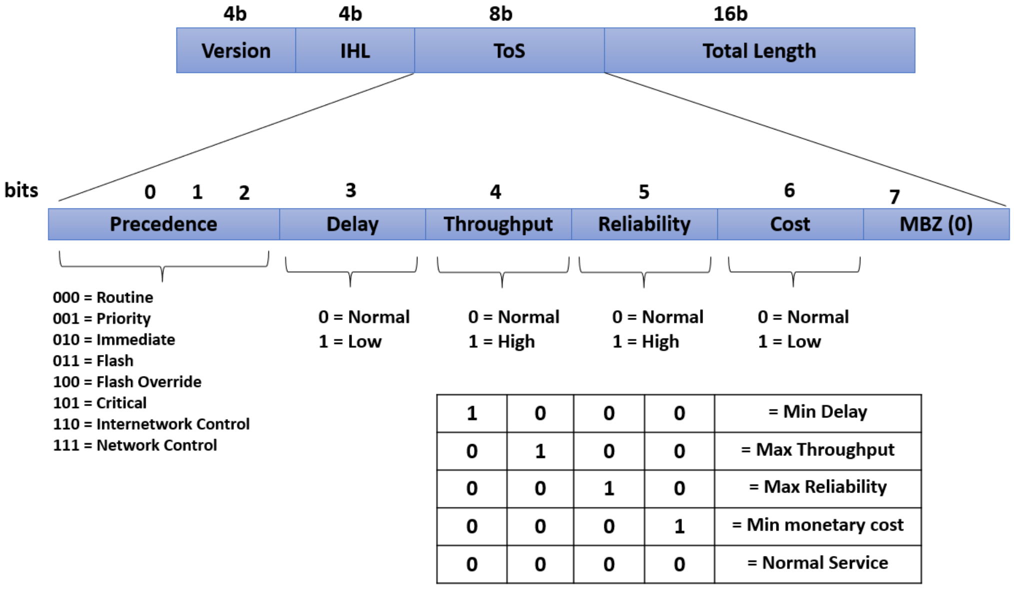
25 Type of Service (ToS) field in the L3 header and the Class of Service (CoS) field in the L2 Header L2 are where classification and marking are completed (9).



IP Precedence Decimal	IP Precedence Binary	IP Precedence Name
0	000	Routine
1	001	Priority
2	010	Immediate
3	011	Flash
4	100	Flash-override
5	101	Critical
6	110	Internetwork Control
7	111	Network control

Figure 4: IP Precedence (IPP) , source: (9)

There is an 8-bit field called ToS in the IP header. If ToS is zero, it means that we have no QoS and the network uses Best Effort model. ToS has been around since the beginning of the IP protocol discussion, i.e., since 1981 when RFC 791 introduced the IP protocol, it has been ToS and used since the beginning, but over time, the use of 8-bit ToS has changed. In the beginning, only the first 3 bits (---XXXXXX) IP Precedence (IPP) were used in the first version of the ToS. As we can see in Figure 4, by setting these 3 bits in the range 0-7, we are prioritizing the packet. As mentioned above, the ToS part has changed over time, one of the reasons why IPP is not enough for marking is that IPP can take numbers 0-7, that is, we can only recognize 8 qualities and it is only Forwarding quality, not Dropping in traffic congestion. The next change was that apart from the first three bits, it used the next four bits as well, Figure 5. This change was unpopular and was later removed. These four bits are named as follows: Delay, Throughput, Reliability and Cost. The 8th bit was practically unused and called Must be Zero (MBZ). These four bits are zero by default, but by setting these bits, it is determined what special requirements this particular traffic has. For example, the fourth bit is set for delay-sensitive voice traffic. This method was not very ideal, it was presented in a very short time and withdrawn very quickly (8).



1	0	0	0	= Min Delay
0	1	0	0	= Max Throughput
0	0	1	0	= Max Reliability
0	0	0	1	= Min monetary cost
0	0	0	0	= Normal Service

Figure 5: Unpopular change in the use of ToS bits for traffic prioritization, source: (8)

Another change was made to the use of the ToS and has been maintained until now and appears to have made a fundamental change, Figure 6. In the new model proposed for DiffServ, the first 6 bits were named Differentiated Services Code Point (DSCP), which means that different codes are used for different services. First, the six bits are specified and the last two are not in used. Then the last two bits Explicit Congestion Notification (ECN) are also added, which are used in Congestion Avoidance, the 6 DSCP bits provides 64 states:

$$ToS = DS\ Field\ (Differentiated\ Service)$$

$$ToS = DS\ Field = \underset{(\boldsymbol{DSCP})}{xxxxxx}\quad \underset{(ECN)}{xx}$$

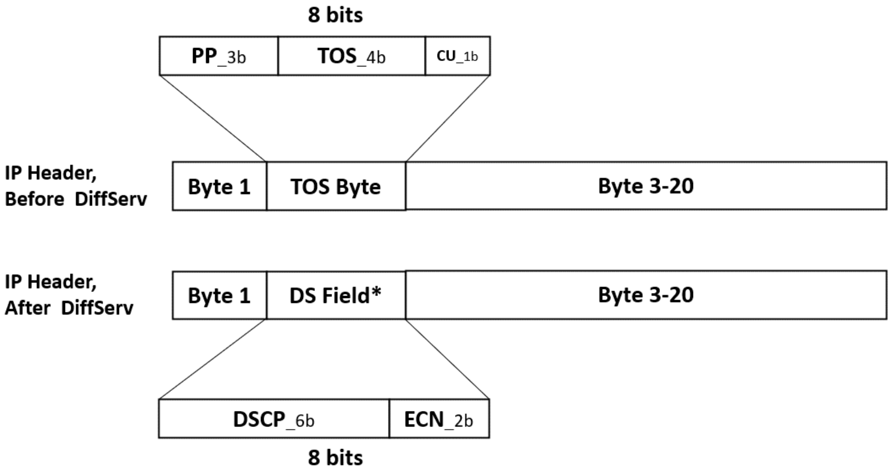


Figure 6: The new method to the use of the ToS, Differentiated Service (DS) Field, source: (8)

Both DSCP and IPP must be compatible with each other because there may still be devices on the network that do not actually support DSCP yet or for some reason are not yet configured and use IPP, so they must be compatible with each other. Therefore, it is recommended to set the DSCP values as follows (8):

DSCP Class Selector Names	Binary DSCP Values	IPP Binary Values	IPP Names
Default/CS0*	000000	000	Routine
CS1	001000	001	Priority
CS2	010000	010	Immediate
CS3	011000	011	Flash
CS4	100000	100	Flash Override
CS5	101000	101	Critical
CS6	110000	110	Internetwork Control
CS7	111000	111	Network Control

Table 8: Default and Class Selector DSCP Values, source: (8)

In 1999, one of the most famous Marking RFC was released, RFC 2597: Assured Forwarding (AF xy). The intention behind switching from IPP to DSCP was to have more states or classes; but, in order to maintain compatibility between IPP and DSCP, we are forced to employ DSCP values, which ultimately result in the same 8 classes (CS0-CS7). There is one area where our opinions on quality diverge somewhat. We believe that quality translates into faster forwarding. For example, if a router has two queues, each holding the same amount of packets, the queue with the greatest priority will transmit the traffic more quickly. We see the quality in fast forwarding, but the quality is not just fast-forwarding. It is true that faster forwarding is a quality dimension, usually any packet with higher IPP or CS is delivered faster. But a queue always has another concern besides forwarding, and that is dropping. Packets must be dropped whenever the queue is full, but a packet with a higher forwarding priority does not necessarily mean it has a higher dropping priority. It is not possible to give priority to both forwarding and dropping at the same time with IPP and DSCP, because the first three bits just set the priority of forwarding and because of their compatibility, the second 3 bits should not be used. In RFC 2597, the use of 6 bits of DSCP was suggested as follows: the first three bits are for forwarding priority, the next two bits are for dropping priority, and the last bit is not used (9).

DSCP: *FFF DD 0*

If a packet arrives with the first three forwarding bits bigger, that packet is sent earlier, but if the packet with the next two bits bigger, it must be dropped earlier. Therefore, this new method is called Assured Forwarding or AF xy, where x represents the forwarding priority and y represents the dropping priority (9):

AF *XY*

X: 1, 2, 3, 4 Forward Precedence

Y: 1, 2, 3 Dro Precedence

AF xy	DSCP [XXX YY0]	AF xy	DSCP [XXX YY0]	AF xy	DSCP [XXX YY0]
AF11	[001 010]	AF12	[001 100]	AF13	[001 110]
AF21	[010 010]	AF22	[010 100]	AF23	[010 110]
AF31	[011 010]	AF32	[011 100]	AF33	[011 110]
AF41	[100 010]	AF42	[100 100]	AF43	[100 110]

Table 9: Assured Forwarding (AF xy), source: (9)

In 1999, RFC 2598 introduced another code along with AF. Explicited Forwarding (EF), with decimal value 46 and binary value 101110, is the DSCP value established for AF53. To reduce latency, jitter, and packet loss, we often label the traffic—which is voice packets—with EF. Afs and EF can be used to have the dropping priority, but the codes that were used for the marking are CS codes, which are essentially equal to IPP (9).

3.4.1 Layer_2 Classification & Marking (C&M):

As previously mentioned, marking in the IP header or layer 3 header is our top priority when it comes to marking. The explanation is because the layer 3 header is actually preserved when traffic is routed from one router to another., so if we have done the marking in the layer 3 header, there will be no change in the marking, but As for the layer 2 header, the header is changed when forwarded from one switch to another. So, it is not very useful to mark the traffic in layer 2 header. But the problem is that we have some equipment in our network that may not understand the L3 header. So it can also be marked in the L2 header but with conditions. Usually, the l2 header does not have the ability to mark or a field to mark. It's best to classify and mark data as close to to the source of the traffic as possible.

Setting trust boundaries is important when C&M data on a local network (LAN). To ensure QoS marking transparency, it's essential to map between Layer 2 and Layer 3 classification schemes. Cisco Catalyst switches are well-equipped with classification and marking capabilities, making them optimal for executing these vital QoS functions. Workgroup switches typically rely on DSCP and CoS for classification and marking, with compatibility with IP precedence achievable due to DiffServ's backward compatibility. Additionally, Layer 2 CoS values can only be transmitted on ports configured as ISL or 802.1Q trunks (7).

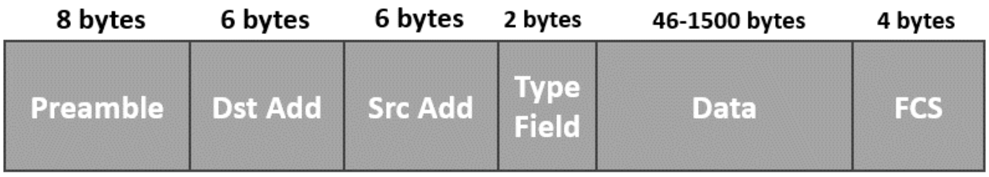


Figure 7: Layer 2 (Ethernet) header, source: (7)

But when the packet is tagged, one field is used for marking, which is called User Priority or CoS (3bits) (7):

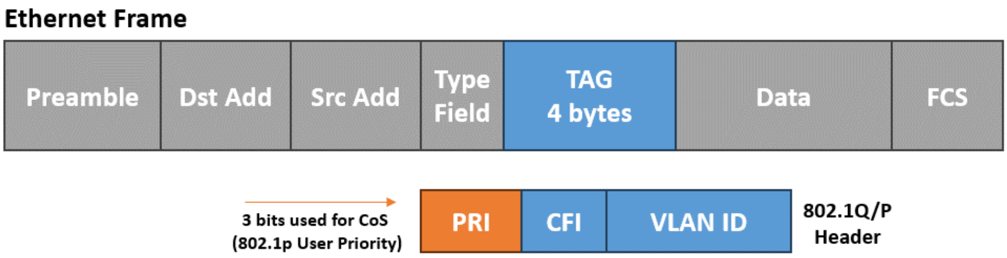


Figure 8: 802.1Q/P Header, source: (7)

The 802.1p User Priority field is also called Class of Service, different types of traffic are assigned different CoS values (Only forwarding precedence) (7):

CoS	Typical Application
7	Reserved
6	Reserved
5	Voice Bearer
4	Videoconferencing
3	Call Signalling

2	High-Priority Data
1	Medium-Priority Data
0	Best-Effort Data

Table 10: Type of CoS and their typical applications, source: (7)

What is usually done is that when the traffic arrives at the router, it maps these CoS to DSCPs, for example, CoS5 is mapped to EF.

3.5 Congestion Management

During times of no congestion, QoS is not needed. Packets transmitted FIFO at line-rate of egress interface (TX-Ring). There are two types of congestion: 1) Egress Congestion: Packets forwarded to egress interface faster than TX-Ring can handle them, 2) Ingress Congestion: Packets arrive on multiple ingress interfaces faster than Forwarding Engine can process them. The congestion causes delay, jitter and drop. When egress traffic cannot immediately be transmitted, it is placed in an egress queue. Multiple related egress queues that are distinguished by priority may be present on a single egress interface. Control over which classified traffic is inserted into each of these queues is made possible by QoS features specifically intended for queuing. Moreover, traffic in queues may be proactively dropped to make place for higher-priority traffic. A QoS mechanism known as congestion management controls queues that keep packets until an egress path is made available to them or the router releases the resources they require. But the congestion management is more than a simple idea, to understand it better and to understand how it works, we need to look at the processes that take place inside the device. In fact, the packets are stored in the device memory (RAM) and when there is free space on the interface to send, the CPU is requested to send the next packet with the help of the pointers. Therefore, we consider the queue size as the number of packets (count). For example, the queue size of an interface is 40, which means it can request the device memory to re-store 40 packets of 100-byte packets, or 40 packets of 1500-byte packets. QoS is not modifying the quantity of physical buffers allocated to an interface, or a particular-sized packet, instead you are taking the existing buffers that have already been defined as interface queues, and modifying how packets are treated when INSIDE those queues. Configuration of buffers is not normally a part of QoS, Buffer configuration would involve modifying the quantity of buffers allowed for particular sized packets. Congestion management features allow you to control congestion by determining the order in which packets are sent out an interface based on priorities assigned to those packets:Queues are created, packets are assigned to them depending on their categorization, and packets inside queues are selectively dropped when they approach pre-established criteria. The packets in a queue are scheduled for transmission. Routing and queuing behaviors are usually impacted by QoS queuing features (like WFQ) on routers. Queuing and scheduling can be distinct characteristics or functions on switches. Among the functions of scheduling is traffic shaping (7):

3.5.1 Software and Hardware Queue

Software Queue is the same queue that we place in the device's memory (RAM) and gradually the packages are delivered to the Interface through the CPU. Due to high CPU usage, the CPU cannot deliver packets, so there is a back-up queue with FIFO logic on the interface, which is called a hardware queue. So, all the configuration happens in the software queue, and we have no control over the hardware queue, it is just so that we can use 100% bandwidth (13).

When the hardware queue becomes empty, the packets are directly placed in the hardware queue, which is not under the admin control, but as soon as the hardware queue is full, the software queue starts filling up, which is under the admin control. When we want to use queuing tools for large amounts of data, we need to reduce the size of the hardware queue so that more data enters the software queue first. Software Queue is the same queue that we place in the device's memory (RAM) and gradually the packages are delivered to the Interface through the CPU. Due to high CPU usage, the CPU cannot deliver packets, so there is a back-up queue with FIFO logic on the interface, which is called a hardware queue. So, all the configuration happens in the software queue, and we have no control over the hardware queue, it is just so that we can use 100% bandwidth (8).

3.5.2 Queuing Methods

The fundamental reason we need Congestion Management is because: 1) By default, queues are configured for FIFO, 2) Incoming bursts can be bad causing congestion of queues. FIFO = no control over the order of which of those packets held back in the queue will be transmitted, 3) Congestion management techniques provide some control of the order-of-transmission (9).

3.5.2.1 FIFO

On most router interfaces and all switches without QoS configured, the default queuing technique is First In First Out (FIFO). In computer networking, the FIFO approach is comparable to a line at a store. Data creates a line when it gets to a device such as a router. The first piece of information received is the first to be transmitted. Although it's an easy and equitable method of handling data, if the line gets too long, there may occasionally be delays (7).

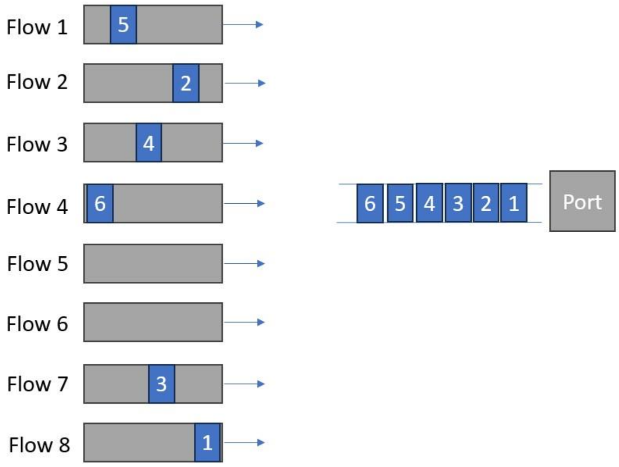


Figure 9: FIFO Queueing method, source: author

Classification	**No Classification (N/A)**
Drop decision	Tail dropping
Maximum number of Queues	1
Maximum Queue length	Depends on IOS
Scheduling inside Queue	FIFO algorithm
Scheule logic between Queues	N/A

Table 11: Queueing method comparison points in FIFO method, source: (7)

3.5.2.2 Weighted Fair Queue (WFQ)

In order to schedule "important" traffic more frequently than other forms of traffic, fair queuing divides traffic into distinct queues. There are various varieties of FQ: 1) Flow-Based

WFQ; 2) CBWFQ; 3) LLQ (assigns weights to traffic based on IP Precedence). One of the factors that sets weighted fair queuing and class-based weighted fair queuing apart from other similar features is the way in which these queues are distributed. The least configuration is needed for the Weighted Fair Queue (WFQ) approach, which is typically used by default. For an interface with less bandwidth, we typically utilize this technique by default. However, if we want to configure QoS on the interface properly, we need to employ the following ways. This method shares the bandwidth by looking at how important each packet of data is (bigger IPP) and how big it is. Packets that are very important or small get 36 more bandwidth. For example, IP phones (VoIP) don't use much data and are important, so they get more bandwidth. This way, important or small packets can move faster through the network (7).

Classification	**Flow-based**
Drop decision	Modified Trail Drop
Maximum number of Queues	4096 Q
Maximum Queue length	Configurable
Scheduling inside Queue	FIFO algorithm
Scheule logic between Queues	Lowest Serial Number = F (IPP, Length)

Table 12: Queueing method comparison points in WFQ method, source: (7)

3.5.2.3 Class-Based Weighted Fair Queuing (CBWFQ)

WFQ is extended by Class-Based Weighted Fair Queuing (CBWFQ), which lets you, the user, establish user-based traffic classes (rather than relying solely on flow-criteria as WFQ does). In this manner, several traffic flows that Flow-Based FQ would typically assign to separate queues are instead combined into one queue. The primary advantage is that you can now decide how much BW is minimum for each class-based flow. CBWFQ: A Variety of Features Consolidated into One: A) Queuing: This method groups traffic according to MQC support, B) Dropping: Uses Tail Drop or WRED (adjustable per queue); C) Scheduling: Shared Round Robin based on bandwidth parameters by default, FIFO inside a single queue (no packet reordering once inserted). Up to 64 queues can be manually configured per interface with CBWFQ; the depth and size of each queue can be changed. A tail drop will occur when a queue supporting a specific class reaches its maximum depth (policy-map command). When a class is given a bandwidth, that's turns CBWFQ (9).

Classification	**Same as CB Marking**
Drop decision	by default: Tail-drop; by config: WRED
Maximum number of Queues	64 Q
Maximum Queue length	Depends on IOS and its memory, flexible
Scheduling inside Queue	FIFO, WFQ
Scheule logic between Queues	Process is not published, Result: Each Queue receives configured percentage BW or configured specific BW.

Table 13: Queueing method comparison points in CBWFQ method, source: (9)

3.5.2.4 Low Latency Queue (LLQ)

Another name for Low-Latency Queuing (LLQ) is PQ/CBWFQ. This is a CBWFQ "add-on" feature. It enables you to create a Priority Queue from one or more of your defined classes. It is identified by Cisco IOS through the use of the "priority" command in a Class&Map. LLQ was created exclusively for voice traffic. This eliminates jitter and allows it to be scheduled or handled before any other traffic. You can send ANY type of traffic into the LLQ using CBWFQ/LLQ. It is necessary to provide the LLQ's bandwidth, which is its MAXIMUM bandwidth during periods of congestion (7).

3.5.2.5 Weighted Round Robin (WRR)

The weight provided to each queue in the WRR (Weighted Round Robin) algorithm determines how much bandwidth is allotted to each queue; higher weights indicate more packets being transmitted. This indicates that a greater portion of the bandwidth is allocated to queues with higher weights in WRR mode. The specified percentages for each queue determine how many packets are served from each during a visit. For instance, in a scenario where all queues operate under WRR with default weights, if there's congestion and all queues are saturated, the bandwidth allocation would be as follows: queue 1 receives 1/15 of the bandwidth, queue 2 receives 2/15, queue 3 receives 4/15, and queue 4 receives 8/15. It's noteworthy that the WRR algorithm implemented in the device differs from the standard

WRR; it utilizes Shaped Deficit WRR (SDWRR) instead of the more typical Deficit WRR (DWRR) (7).

3.6 Congestion Avoidance

Term used to define a set of features that attempt to prevent queues from becoming congested. The Congestion Avoidance Can be done in three places (depending on hardware platform): A) Ingress interface queue, prior to lookup by forwarding engine, B) At the forwarding engine (policing), C) Within the egress queue (drop thresholds). Typical CA methods within queues assume that most traffic is adaptive to traffic drops (TCP), Congestion without QoS causes Tail Drop which can lead to Global Synchronization (15).

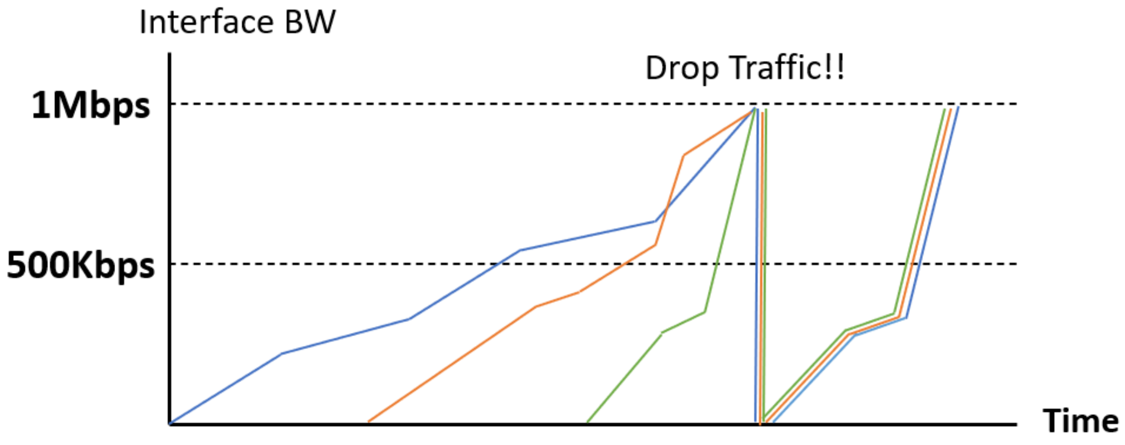


Figure 10: Global Synchronization, source: (15)

The above picture describes the Global Synchronization! User complains that I’ve got a 1 Mbps circuit and I should be transmitting at one meg or real close to it. But I’m only getting half of that. Where is all that extra bandwidth going? Where is my wasted bandwidth? That’s because all these streams are synchronized. They are going down at the same time, up at the same time, down at the same time and so you’ve got all this space right here that’s repeating itself that’s unused. And so over time, the average is far less than what the actual rate could be of the interface. So that is called Global synchronization (9).

Congestion Avoidance within queues includes the following:

3.6.1 WTD (Weighted Tail Drop):

Tail drop is the standard packet-dropping technique, which treats all traffic identically and doesn't give service classes any preference. When output queue buffers full up in tail drop, all packets—regardless of priority— are lost until the congestion passes. This approach is not the greatest for TCP traffic since it can cause TCP global synchronization, which drastically lowers connection utilization. Avoiding tail drop is advised to prevent such issues. The majority of switching hardware uses this method, along with DSCP-to-Threshold Mappings and programmable thresholds (8).

A simple queue management technique that doesn't modify anything in the queue memory is the Drop Tail approach. On the other hand, the RED (Random Early 42 Detection) algorithm is an active queue management technique that was among the first major methods for actively managing queues (17).

Average Queue Depth Versus Thresholds	Action	Name
Average < minimum threshold	No Packets dropped	No Drop
Minimum threshold < average depth < maximum threshold	A percentage of packets dropped. Drop percentage increases from 0 to a maximum percent as the average depth moves from the minimum threshold to the maximum.	Random Drop
Average depth > maximum threshold	All new packets discarded similar to tail dropping	Full Drop

Table 14: Three Categories of when RED will Discard Packets and How Many, source: (17)

3.6.2 WRED (Weighted Random Early Discard):

The cisco implementation of RED is WRED. It Can be implemented in both Switches and Routers. Random packet drops start at the min- threshold, increase in a linear format until max-threshold is reached. After max-threshold is reached, WRED drops 100% of all subsequent packets 43 received. The RED algorithm detects network congestion by calculating a weighted average of the queue length. It marks or discards incoming batches based on predefined lower and upper threshold values. However, a drawback is its inability to prioritize different types of traffic. To address this, the WRED (Weighted RED) algorithm, an extension of RED, was introduced. WRED allows multiple thresholds for a queue, assigning each to a specific traffic class. This way, the WRED method considers priority before dropping a packet, enhancing traffic management (7).

3.6.3 Explicit Congestion (ECN):

Congestion avoidance methods are used to keep an eye on how much traffic is flowing through a network to try and prevent it from getting too 44 congested at certain points. Instead of just dropping packets when things get busy, more advanced techniques are used to manage the flow. By adding explicit congestion notification (ECN) to IP, routers have another way to let other routers know if there's too much traffic. ECN works by marking packets when the average queue length reaches a certain limit instead of dropping them outright. It introduces two flow control bits, the ECT bit and the CE bit, as additions to the DiffServ field. ECN serves as an extension to WRED, eliminating the necessity to depend solely on packet loss as an indicator of congestion. To configure ECN on Cisco IOS routers, you need to use MQC (Modular QoS Command-Line Interface) and specifically the command "random-detect ecn". Information about ECN marking is shown with the "show policy-map interface" command only if ECN is enabled on the interface (17).

3.7 Traffic Conditioning (Shaping & Policing)

Shaping is like pruning a shrub and the policing use the same word as police, and the duty of the police on the highway is to control the car so that it does not commit a violation and to fine that car if a violation occurs. Between ISP and Customer there is a pre-defined, contracted rate (called CIR). ISP will police ingress traffic, traffic that is non-conforming is caught by policer and: 1) Dropped and 2) Marked-down. Customer typically doesn't want any traffic dropped (delay is better than drops), shaping is done on egress interface leading to ISP. The purpose of shaping is to make full use of the bandwidth and queue additional traffic. What it does is, if additional traffic comes in, it queues the traffic and gradually sends that traffic. The difference in policing is that we don't queue and drop additional traffic. Both Shaping and Policing measure rate of traffic against a configured rate called the Committed Information Rate (CIR), Shaper buffer excess traffic but Policers typically excess drop traffic. On routers (18)(20):

- Policer can be applied on ingress or egress interfaces...but usually done on ingress.
- Typically, ISPs will enforce contracts with Policers.
- Shapers usually done on egress connection to the ISP.

Most Switches do support some kind of policing, but not shaping.

3.7.1 Shaping

The shaper performing the shaping measures the contract rate on outgoing traffic, queues the traffic if it exceeds the CR, and gradually drains the queue. Shaping is actually used between the customer and the provider, usually because there is a policer on the provider side, and if we send more traffic than the contract rate, the provider actually drops it. Shapers temporarily hold and slow down egress traffic rates that exceed the desired rate until they fall below the defined rate. If the egress traffic rate is already below the desired rate, it is sent immediately. In the outgoing direction, only packets that exceed the defined profile are queued. They remain in the queue until the buffer reaches its capacity. This buffering approach helps reduce the need for TCP retransmissions. Unlike other methods, this one doesn't support marking or re-marking packets. However, it does facilitate interaction with Frame Relay congestion signals, providing more control over network traffic (9)(19).

3.7.2 Policing

When incoming or outgoing traffic exceeds a predetermined traffic rate, policers drop or re-mark (give a lower priority). For example, excess traffic that was originally flagged as AFx1 could be re-marked as AFx2. In both incoming and outgoing traffic, packets that don't meet the defined profile criteria are dropped. This dropping leads to TCP retransmissions, impacting network performance. The system offers options for packet marking or remarking to manage traffic effectively. Furthermore, it consumes fewer buffers compared to shaping methods, reducing resource utilization. Shaping, on the other hand, necessitates an additional queuing system, adding complexity and overhead to network management (8) (19).

3.8 QoS in LAN

Since the CPU cannot see the packet, switches implement QoS functions in hardware; so, switches have fewer QoS functionalities. Different QoS characteristics are supported by different switch hardware systems. Upon global QoS enabling (24):

- An internal QOS value known as "internal DSCP" was assigned to every incoming frame.
- By default all ports are "untrusted" = internal DSCP 0.
- Whatever the internal DSCP is...that same value will copy into the TOS byte of egress frames.
- On ports connected to critical equipment (i.e. IP Phones) best practice is to set "mls qos trust dscp" on switchport.

When QoS is enabled globally, a switch will assign an internal "QoS Label" to every frame. Every frame must be placed into some kind of egress transmission queue, which is dictated by this internal QoS Label. Ethernet carries more than just IP. So even frames carrying only L2 PDUs must be assigned a QoS Label. This internal QoS Label is frequently called, "internal DSCP" (24).

The internal DSCP value is matched upon when QoS actions are specified that depend on DSCP classification. The ingress interface's trust setting governs this internal DSCP configuration. Switches with QoS rely on interface "trust" settings to calculate internal DSCP. 10 G, 40 G, and 100 G of bandwidth can be employed in a LAN network.…. We use QoS when there is lack of bandwidth, because the source of all the issues that occur is bandwidth, and if we have enough bandwidth, we don't need QoS, but we always have lack 50 of bandwidth, so we always need QoS. In the high-speed links, Serialization delay, queuing delay and forwarding delay are close to zero and there is enough bandwidth for all applications, but no matter how much we increase the bandwidth of a network, we still face problems that limit the bandwidth. We need bandwidth because there are some applications that will consume whatever bandwidth we allocate them (9).

4. Practical Part

The practical part consists of three parts: a) Network Simulation, b) EVE-NG setup, c) QoS scenarios. The first part distinguishes between the two terms simulation and emulation and describes the most popular emulators used to emulate our practical part. The second part describes how to prepare and set up the laboratory for the deployment of the third part. The third part describes and tests different scenarios of QoS implementation in a company's network, **because of the absence of compatibility with other vendors in simulation environments, I selected Cisco and executed all scenarios using Cisco equipment (IOL/U)**.

4.1 Network Simulation

Although "simulation" and "emulation" of a system are sometimes used synonymously, they have distinct meanings in technical language: When something is simulated, it indicates that while the implementation of one system is entirely distinct, it functions similarly to another. (26). While the basic operation of the system is replicated in a simulation, not all of the simulation's rules remain adhered to. With the simulator, we can gain a decent understanding of how a system functions. The main network simulators are extremely helpful; these include Cisco Packet Tracer, OPNET, and Network Simulator (NS). On the other hand, system emulation attempts to replicate every feature of a system, even to the point of creating a perfect duplicate of the system. Well-known programmes that assist emulators in graphical network simulation are GNS-3 and EVE-NG. This thesis focuses on emulating QoS scenarios rather than simulating them. Emulators provide a more realistic experience than network simulators since they support real device command lines. This chapter will discuss several emulators and evaluate how to use EVE-NG environments to simulate quality of service issues (30)(28).

4.1.1 Emulators

A hardware or software emulator allows one computer system (the host) to use the tools, peripherals, and software of another computer system (the guest). As a result, the host acts more like the visitor (27). These common emulators are supported by EVE-NG:

- **Dynamips:** This Cisco router emulator supports 1700, 2600, 2691, 3600, 3725, 3745, and 7200 platforms, among others. It is compatible with Windows, Linux, and Mac OS X and operates by booting a real Cisco IOS image straight (29).

- **QEMU:** This open-source emulator features virtualized modes in addition to a machine emulation. Qemu is used by several contemporary appliances, such as Cisco's VIRL images. Nested virtualization is necessary for Qemu and IOL/IOU, however it is not supported by Windows and needs to be run on Linux (32).

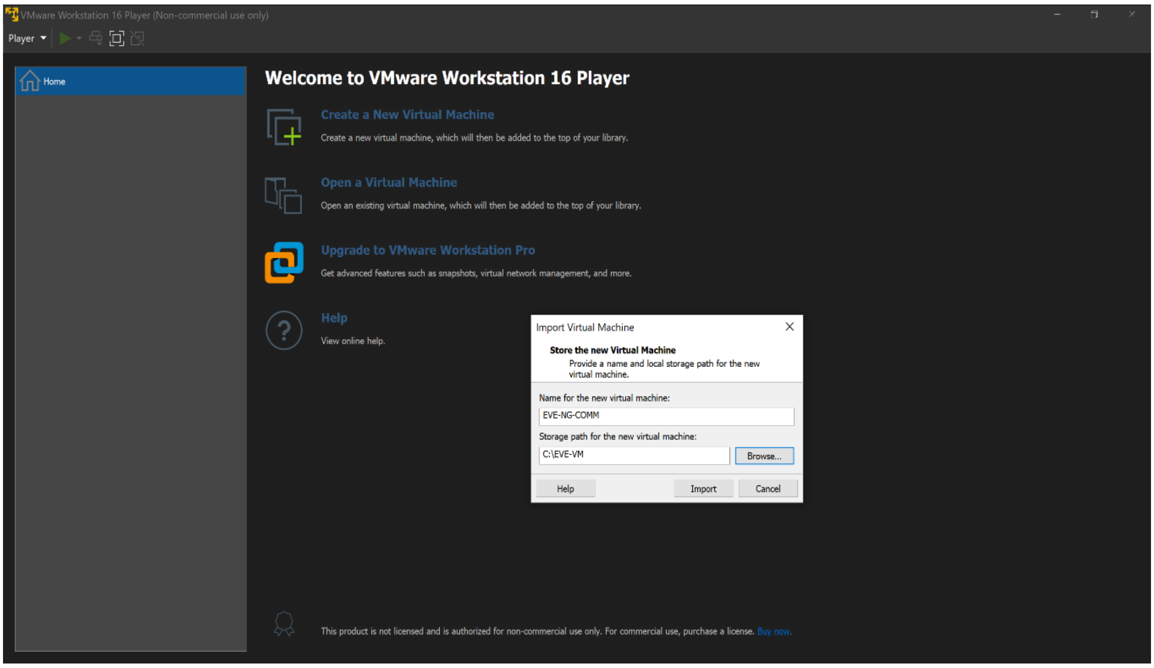
- **IOL/IOU:** An internal emulator from Cisco is called IOS on Linux (IOL) or IOS on Unix (IOU). The Unix version is designed for Sparc architecture, whereas the Linux version is optimised for i386 architecture. It can only be used by authorised customers or Cisco employees with an official licence that can be obtained online or from Cisco. It is not advised to use IOL/IOU images that you find online since they may include problems. These images are not formally issued by Cisco. It is a reasonable option, though, as it does not require a lot of CPU or memory resources and supports both L2 and L3 Cisco images (30).

- **vPC:** It's a simple, GUI-less personal computer emulator that can only execute IP configuration, traceroute, and ping commands. IP addresses can be assigned using DHCP (30).

4.2 Setting Up the EVE-NG

EVE-NG is the first clientless network emulation software, allowing it to run in a fully isolated environment. It comes in three versions: Community, Professional, and Learning Center (31). The latter two require a license purchase. All versions support the emulators mentioned in section 4.1.1. The virtual machine EVE-NG is available as an ISO and OVF file. It is officially supported on Google Cloud and can be installed on actual hardware or on a hypervisor.

Hypervisor Install: On virtual machine platforms, EVE-NG can be installed by importing its ISO or OVF file. Workstation (14.0 or later), Fusion (8 or later), Player (14.0 or later), and ESXi (6.0 or later) are among the VMware products that it officially supports.



VMware Workstation 16 Player (Non-commercial use only)

Welcome to VMware Workstation 16 Player

Create a New Virtual Machine

Create a new virtual machine, which will then be added to the top of your library.

Open a Virtual Machine

Open an existing virtual machine, which will then be added to the top of your library.

Upgrade to VMware Workstation Pro

Get advanced features such as snapshots, virtual network management, and more.

Help

View online help.

Import Virtual Machine

Store the new Virtual Machine

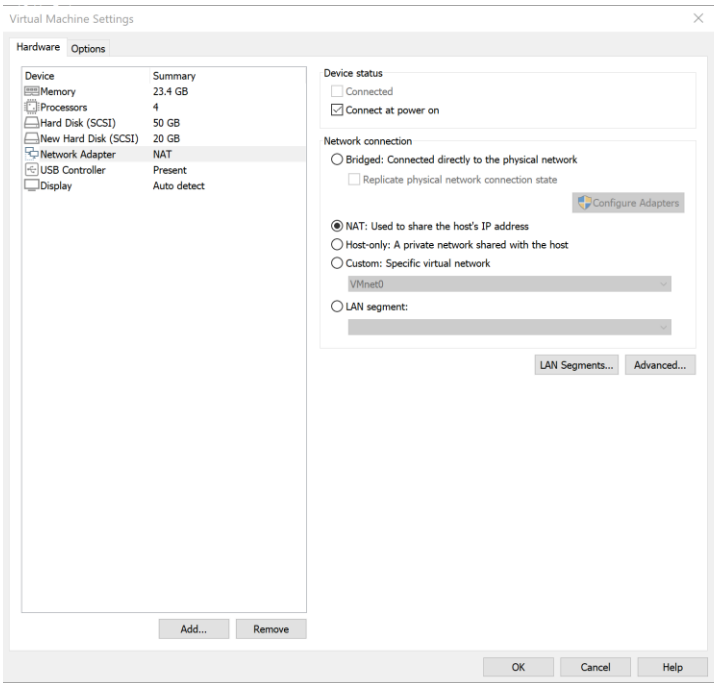
Provide a name and local storage path for the new virtual machine.

Name for the new virtual machine: EVE-NG-COMM

Storage path for the new virtual machine: C:\EVE-VM

This product is not licensed and is authorized for non-commercial use only. For commercial use, purchase a license. Buy now.

Figure 11: EVE-NG Hypervisor Installation, source: author



Virtual Machine Settings

Device	Summary
Memory	23.4 GB
Processors	4
Hard Disk (SCSI)	50 GB
New Hard Disk (SCSI)	20 GB
Network Adapter	NAT
USB Controller	Present
Display	Auto detect

Device status

□ Connected

☑ Connect at power on

Network connection

○ Bridged: Connected directly to the physical network

□ Replicate physical network connection state

● NAT: Used to share the host's IP address

○ Host-only: A private network shared with the host

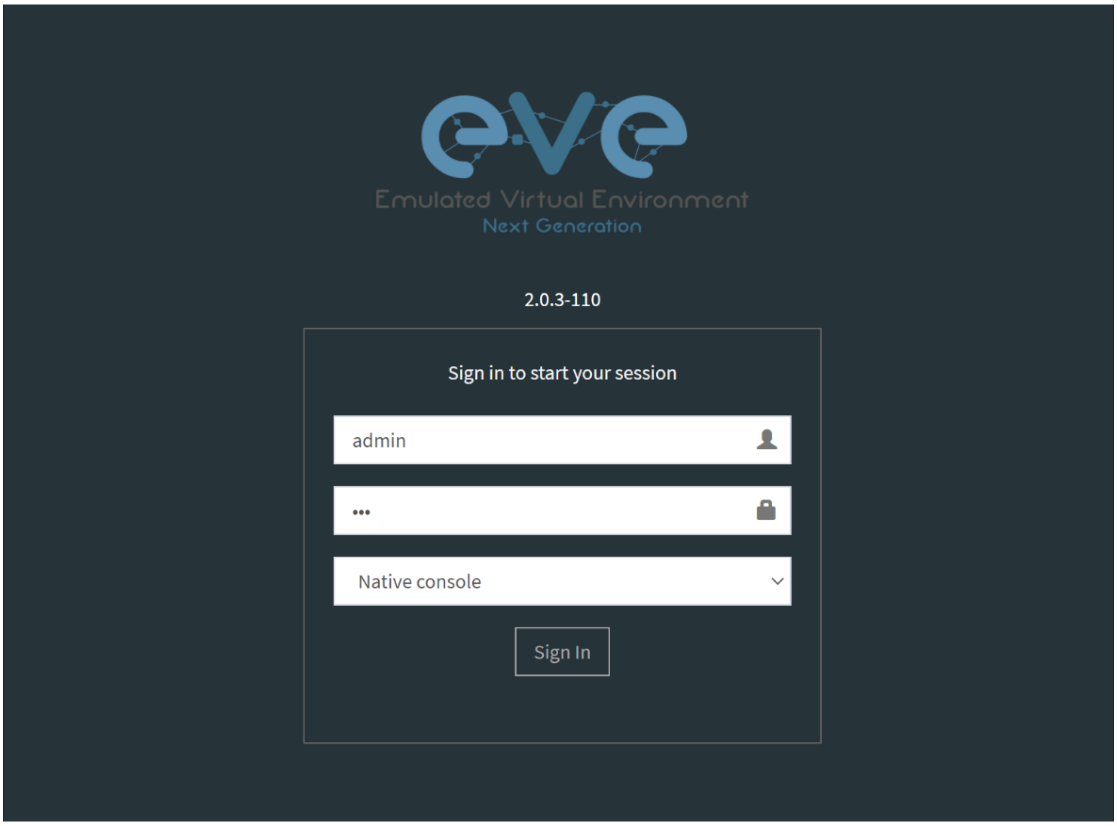
○ Custom: Specific virtual network

VMnet0

○ LAN segment:

Figure 12: Virtual Machine Settings for EVE-NG Installation, source: author

EVEN-NG graphical environment: With the provided management IP address (given via DHCP or static during installation), one can access the EVE-NG GUI via a web browser. The two console types it offers are: A) Native console, which makes use of programmes like Putty and UltraVNC; and B) HTML5 console, which provides a clientless solution that can be accessed directly through the browser without the need for terminal constraints like SecureCRT or Telnet.



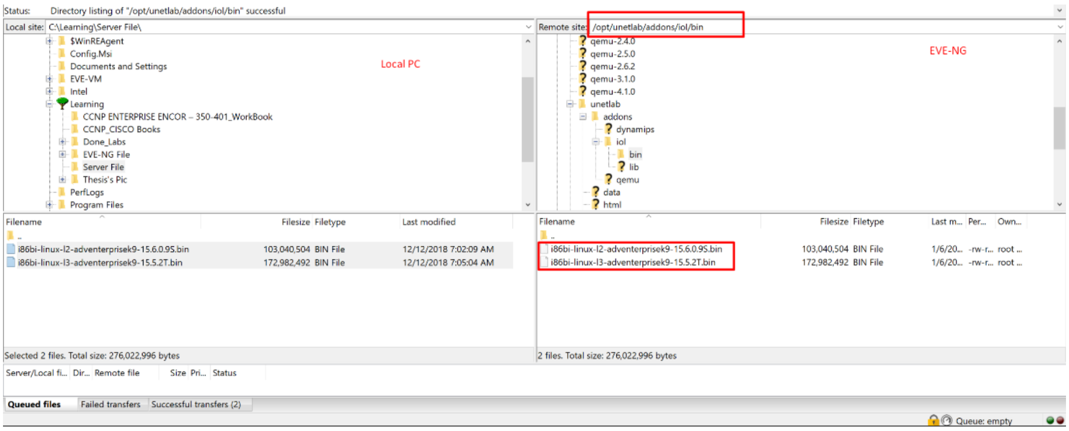
admin

•••

Native console

Figure 13: EVE-NG Graphical Environment, source: author

Use an FTP client programme like WinSCP or FileZilla to log in to the server using the management IP address in order to upload emulator images to the EVE-NG server. Make sure that every emulator is uploaded to the appropriate directory:


Status: Directory listing of "/opt/unetlab/addons/iol/bin" successful

Local site: C:\Learning\Server File\

Remote site: /opt/unetlab/addons/iol/bin

Filename	Filesize	Filetype	Last modified
..			
i86bi-linux-l2-adventerprisek9-15.6.0.9S.bin	103,040,504	BIN File	12/12/2018 7:02:09 AM
i86bi-linux-l3-adventerprisek9-15.5.2T.bin	172,982,492	BIN File	12/12/2018 7:05:04 AM

Selected 2 files. Total size: 276,022,996 bytes

Filename	Filesize	Filetype	Last m...	Per...	Own...
..					
i86bi-linux-l2-adventerprisek9-15.6.0.9S.bin	103,040,504	BIN File	1/6/20...	-rw-r...	root ...
i86bi-linux-l3-adventerprisek9-15.5.2T.bin	172,982,492	BIN File	1/6/20...	-rw-r...	root ...

2 files. Total size: 276,022,996 bytes

Figure 14: Uploading Emulator Images into EVE-NG via FileZilla FTP client tool, source: author

Using the following two emulators, all required emulator images are obtained from the Internet:

4.2.1 Create a custom Windows host in EVE-NG

This thesis uses a Qemu running Windows 7_32-bit with a setup that includes IP Traffic – Test & Measurement, VLC Media Player, Linphone and FTP Server. To start, download the Windows ISO file in the format ((GetMyOS)Windows_7_Ultimate_X86_SP1_En_Aug_2020).

- Access the EVE-NG server and create a new image directory. For directory naming, use the EVE-NG documentation and begin with "win-". Recall that EVE-NG uses specific names to identify folders. Use an SSH client, such as Putty or SecureCRT, to connect to the EVE-NG server:

```
root@eve-ng:~#mkdir /opt/unetlab/addons/qemu/win-7-ENARSI/
```

- Upload the image file with FTP clients such as FileZilla or WinSCP to the newly formed directory. Make use of the following credentials: Host: 192.168.10.128, Password: eve, Username: root, Port: 22

```
root@eve-ng:/opt/unetlab/addons/qemu# ls -l
drwxr-xr-x 2 root root 4096 Jan 5 18:35 win-7-ENARSI
```

- Change the ISO file's name to cdrom.iso. Use this command to navigate to the specified directory (win-7-ENARSI):

```
root@eve-ng: cd /opt/unetlab/addons/qemu/win-7-ENARSI
```

- Rename the file with the mv command.

```
Mv \(GetMyOS\) Windows_7_Ultimate_X86_SP1_En_Aug_2018.iso cdrom.iso

root@eve-ng:/opt/unetlab/addons/qemu/win-7-ENARSI# ls
cdrom.iso
```

Generate a new virtual hard disk named virtioa.qcow2 within the image folder.

Verify the HDD image's name; for example, virtioa for the Windows host (Qemu emulator). The virtual disc storage is in the ".qcow2" format, with a 50G maximum capacity chosen.

```
root@eve-ng:/opt/unetlab/addons/qemu/win-7-ENARSI#
/opt/qemu/bin/qemu-img create -f qcow2 virtioa.qcow2 50G
```

- Open the browser and navigate to the server's IP address (192.168.10.128). Use the admin/eve credential to log into EVE-NG. Make a new lab, add a win-7-ENARSI host, and use the Management Cloud node that comes with EVE-NG to connect it to the internet (Home LAN). Launch the Win Node, then carry out the Windows 7 installation.

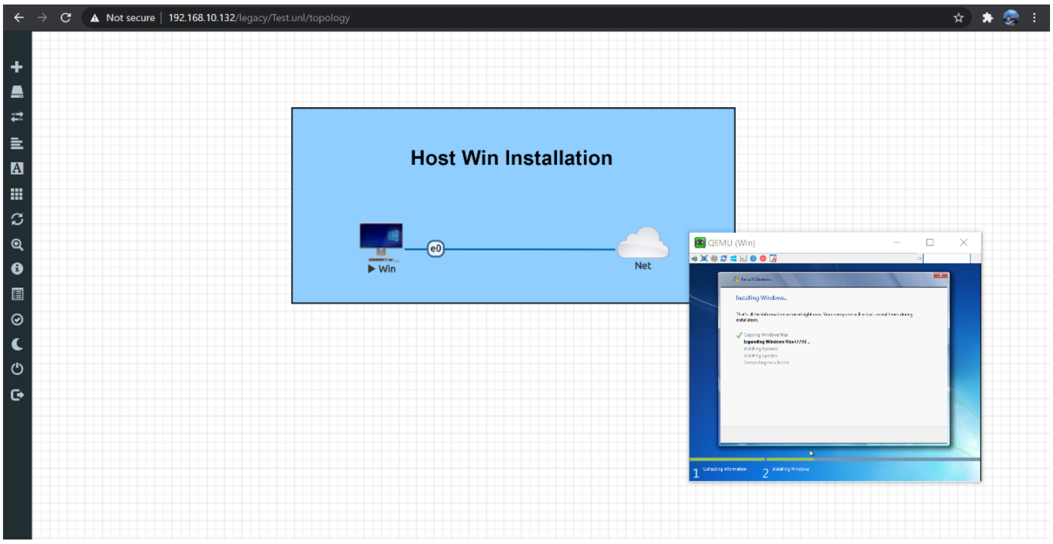


Figure 15: Host Windows Installation, source: author

- Install the lab-specific software and commit the changes to make it the default image going forward. Locate the image that has been installed:

```
root@eve-ng:/opt/unetlab/addons/qemu/win-7-ENARSI#cd
/opt/unetlab/tmp/0/bb3a63ec-ef76-4a1b-826e-c03730271385/1/
```

- Save the changes to the virtioa local repository.Qcow2.

```
root@eve-ng:/opt/unetlab/tmp/0/bb3a63ec-ef76-4a1b
826ec03730271385/1#/opt/qemu/bin/qemu-img commit virtioa.qcow2 Image
committed
```

- Ultimately, it is recommended to remove the cdrom.iso file.

```
root@eve-ng:/opt/unetlab/addons/qemu/win-7-ENARSI# ls
cdrom.iso virtioa.qcow2

root@eve-ng:/opt/unetlab/addons/qemu/win-7-ENARSI# rm -f cdrom.iso

root@eve-ng:/opt/unetlab/addons/qemu/win-7-ENARSI# ls
```

3.2.2 Upload IOL/IOU and generate a license Key

Several emulators, such as Dynamips and Qemu, were tried with Cisco IOS images to simulate QoS scenarios; however, they did not support all commands. As a result, IOL was selected because of its extensive command support. To create a license key and upload IOL images to an EVE-NG server, follow these instructions:

- First, upload the images to the /opt/unetlab/addons/iol/bin/ directory. For an expedient procedure, utilize FileZilla or WinSCP as your FTP clients.

  Verify whether images exist in the specified directory by running the ls command:

```
root@eve-ng:~# ls /opt/unetlab/addons/iol/bin/
i86bi-linux-l2-adventerprisek9-15.6.0.9S.bin
L2-ADVENTERPRISEK9-M-15.2-20150703.bin
L3-ADVENTERPRISEK9-M-15.4-2T.bin
```

- Execute the given command to set the IOL/U's permissions:

```
root@eve-ng:~# /opt/unetlab/wrappers//unl_wrapper -a fixpermissions
```

- To enable the functioning of IOU/IOL images, generate the IOU licensing key within EVE-NG by following the given instructions.

  - Navigate to the specified directory.

```
root@eve-ng:~# cd /opt/unetlab/addons/iol/bin
```

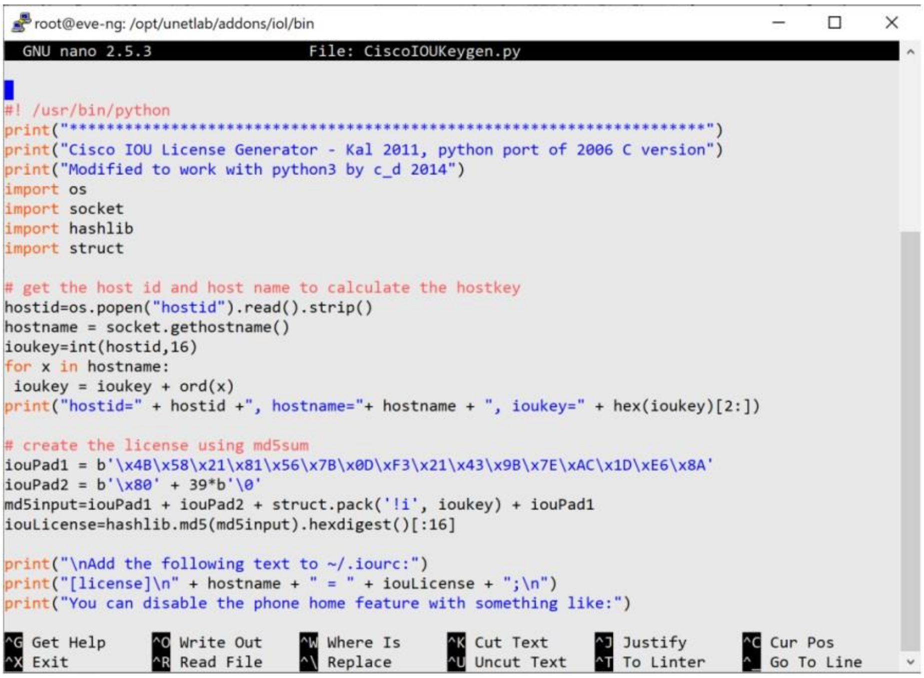
  - Create a new file with the name Cisco IOU keygen.

***root@eve-ng:/opt/unetlab/addons/iol/bin#** sudo touch CiscoIOUKeygen.py*

- Insert the provided script (which can be found online) into the file.

***root@eve-ng:/opt/unetlab/addons/iol/bin#** sudo nano CiscoIOUKeygen.py*

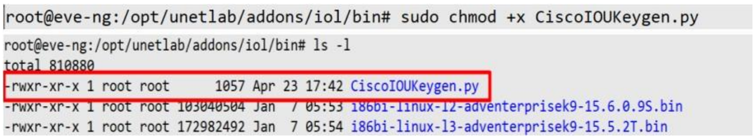
- Download the recommended script from the internet and follow with EVE-NG's instructions.



```
#! /usr/bin/python
print("***********************************************************************")
print("Cisco IOU License Generator - Kal 2011, python port of 2006 C version")
print("Modified to work with python3 by c_d 2014")
import os
import socket
import hashlib
import struct

# get the host id and host name to calculate the hostkey
hostid=os.popen("hostid").read().strip()
hostname = socket.gethostname()
ioukey=int(hostid,16)
for x in hostname:
 ioukey = ioukey + ord(x)
print("hostid=" + hostid +", hostname="+ hostname + ", ioukey=" + hex(ioukey)[2:])

# create the license using md5sum
iouPad1 = b'\x4B\x58\x21\x81\x56\x7B\x0D\xF3\x21\x43\x9B\x7E\xAC\x1D\xE6\x8A'
iouPad2 = b'\x80' + 39*b'\0'
md5input=iouPad1 + iouPad2 + struct.pack('!i', ioukey) + iouPad1
iouLicense=hashlib.md5(md5input).hexdigest()[:16]

print("\nAdd the following text to ~/.iourc:")
print("[license]\n" + hostname + " = " + iouLicense + ";\n")
print("You can disable the phone home feature with something like:")
```

Figure 16: IOU license generator script, source: author

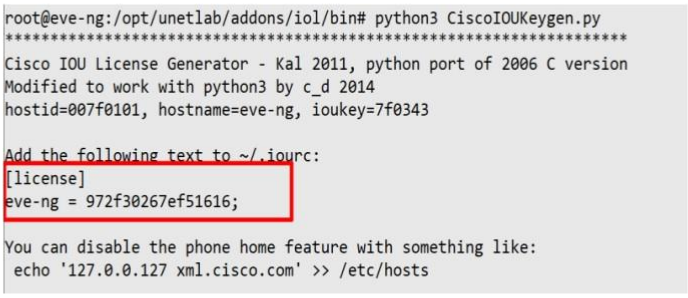
- Grant execution permission to the script file with the following command.



```
root@eve-ng:/opt/unetlab/addons/iol/bin# sudo chmod +x CiscoIOUKeygen.py
```

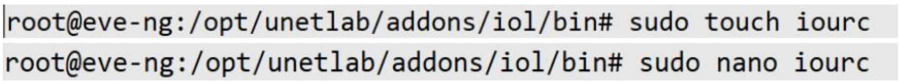
```
root@eve-ng:/opt/unetlab/addons/iol/bin# ls -l
total 810880
-rwxr-xr-x 1 root root      1057 Apr 23 17:42 CiscoIOUKeygen.py
-rwxr-xr-x 1 root root 103040504 Jan  7 05:53 i86bi-linux-l2-adventerprisek9-15.6.0.9S.bin
-rwxr-xr-x 1 root root 172982492 Jan  7 05:54 i86bi-linux-l3-adventerprisek9-15.5.2T.bin
```

- Execute the license generator script using the following command: **#python3 CiscoIOUKeygen.py**

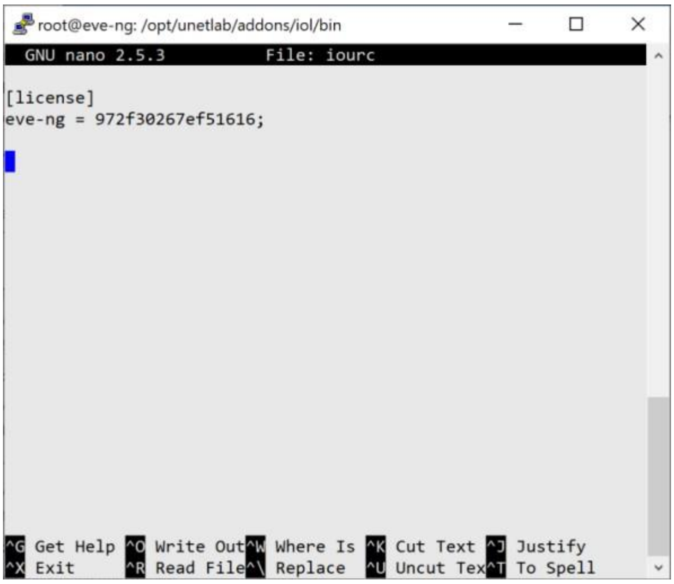


```
root@eve-ng:/opt/unetlab/addons/iol/bin# python3 CiscoIOUKeygen.py
*********************************************************************
Cisco IOU License Generator - Kal 2011, python port of 2006 C version
Modified to work with python3 by c_d 2014
hostid=007f0101, hostname=eve-ng, ioukey=7f0343

Add the following text to ~/.iourc:
[license]
eve-ng = 972f30267ef51616;

You can disable the phone home feature with something like:
 echo '127.0.0.127 xml.cisco.com' >> /etc/hosts
```

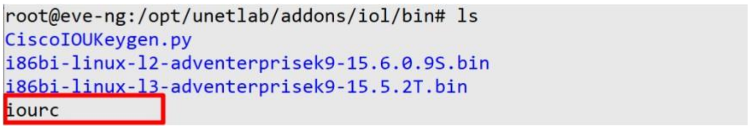
- Generate the iourc file to store the license key generated above.



```
root@eve-ng:/opt/unetlab/addons/iol/bin# sudo touch iourc
root@eve-ng:/opt/unetlab/addons/iol/bin# sudo nano iourc
```



```
  GNU nano 2.5.3              File: iourc

[license]
eve-ng = 972f30267ef51616;

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell
```



```
root@eve-ng:/opt/unetlab/addons/iol/bin# ls
CiscoIOUKeygen.py
i86bi-linux-l2-adventerprisek9-15.6.0.9S.bin
i86bi-linux-l3-adventerprisek9-15.5.2T.bin
iourc
```

- Establish the permissions for the IOL/U images using the provided command.

```
root@eve-ng:~#  /opt/unetlab/wrappers//unl_wrapper -a
fixpermissions
```

4.3 QoS Scenarios

This part describes and tests two scenarios of QoS implementation in a company's network, because of the absence of compatibility with other vendors in simulation environments, I selected Cisco and executed all scenarios using Cisco equipment. The primary focus lies on video and voice traffic, as Quality of Service (QoS) is initially designed and deployed specifically for media traffic. For the deployment of Quality of Service (QoS), we intend to adopt the hierarchical architecture proposed by Cisco.

This strategy offers numerous advantages, such as scalability for IP network expansion, sufficient network resources at the core and distribution layers, simplified network management and improved security measures. The architecture is divided into three main layers: Core, Distribution and Access (12).

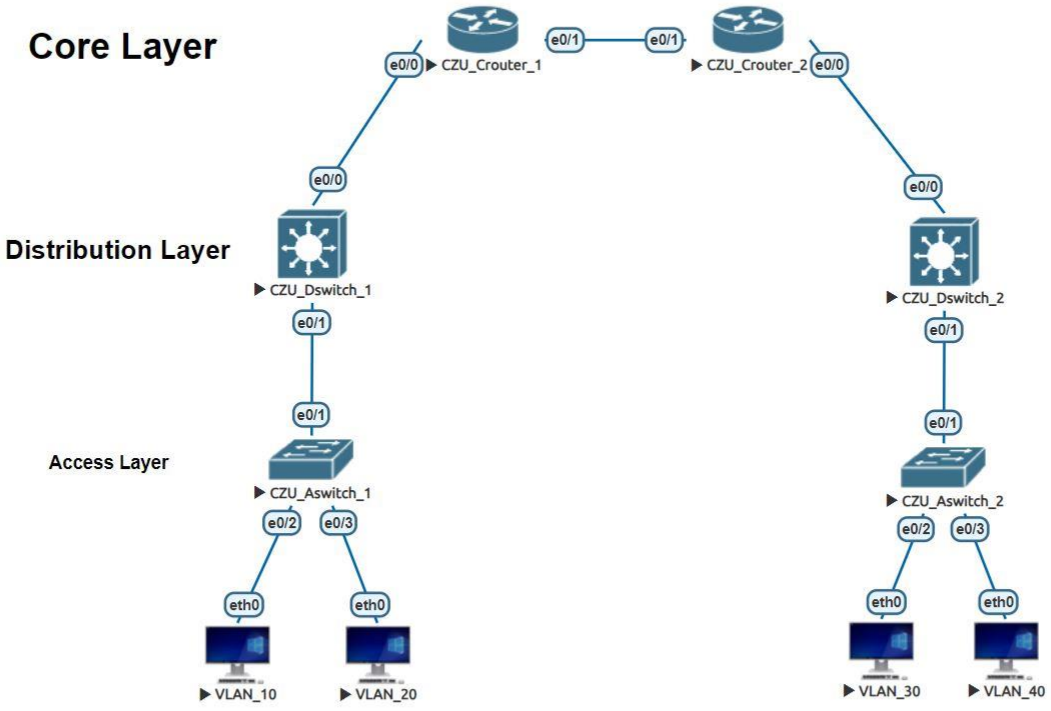


Figure 17: Cisco Hierarchical Architecture, source: author

Core Layer: At this tier of the Cisco architecture, high-speed routing occurs, facilitating rapid data transmission between two locations. Functioning as a backbone link, it boasts ample bandwidth to ensure swift and reliable data delivery. It's imperative to minimize configuration adjustments at this level to prevent any potential delays in traffic (10).

Distribution Layer: At the second level, incoming traffic from the access tier is aggregated and processed. Here, traffic is organized and directed towards the core network. The primary responsibility at this level involves traffic management tasks, such as implementing Access Control List (ACL) rules, applying Quality of Service (QoS) mechanisms, and routing between logically segmented VLANs. Additionally, the device must possess robust computational capabilities and ample redundancy in paths and bandwidth to accommodate the high volume of packet processing (10).

Access Layer: The final tier of the hierarchical model encompasses all end devices, including VoIP telephones, printers, laptops, workstations, mobile phones, and others, which are granted access to the IP network (10).

he primary objective requires enabling network access while safeguarding against unauthorized access. Achieving this goal may involve employing port security measures on switches. Furthermore, a common strategy involves partitioning networks logically using VLANs. This practice serves multiple purposes, including minimizing broadcast traffic on individual switches, enhancing overall security, simplifying group management within the IP network, and enabling multiple end devices to connect to a single subnet, even if they are not directly connected to the same switch physically.The assessment will occur on endpoint stations situated within VLAN20 and VLAN50. Each station is equipped with the software tools specified below:

- IP Traffic – Test & Measure
- VLC Media Player
- Linphone
- FTP server

The traffic generation will be executed through IP Traffic – Test & Measurement applications. Additionally, VLC Player and Linphone applications will facilitate user perception assessments. The bandwidth allocation for voice traffic is designated as 64 kbit/s, while video traffic is allotted 3.750 Mbit/s.

4.3.1 Scenario #1:

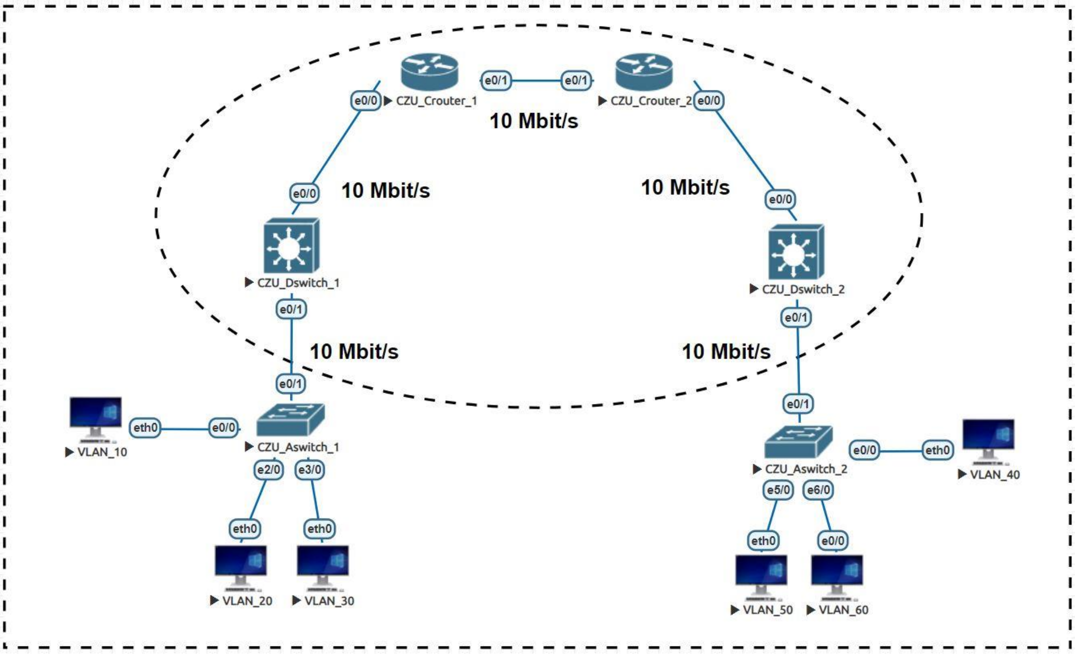


Figure 18: Network Topology in Scenario #1, source: author

Network Setup (Scenario#1) parameters:

Treatment method	Traffic	DSCP	Resource Allocation
DSCP	VOICE VIDEO	EF CS4	LLQ, 200 kbit/s CBWFQ, 6000 kbit/s

Table 15: Router Handling Process in Scenario #1, source: author

Treatment method	Class	Traffic	DSCP/CoS	Interface
Trust boundaries in incoming tags	-	All	N/A	e0/1
DSCP	- -	VOICE VIDEO	EF CS4	e0/2-4
	1	Other	0,1,2	

WRR	2 3	Other VIDEO	3 4	e0/2-4
Accelerated Forwarding	4	VOICE	5,6,7	e0/2-4

Table 16: Switch Handling process in Scenario #1, source: author

The initial measurement indicates that implementing quality of service (QoS) mechanisms exclusively on routers does not yield substantial improvements across all performance metrics for voice and video traffic. Initially, the network operated within acceptable parameters. However, upon introducing a load, a noticeable degradation was observed in video traffic, which utilized a bandwidth of 3.75 Mbit/s. In contrast, voice traffic demonstrated superior performance, consuming only 64 kbit/s of bandwidth. Despite experiencing higher jitter, voice traffic-maintained throughput close to the designated bandwidth for both types of traffic, with delays ranging from 10 to 40 ms.

Note: All results are stored separately in folder “**Farhad_Abbasi_MSc_Thesis_Result**”.

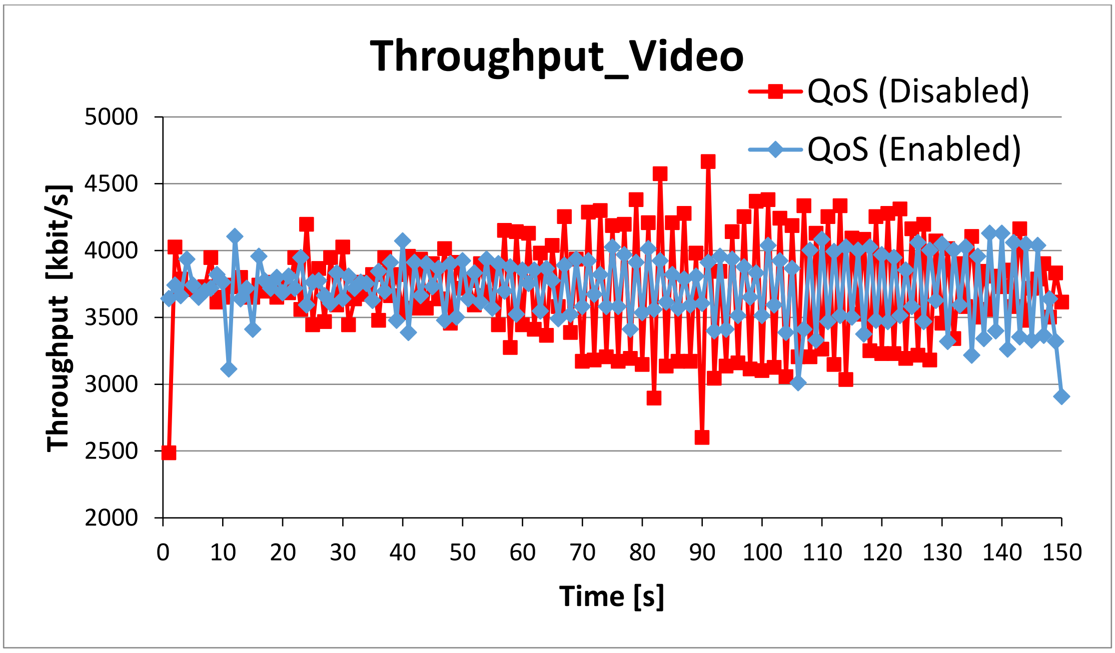


Figure 19: The measurement result regarding Throughput in Video traffic in NotExpanded scenario #1, source: author

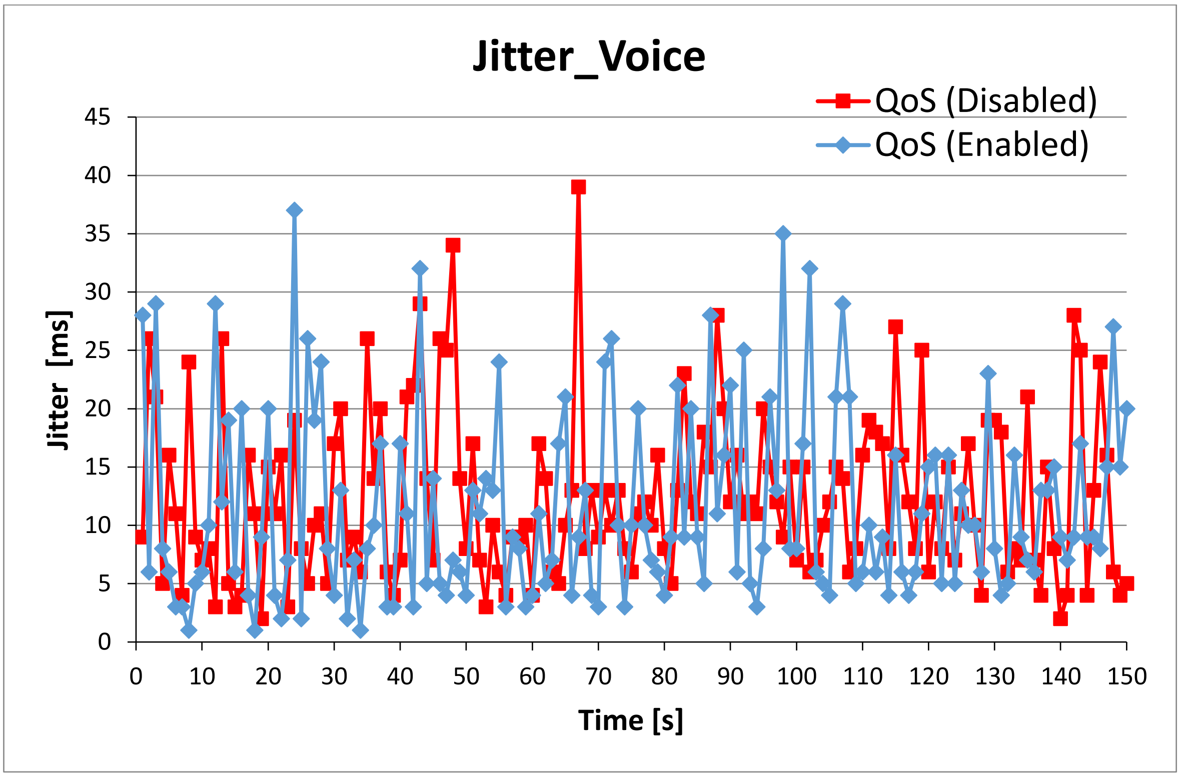


Figure 20: The measurement result regarding Jitter in Voice traffic in NotExpanded scenario #1, source: author

In the subsequent evaluation phase, expanding QoS implementation to switches resulted in a discernible improvement. QoS mechanisms implementation led to significant enhancements across all measured traffic parameters within the loaded network. Bandwidth remained stable at the designated value, with no observed loss, minimal Jitter, and consistent delay levels. Notably, voice traffic achieved a delay of less than 1 ms, represented as a value of 0 ms in graphical representations.

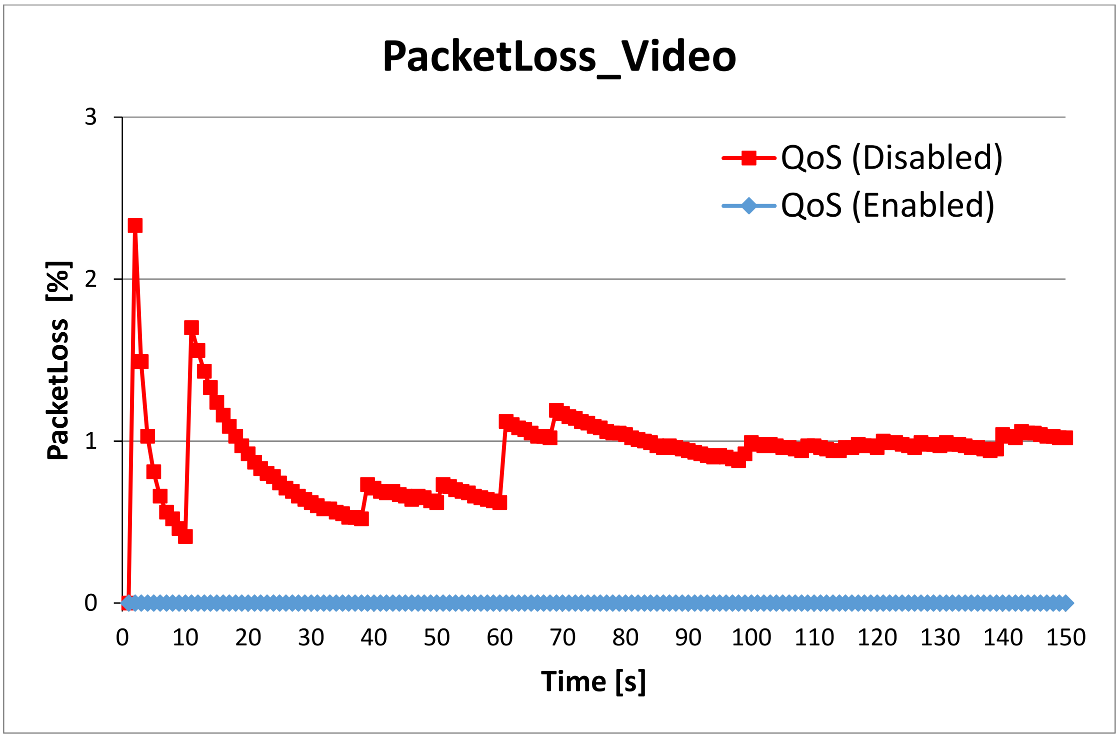


Figure 21: The measurement result regarding PacketLoss in Video traffic in Expanded scenario #1, source: author

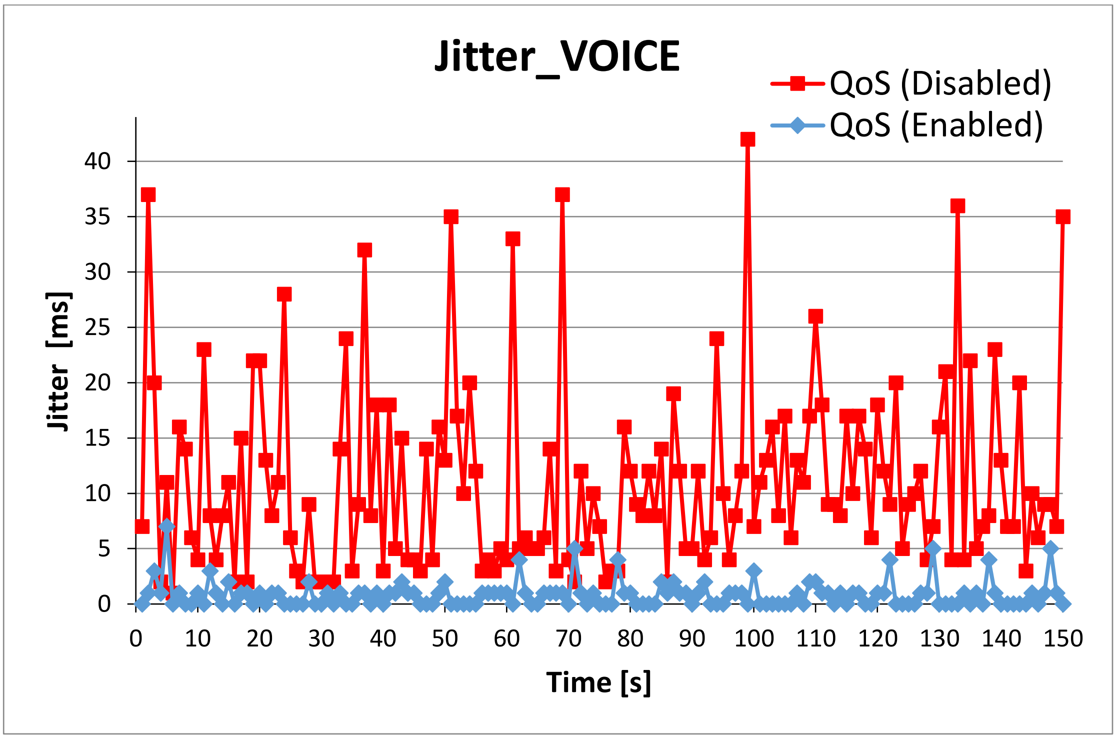


Figure 22: The measurement result regarding Jitter in Voice traffic in Expanded scenario #1, source: author

Note: All results are stored separately in folder "**Farhad_Abbasi_MSc_Thesis_Result**".

User perspective or User viewpoint

Voice and video traffic without QoS

Voice calls conducted through Linphone maintained consistent quality despite network congestion. This consistency stems from the program's minimal bandwidth requirements, with the demanding G.711-µlaw codec utilizing only 64 kbit/s. Minor fluctuations in loss and delay during operation have negligible effects on call quality.

Streaming video with VLC Player over an uncongested network proceeded smoothly. However, accessing via FTP transfer revealed noticeable issues, including the initial appearance of macroblocks followed by subsequent picture and sound dropout. Despite requiring an average bandwidth of approximately 3500 kbit/s for optimal performance, this bandwidth was unavailable. Fortunately, the program's built-in buffering mechanism prevented disruptions caused by delay fluctuations.

Voice and video traffic with QoS

Once again, voice calls experienced no noticeable quality issues in either congested or uncongested networks. Packet marking verification within the DS (Differentiated Services) domain was performed using the Wireshark program. It is advisable to implement Quality of Service (QoS) in situations involving a substantial number of concurrent calls. Streaming video to an uncongested network proceeded smoothly. However, when loaded via FTP transfer, a significant improvement was observed compared to a loaded network without QoS implementation. Throughout the transmission, the video remained consistently visible without the occurrence of macroblocks, and the audio remained uninterrupted.

4.3.2 Scenario #2

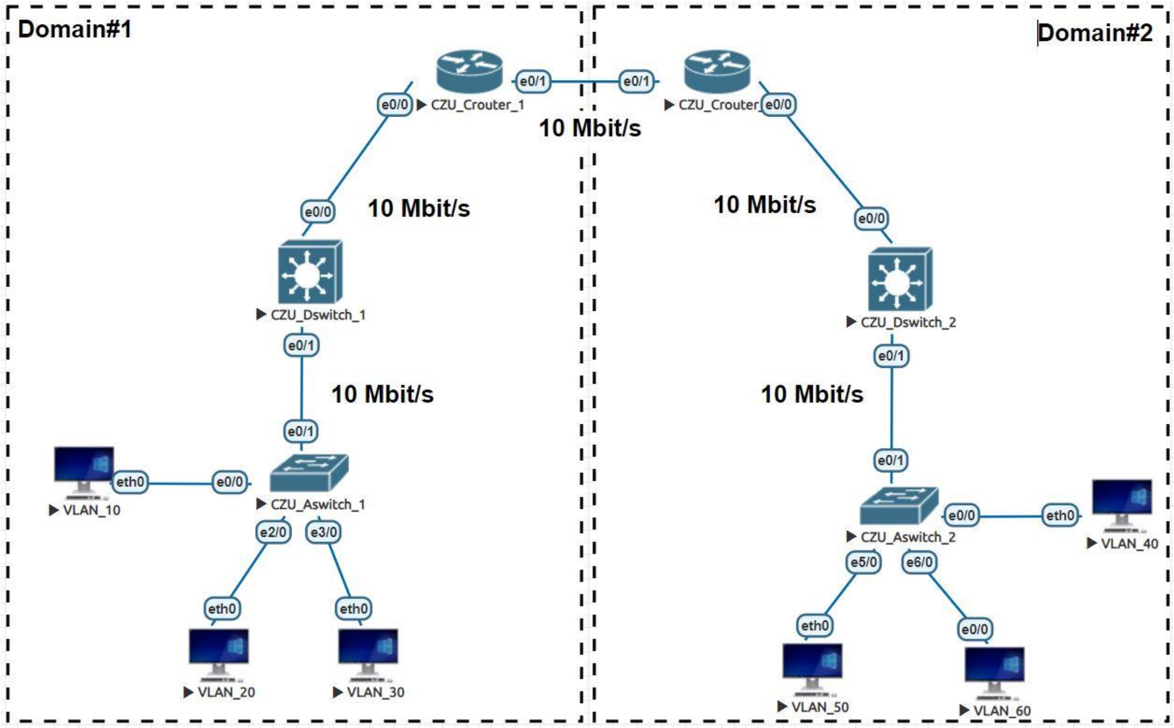


Figure 23: Network Topology in Scenario #2, source: author

Network Setup (Scenario#2) parameters:

Treatment method	router	Traffic	DSCP	Interface
DSCP	CZU_Crouter_1	VOICE VIDEO	CS4 CS3	e0/1 e0/1
DSCP	CZU_Crouter_2	VOICE VIDEO	EF CS4	e0/1 e0/1

Table 17: Mapping table for DS (Differentiated Service) in Scenario #2, source: author

Treatment method	router	Traffic	Resource Allocation	Interface
DSCP	CZU_Crouter_1 CZU_Crouter_2 CZU_Dswitch_1 CZU_Dswitch_2	VOICE VIDEO	LLQ, 200 kbit/s CBWFQ, 6000 kbit/s	e0/0-1

Table 18: Handling process of the first domain of DS (Differentiated service) in Scenario #2, source: author

Treatment method	router	Traffic	Resource Allocation	Interface
DSCP	CZU_Crouter_1 CZU_Crouter_2 CZU_Dswitch_1 CZU_Dswitch_2	VOICE VIDEO	CBWFQ, 100 kbit/s CBWFQ, 4000 kbit/s	e0/0-1

Table 19: Handling process of the second domain of DS (Differentiated service) in Scenario #2, 3rd part, source: author

Treatment method	Class	Traffic	DSCP/CoS	Interface
Trust boundaries in incoming tags	-	All	N/A	e0/1
DSCP	-	VOICE	EF	e0/2-4
	-	VIDEO	CS4	
WRR	1	Other	0,1,2	e0/2-4
	2	Other	3	
	3	VIDEO	4	
Accelerated Forwarding	4	VOICE	5,6,7	e0/2-4

Table 20: Switch (CZU_Aswitch_1) Handling process in Scenario #2, source: author

In the second scenario, once more employing the hierarchical Cisco architecture, two distinct Differentiated Service (DS) domains will be established utilizing all accessible routers and switches. Network routing will be facilitated through the OSPF protocol. Within both domains, diverse DSCP values will be designated to enable QoS implementation via differentiated services. The network parameter configurations remain same to those outlined in the initial proposed network, with distinctions found solely in the traffic handling rules within DS domain 2. Bandwidth will be constrained to 10 Mbit/s across all network elements within both DS domains. This restriction effectively replicates line load conditions. With the network configured accordingly, the measurement will unfold in three distinct phases:

- The analysis will assess the impact of remarking between the two domains on traffic jitter, delays, bandwidth utilization, and packet loss rates for both data and voice traffic within a loaded network implementing QoS. The setup closely resembles the first scenario, with devices (CZU_Crouter_1-2, CZU_Dswitch_1-2) configured as specified in Table 23, however utilizing traffic mapping on the domain interface as per Table 22. In DS domain 2, incoming packets are remarked, with DSCP mark EF reassigned to CS4 for voice traffic and CS4 packets rerouted to CS3 for video traffic. Conversely, DS domain 1 re-marks incoming traffic in the opposite manner: packets with DSCP mark CS4 are reassigned to EF, while those with CS3 mark are redirected to CS4.

- Following the evaluation of remarking effects, the implementation of Weighted round-robin scheduling service for individual classes will be terminated at switch CZU_Aswitch_2. Subsequently, traffic parameters will be reassessed in a loaded network, both with and without QoS measures in place.

- A distinction will be observed between utilizing a bandwidth-bound queue and a low delay queue. To achieve this, it will be essential to reconfigure routers CZU_Crouter_2 and CZU_Dswitch_2 in accordance with Table 24. Subsequently, the influence on individual traffic parameters in a QoS-enabled loaded network will be evaluated.

Measurement results in scenario #2:

- Re-Marking at the interface of both domains exhibited no discernible impact on the resultant traffic. The resolution capability of the software equipment stands at a delay of 1 ms. Comparison of delay values from the preceding scenario revealed no significant alterations. Based on this observation, it can be inferred that re-marking occurs within a 1 ms timeframe.

- Cancellation of the Weighted Round-Robin scheduling service for various classes on switch CZU_Dswitch_2 significantly influenced all traffic parameters within the IP network. Notably impacted were the delay and loss rate. The jitter between 40-100ms for both types of traffic, with QoS utilization resulting in a range of 10-40ms.

Interestingly, the presence or absence of QoS did not affect the loss rate, which remained consistent at 0-5%.

- Transitioning from a low-delay queue to a guaranteed-bandwidth queue resulted in minimal impact on the voice load processed by the routers. Latency experienced a mere 3ms increase in a congested network. Given the negligible difference, no practical implications arise in the current network setup.

Note: All results are stored separately in folder "**Farhad_Abbasi_MSc_Thesis_Result**".

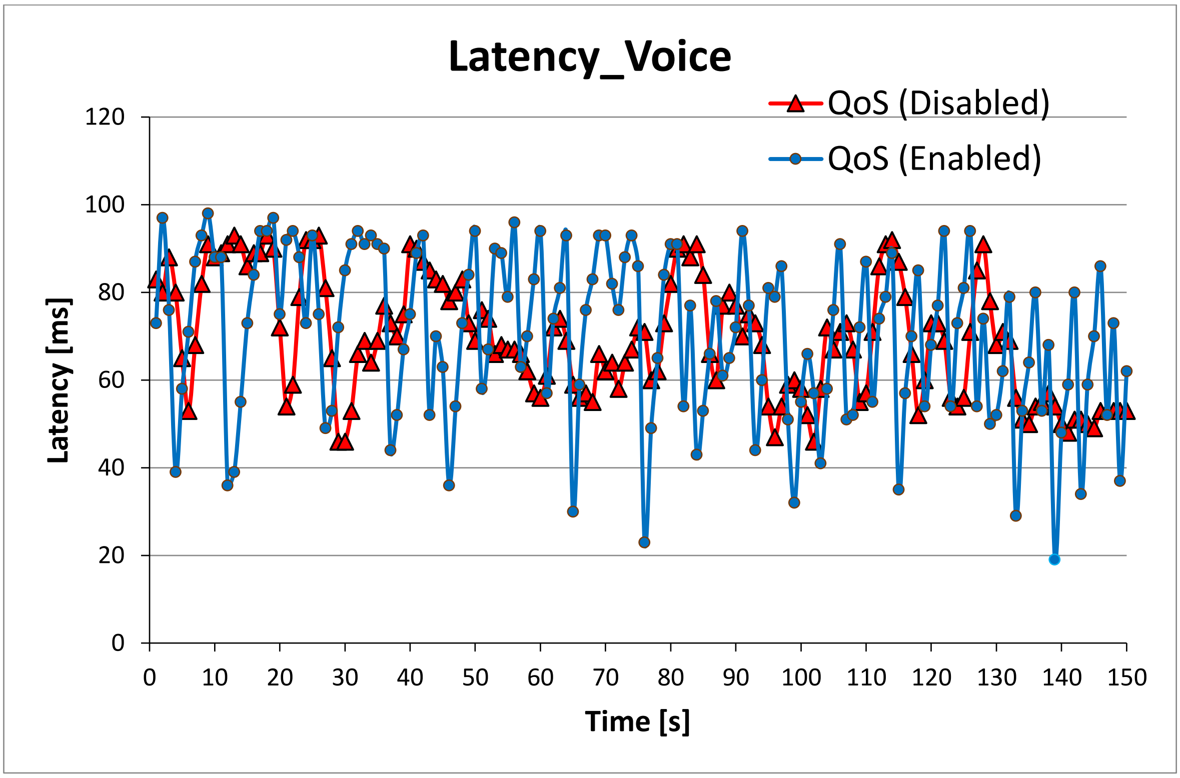


Figure 24: The measurement result regarding delay in Voice traffic in scenario #2, source: author

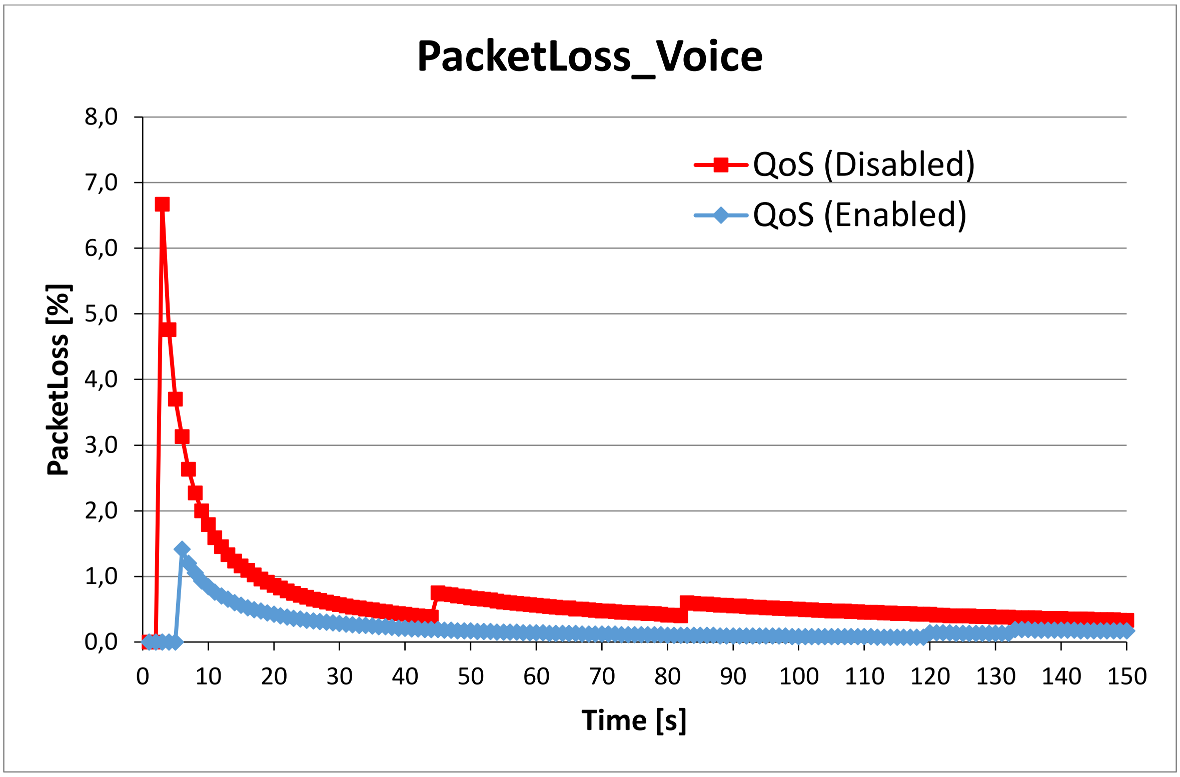


Figure 25: The measurement result regarding Packet Loss in Voice traffic in scenario #2, source: author

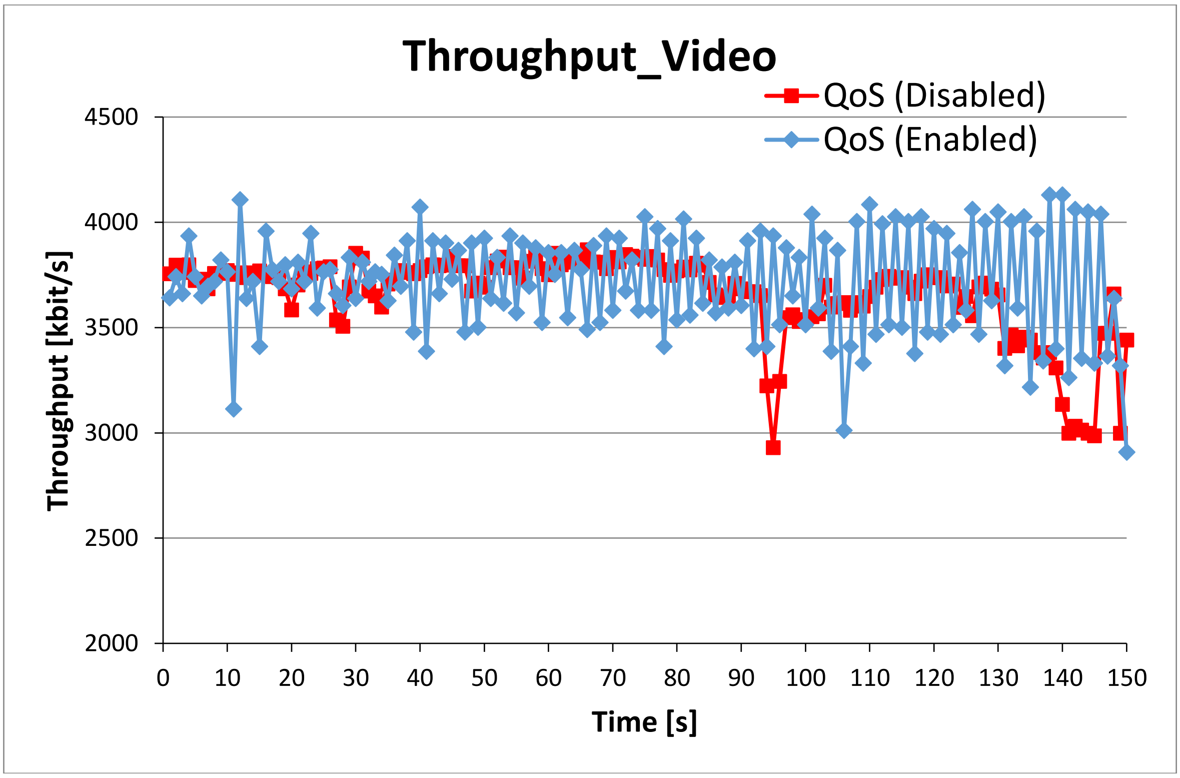


Figure 26: The measurement result regarding Throughput in Video traffic in scenario #2, source: author

5. Results and Discussion

The first scenario's measurement findings show that there is very little benefit over not employing QoS. But QoS has a lot of benefits: more constant delays, less latency, no packet loss, and faster data flow—which might potentially be as low as 0 ms for voice interactions. User testing showed that when the network was busy, using QoS improved the quality of video streaming. Poor quality of service (QoS) videos frequently included audio pauses and jagged images. A few calls might get through the network without any issues if QoS is used, but more calls would cause dropped calls and other issues.

	Latency [ms]	**Throughput [kbit/s]**	**Jitter [ms]**	**Packet Loss [%]**
Video QoS (Disabled)	15.48	3744.2	0.813	0.0926
Voice QoS (Disabled)	8.91	68.36	2.8	0.0266

Table 21: The average test results for unloaded network QoS parameters in Scenario #1_Not Expanded, source: author

	Latency [ms]	**Throughput [kbit/s]**	**Jitter [ms]**	**Packet Loss [%]**
Video QoS (Disabled)	72.38	3713.34	1.46	0.61
Voice QoS (Disabled)	67.98	68.29	12.44	0.00
Video QoS (Enable)	75.28	3713.29	1.35	0.50
Voice QoS (Enabled)	70.37	68.14	11.38	0.68

Table 22: The average test results for loaded network QoS parameters in Scenario #1_Not Expanded, source: author

	Latency [ms]	**Throughput [kbit/s]**	**Jitter [ms]**	**Packet Loss [%]**
Video QoS (Disabled)	15.41	3729.31	0.84	0.01
Voice QoS (Disabled)	9.3	68.44	3.41	0.00

Table 23: The average test results for unloaded network QoS parameters in Scenario #1_Expanded, source: author

	Latency [ms]	**Throughput [kbit/s]**	**Jitter [ms]**	**Packet Loss [%]**
Video QoS (Disabled)	73.89	3706.35	1.3	0.91
Voice QoS (Disabled)	68.56	68.36	11.2	0.13
Video QoS (Enable)	34.82	3740.87	1.473	0.00
Voice QoS (Enabled)	10.42	68.44	0.813	0.00

Table 24: The average test results for loaded network QoS parameters in Scenario #1_Expanded, source: author

The outcomes of the measurement in scenario #2 It shows that remarking at the interface between two Differentiated Services (DS) domains has little impact on traffic parameters. Additionally, voice traffic resources can be reallocated from a guaranteed-bandwidth queue to a low-delay queue with no impact in small networks. However, in larger networks, this reallocation may cause undesirable delays. In addition, when the quality of services on the switch is terminated, all evaluated metrics of the individual operations significantly degrade. These results demonstrate how important it is to consider network size and how important quality of service strategies are for maintaining optimal performance.

	Latency [ms]	**Throughput [kbit/s]**	**Jitter [ms]**	**Packet Loss [%]**
Video **QoS (Disabled)**	20.2	3740.92	0.62	0.30
Voice **QoS (Disabled)**	15.22	68.42	3.38	0.00

Table 25: The average test results for unloaded network QoS parameters in Scenario #2, source: author

	Latency [ms]	**Throughput [kbit/s]**	**Jitter [ms]**	**Packet Loss [%]**
Video **QoS (Disabled)**	78.24	3657.82	2.04	1.06
Voice **QoS (Disabled)**	69.51	68.41	13.94	0.68
Video **QoS (Enable)**	75.28	3713.29	1.35	0.50
Voice **QoS (Enabled)**	70.37	68.14	11.38	0.20

Table 26: The average test results for loaded network QoS parameters in Scenario #2, source: author

6. Conclusion

The main goal of this master thesis is to propose and validate a solution for theimplementation of Quality of Service (QoS) in the network of an IT outsourcing company.

In the literature review part, explored Quality of Service (QoS) theory and its application across IP network. It covered in details about traffic Characteristics, Planning & Implementing QoS policies, QoS Models, Packet Classification and Marking, Congestion Management, Congestion Avoidance, Shaping & Policing, QoS in Local Area Networks (LAN).

In the practical part, two scenarios of Quality of Service (QoS) implementation were tested within a company's network, using only Cisco equipment exclusively due to limitations on interoperability with other vendors in the simulation environments. The network design followed a hierarchical architecture proposed by Cisco, incorporating Access, Distribution, and Core layers. Additionally, to meet contemporary network requirements, other technologies such as port security, VLANs, and IP routing were also configured. To generate and measure traffic in the QoS scenarios, various tools including IP Traffic – Test & Measure, VLC Media Player, Linphone, and FTP server were configured.

Following the analysis of scenarios #1 and #2 When it comes to managing traffic parameters and ensuring a better user experience, Quality of Service (QoS) is crucial for enhancing network performance. Because of this, it is recommended to use QoS algorithms across the network to lower latency, avoid packet loss, and maintain constant data flow. Furthermore, selecting the optimal QoS configuration requires careful consideration of the particular requirements and network architecture. To avoid unwanted delays in bigger networks, voice traffic resource allocation needs to be carefully handled. Finally, QoS cancellation on switches need to be avoided as it significantly reduces the effectiveness of individual operations. Peak network performance and user satisfaction can generally be maintained by giving QoS implementation and customization priority based on network size and requirements.

7. References

1. **Brad Edgeworth, Ramiro Garza Rios, Jason Gooley, David Hucaby.** *CCNP and CCIE Enterprise Core Encore 350-401.* San Jose, CA : Cisco Press, 2020.

2. **Steve Jordan, Anthony Bruno.** *CCNP Enterprise Design 300-420 Official Cert Guide: Designing Cisco eEnterprise Networks.* [ed.] 1st edition. Hoboken, New Jersey : s.n., September 9, 2020.

3. **Aruba Education Development Team.** *Aruba Certified Design Professional: Official Certification Study Guide (HPE6-A47).* San Francisco, CA : HPE Press, August 2, 2018.

4. **Allred, Miriam.** *Aruba Certified Switching Professional: Official Certification Study Guide (HPE6-A45).* San Francisco : Hewlett Packard Enterprise Press, 2018.

5. **Lammle, Todd.** *CCNA Certification Study Guide, Volume 1: Exam 200-301.* USA : John Wiley, 2020. ISBN 978-1-119-65902-0.

6. **Lammle, Todd**. *CCNA Certification Study Guide, Volume 2: Exam 200-301.* USA : John Wiley, 2020. ISBN 978-1-119-65918-1.

7. **Szigeti, Tim.** *End-to-End QoS Network*, Second Edition, Cisco Press, 2013. ISBN-13: 978-1-58714-369-4.

8. **Alvarez, Santiago.** *QoS for IP/MPLS Networks.* s.l. : Cisco Press , 2006. SBN-10: 0-13-343499-0, ISBN-13: 978-0-13-343499-6.

9. **Szigeti, Tim.** *End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs.* s.l. : Cisco Press, 2004. ISBN-10: 1-58705-176-1, ISBN-13: 978-1-58705-176-0.

10. ***Connecting networks v6****: companion guide.* Indianapolis, IN : Cisco press, 2018. ISBN 978-1-58713-432-6.

11. **D, TEARE**. *Implementing Cisco IP routing (ROUTE) foundation learning.* s.l. : Cisco Press, 2015. ISBN 978-1-58720-456-2.

12. **DOYLE, J**. *Routing TCP / IP, Volume II: CCIE Professional Development. Second Edition.* Indianapolis : Cisco Press, 2017. ISBN 1-58705-202-4.

13. **QoS: Congestion Management Configuration Guide**, Cisco IOS XE Release 3S. [Online] https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_conmgt/configuration/xe-3s/qos-conmgt-xe-3s-book/qos-conmgt-oview.html.

14.**Lutkevich,B**.techtarget.[Online]https://www.techtarget.com/searchunifiedcommunications/definition/QoS-Quality-of-Service.

15.**QoS and Congestion Avoidance**. [Online] 2010. https://www.routeralley.com/guides/qos_congestion_avoidance.pdf.

16. **QoS: Congestion Avoidance Configuration Guide**. [Online] https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_conavd/configuration/xe-16/qos-conavd-xe-16-book.pdf.

17. **what-when-how, Congestion Avoidance**. [Online] https://what-when-how.com/ccnp-ont-exam-certification-guide/congestion-avoidance/.

18. **Compare Traffic Policy and Traffic Shape to Limit Bandwidth**. [Online] https://www.cisco.com/c/en/us/support/docs/quality-of-service-qos/qos-policing/19645-policevsshape.html.

19. **QoS Policing and Shaping.** [Online] https://ipcisco.com/lesson/policing-and-shaping/.

20. **Traffic Policing vs. Traffic Shaping**. [Online] https://www.routerfreak.com/traffic-policing-vs-traffic-shaping/.

21. **IPv6 QoS: Queueing**. [Online] https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_conmgt/configuration/xe-16-10/qos-conmgt-xe-16-10-book/ip6-qos-queue-xe.pdf.

22. **COMMSCOPE: IPv6 QoS**. [Online] https://docs.commscope.com/bundle/fastiron-08090-trafficguide/page/GUID-02F719EE-D9E4-4A1C-814A-12A18A89B3DC.html.

23. **RFC 2460**: Internet Protocol, Version 6 (IPv6) Specification. [Online] 1998. https://www.ietf.org/rfc/rfc2460.txt.

24. **nojitter**: QoS in the LAN? [Online] https://www.nojitter.com/qos-lan-youre-kidding.

25. **RFC 2328**: OSPF Version 2. [Online] http://www.ietf.org/rfc/rfc2328.txt.

26. **Jadi**. Emulation-vs-Simulation. [Online] 2016. https://jadi.net/2016/06/emulation-vs-simulation/.

27. **Wikipedia. Emulator**. Wikipedia.org. [Online] https://en.wikipedia.org/wiki/Emulator.

28. **GNS3: Documentation**. [Online] https://docs.gns3.com/docs/.

29. **Dynamips. Wikipedia.org**. [Online] https://en.wikipedia.org/wiki/Dynamips.

30. **EVE-NG: Documentation**. [Online] https://www.eve-ng.net/index.php/documentation/.

31. **EVE-NG: community-cookbook**. [Online] https://www.eve-ng.net/index.php/documentation/community-cookbook/.

32. **Main Page," wiki.qemu.org**. [Online] https://wiki.qemu.org/Main_Page.

List of Figures

List of Tables

List of abbreviations

ACL	ACCESS CONTROL LIST
BE	Best-Effort
BER	Bit Error Rate
CBWFQ	Class-Based Weighed Fair Queuing
CIR	Committed Information Rate
CoS	Class of Service
DiffServ	Differentiated Services
DSCP	Differentiated Services codepoint
EVE-NG	Emulated Virtual Environment-Next Generation
ECN	Explicit Congestion Notification
FIFO	First In First Out
FQ	Fair Queuing
FTP	File Transfer Protocol
IP	Internet Protocol
IPP	IP precedence
IntServ	Integrated Services
IOL	IOS on Linux
LAN	Local Area Network
LFI	Link Fragmentation & Interleaving
LLQ	Low Latency Queuing
MQC	Modular QoS Command-Line Interface
NS	Network Simulator
PHB	Per–Hop Behavior
PIR	Peak Information Rate
PMP	Pulse Code Modulation
QoE	Quality of Experience
QoS	Quality of Service
RED	Random Early Detection
RSVP	Resource reservation Protocol
SLA	Service Level Agreement
TCP	Transmission Control Protocol
TOS	Type of Service
UDP	User Datagram Protocol
VoD	Video on Demand
VoIP	Voice over Internet Protocol
WAN	Wide Area Network
WFQ	Weighted Fair Queuing
WRED	Weighted Random Early Detection
WRR	Weighted Round Robin

Appendix

Appendix_A

Tables of IP addresses scheme and configured ACLs

Appendix_B

Results of Measurement:

1) Scenario#1_NotExpanded
2) Scenario#1_Expanded
3) Scenario#2

Appendix_A:

Devices	Interfaces	Ip addreses	Gateway
CZU_Crouter_1	e0/0	172.16.0.5/30	N/A
	e0/1	10.0.0.5/30	N/A
CZU_Crouter_2	e0/0	172.16.3.1/30	N/A
	e0/1	10.0.0.6/30	N/A
CZU_Dswitch_1	e0/0	172.16.0.6/30	N/A
	e0/1	172.16.1.1/24	N/A
CZU_Dswitch_2	e0/0	172.16.3.2/30	N/A
	e0/1	172.16.2.1/24	N/A
CZU_Aswitch_1	e0/1	172.16.1.2/24	N/A
	VLAN10	172.16.10.1/24	N/A
	VLAN20	172.16.20.1/24	N/A
	VLAN30	172.16.30.1/24	N/A
CZU_Aswitch_2	e0/1	17.16.2.2/24	N/A
	VLAN10	172.16.40.1/24	N/A
	VLAN20	172.16.50.1/24	N/A
	VLAN30	172.16.60.1/24	N/A
PC (VLAN_10)	NIC	172.16.10.3/24	172.16.10.1/24
PC (VLAN_20)	NIC	172.16.20.3/24	172.16. 10.1/24
PC (VLAN_30)	NIC	172.16.30.3/24	172.16. 10.1/24
PC (VLAN_40)	NIC	172.16.40.3/24	172.16.40.1/24
PC (VLAN_50)	NIC	172.16.50.3/24	172.16. 40.1/24
PC (VLAN_60)	NIC	172.16.60.3/24	172.16. 40.1/24

Table 27: Network Configuration (IP addresses) , source: author

ACL (ACCESS CONTROL LIST)	TRAFFIC	PORTS
VIDEO	VIDEO	1050-1900
VOICE	VOICE	5060,49100- 49500

Table 28: ACCESS CONTROL LISTS, source: author

Appendix_B

All results are stored separately in folder “**Farhad_Abbasi_MSc_Thesis_Result**”.