Gediminas Stankevičius

Kaunas College "Electrolan" computer network

Bachelor’s thesis

Bachelor of Engineering

Program in Information Technology

2021

South-Eastern Finland
University of Applied Sciences

XAMK
South-Eastern Finland
University of Applied Sciences

Author (authors)	**Degree title**	**Time**
Gediminas Stankevičius	Bachelor of Engineering	March 2021

Thesis title	
Kaunas College "Electrolan" computer network	59 pages 10 pages of appendices

Commissioned by

Supervisor

Matti Juutilainen

Abstract

The purpose of this project is to create and implement a network in Kaunas college that could be used for e-sports and similar events, with around 100 attendees.

At the moment Kaunas college has no network ready to host e-sports matches and tournaments. That being said, various e-sports events are becoming increasingly popular. Newer generations are even more likely to take an interest in e-sports rather than traditional sports. Organizing these kinds of events would increase the likelihood of potential students becoming interested in the university or in certain degrees being offered. In addition to that, it would also help spread the name of the university while also improving people's opinion of it.

Kaunas college has enough network equipment needed for this kind of network. The equipment is usually only used for bachelor’s degree projects or to help students get to know how the equipment works. Using the network devices for e-sports events would be rather simple since it is only used rarely. That being said, guaranteeing a stable network at the events for that many people requires certain preparations.

The layout and configuration of the network devices was done in a way to guarantee as much network bandwidth as possible. The devices have been configured in a way that creates a network less prone to various network disturbances. A QoS (Quality of Service) configuration was created to prioritize certain traffic on the network. Various network vulnerabilities were also addressed. The chances of the vulnerabilities being exploited have been minimized by certain device configurations. The network was also equipped with the ability to remotely manage all of the network devices, which can be very useful during events. A cache server was also created. This server helps minimize the possibility of a reduced internet connection with remote networks. The server automatically downloads the games that event users are trying to download at the events. Once the server has downloaded the game, it serves them to the event attendees locally when requested.

Keywords

e-sports, cache server, local area network

CONTENTS

APPENDICES

1 INTRODUCTION

The objective of this thesis is to plan out and configure a computer network at the Kaunas college to offer students and other people the ability to organize various e-sports and similar events with the hardware already available in the college.

To hold e-sports events, a network must meet certain requirements. One of these requirements is that the network must be fast and have the ability to prioritize certain data over other less important data. The optimal network bandwidth between a local or remote server and a participant in the e-sports event should be around 4Mbps. If it is lower, the participants might have slowdowns in games during the events. However, speed is not the only thing the network needs. The network must also have a latency to the game servers of less than 150ms. Higher latency might create stutters and inconsistencies in events requiring fast reaction times. Minimizing the network traffic to the outside network is also an important part of the network. If too many people have to access and download certain resources outside of the local network, a bottleneck situation can occur where the download speeds slow down due to the large amount of people downloading something at the same time and the network edge devices not being able to reliably provide them with the expected speeds. To avoid this, it is best to use a local cache server that stores large game or other software files. Lastly, the network must be able to support a relatively large number of connected devices. Since the network in Kaunas college is not designed for these kinds of activities it does not fulfill these requirements. For this reason, it was decided that the best approach would be to create a separate network segment with a connection to the network edge as to not disturb the rest of the network.

Kaunas college has enough network devices, usually used for teaching purposes only, that can be used to create a temporary network for hosting the desired e-sports events. Since these devices are used rarely and e-sports events usually last no more than 1–2 days, they are the perfect choice for this project. One of the goals for this network is for it to support at least 100 devices, while giving them a stable and reliable enough connection. Because of this, multiple switches will be needed in the network. Thankfully, the available network hardware consists of three Cisco 2950 and another three Cisco 2960 switches. Also, two Cisco 2911 routers with an SM-ES2-24-P EtherSwitch module. On top of all that, there are a few old HP P4500 G2 servers with fast enough network interface cards that can be used for local hosting of the cache server.

2 BACKGROUND STUDY

Before configurations can be started it is important to have a clear idea of the strengths and weaknesses of the available hardware. To maximize the strengths and minimize the weaknesses, various technologies can be used. Doing a background study on all these technologies can help better understand their effects on the network.

2.1 Available network hardware

In total, the network will have three different switch models. All of them have at least 24 interfaces, however, each of them has their own differences which will determine how and where they will be used in the network.

Figure 1 Cisco 2950 switch

Table 1 Specifications of the Cisco 2950 switch

10/100 interfaces	24
10/100/1000 interfaces	0
Flash memory	8MB
RAM memory	16MB
Energy requirements	30W
Model	2950-24
Manufacturer	Cisco

Figure 2 Cisco 2960 switch

Table 2 Specifications of the Cisco 2960 switch

10/100 interfaces	24
10/100/1000 interfaces	2
Flash memory	32MB
RAM memory	64MB
Energy requirements	22W
Model	2960-24TT-L
Manufacturer	Cisco

Figure 3 Cisco SM-ES2-24-P EtherSwitch module

Table 3 Specifications of the Cisco SM-ES2-24-P EtherSwitch module

10/100 interfaces	23
10/100/1000 interfaces	1
Flash memory	-
RAM memory	-
Energy requirements	22W
Model	SM-ES2-24-P
Manufacturer	Cisco

In total, the switches have 190 interfaces capable of 100Mbps speeds and six 1Gbps interfaces. The 100Mbps interfaces are planned to be used for connecting end devices to the network while the 1Gbps interfaces will be used for connections between network devices. However, because none of these switches are capable of layer 3 routing, a router will also be used in the network with the “Router on a stick” configuration.

Figure 4 Cisco 2911 router

Table 4 Specifications of the Cisco 2911 router

10/100/1000 interfaces	3
Flash memory	256MB
RAM memory	512MB
Energy requirements	40W
Model	2911
Manufacturer	Cisco

Since this router has three 1Gbps interfaces, two of them will be used for connections to the switches, while the third one will be used to connect the network to a network edge device. Also, Router-Switch (no date) states that in order for the EtherSwitch modules to function, they need to be connected to an integrated service router. Because of this, both Cisco 2911 routers will be used in the network.

The network will also have a cache server for storing certain files locally. Since the server’s main purpose is storing files and delivering them to end users, a large enough storage capacity and a fast enough network card are needed. The HP P4500 G2 should be able to fulfill both of the requirements.

Figure 5 HP P4500 G2 server

Table 5 Specifications of the HP P4500 G2 server

Processor	E5520
Processor cores	4 physical, 8 logical
Processor frequency	2.26GHz (Turbo 2.53GHz)
RAM type	DDR3 ECC
RAM storage	6GB
Physical storage	12TB
Network interfaces	2 x 1Gbps

2.2 Type of network topologies

Computer networks can have very different topologies depending on the number of users it needs to support, workloads or other needs of the network. However, according to Alison (2008), computer networks can usually be categorized into six main topologies: bus, ring, star, mesh, tree, hybrid.

Bus topology is one of the oldest network topologies. It works by having all of the network devices connect to the same central data lane. Due to the way devices are connected in this topology they create a lot of packet collisions and unnecessary noise when communicating with one another.

Ring topologies have all the network devices connect to one another in a circular loop. Packets in this topology are sent from one device to another in circular fashion until they reach the destination device.

Star topology is one of the most popular network topologies currently used. In this topology end devices are connected to a single central device, usually a switch or a hub. This topology can also be expanded by connecting multiple central devices to create an extended star. Whenever an end device wants to send a packet to another device, the packet travels through the central devices until it reaches the destination. This topology is great for connecting a large number of devices together, especially when the extended star variant of the topology is used.

Mesh topologies connect every single device in the network to one another. This topology is great for ensuring reliability since one or more broken links in the network don’t cause parts of the network to separate.

Tree topology is a hierarchical topology. In this topology a root device is at the top of the tree and has two or more devices connected to it, those other devices then also have two or more devices connected to them and so on.

Hybrid topologies are made up of a mix of other different topologies. They can be useful in certain legacy or specialized environments since they can incorporate older topologies and connect them with newer more common types of topologies. They however are very hard to maintain.

Since one of the goals of this project is to have the network support at least 100 end devices the network topology must be easy and quick to scale up according to these needs. After analyzing the strengths and weaknesses of these different topologies it was decided that for this project the star topology would be the best option. Specifically, the extended star variant, since it can connect multiple central devices to expand the number of end devices in the network.

2.3 Possible bottlenecks in the network and ways to mitigate them

Making sure a network doesn’t have any major bottlenecks, which could cause unwanted slowdowns and reductions in stability and reliability, is an important part when planning out a network. According to Sterbenz, et al. (2001) "The maximum bandwidth along a path is limited by the minimum bandwidth link or node, which is the bottleneck". A major bottleneck in any network can reduce its maximum bandwidth or even cause latency issues. These types of problems are especially harmful in e-sports environments, where a bad or unreliable connection can change the outcome of a competitive match. However, bottlenecks in the network can occur for a large variety of reasons.

One common reason a computer network can experience slowdowns is that the network hardware is unable to reliably handle the needed bandwidth. When a network device has to send a large stream of data to another network device it is important that the link between these devices is capable of doing so without having to drop packets. This dropping of packets can happen if the interfaces connecting the network devices have a low enough bandwidth. This problem can commonly occur with switches that are connected via low bandwidth links. When this happens the users on their end devices can experience noticeable slow downs in the network. To avoid this, it is always best to connect each network device using the interfaces with the highest maximum bandwidth capacity. Since the network in this project has three Cisco 2950 switches, which according to Cisco (2006) only have 100mbps interfaces, it will be important to aggregate some of these interfaces into one logical interface to help maximize the available bandwidth.

However, in certain situations it is not possible to have a link with high enough bandwidth. In these situations, the network needs a way to prioritize more important data over the less important one. Without the ability to prioritize what type of data needs to be sent out first, a network device can slow down the traffic flow by constantly dropping packets. Since the network in this project needs to work as a separate segment, a Cisco 2911 router will be used for its network edge. The interface used to connect it to Kaunas college's network edge has a maximum bandwidth of 1Gbps. Thankfully, most routers have various functionalities to help customize the priority of certain packets over others.

Bradley (2021) notes that a slow or unresponsive remote server outside the local network can cause noticeable slowdowns for users trying to connect to it or download files from it. This can be mitigated by making sure that when users need to download certain large files, stored outside the local network, they are available locally. If the files are stored locally, they also don't need to go through network edge devices thereby reducing the workload they have to deal with. In this project this is planned to be achieved by using a local cache server which intercepts certain requests to the outside network and redirects them to devices in the local network.

2.3.1 Port channel technology

Port channel is a technology that allows the combination of multiple physical interfaces or devices into a singular logical interface or device. It is commonly used to increase the maximum bandwidth between network devices by combining multiple lower bandwidth links into one link with a higher bandwidth capacity. To use the port channel functionality, both network devices need to support it. However, there are multiple port channel types a device can use. According to Joel (2012) there are four main types.

EtherChannel/ Link Aggregation is the most common way the port channel technology is used. It works by combining multiple physical interfaces on a device into one logical interface. To do so it uses the LACP or PAgP protocol. The protocols communicate with other EtherChannel links to determine if both of them are capable of creating a connection. If the protocols determine that the interfaces on both devices are configured correctly it links them together.

Virtual Port Channel (vPC) is a way to combine two physical switches into one logical switch. When doing so both switches can still be individually configured, however, from the perspective of other network devices, they are seen as a singular switch.

Virtual Port Channel Plus (vPC+) works similarly to vPC. However, vPC+ is used in FabricPath domains. FabricPath is a Cisco technology which helps large data centers improve the traffic flow to the servers by combining "traditional, Spanning Tree-based Ethernet with a next-generation architecture that uses a link-state protocol to allow for multiple active paths" -Jim (2012).

Enhanced Virtual Port Channel is similar to vPC+, but it is mainly used with Cisco Nexus 5500 type switches to help give them addition functionality.

Since the network in this project needs a way to aggregate several low bandwidth interfaces in ones with higher bandwidth, EtherChannel/ Link Aggregation will be used.

2.3.2 QoS technology

QoS (quality of service) is a technology found in routers, switches and other devices, which helps to regulate network traffic in a computer network. It works by giving priority to certain traffic in the network over other less important or less critical traffic. Barreiros, et al. (2016) notes that "QoS usage has increased to the point where it is now considered a necessary part of network design and operation". QoS is often used with the VoIP (voice over IP) protocol to make sure that a congested network does not cause stutters and latency in applications using the protocol.

According to Cisco (2017), Cisco routers have multiple ways to apply QoS to a network. Each way has its own differences, strengths and weaknesses. They all can be categorized into three main groups by their functionality: “queuing”, “shaping”, “policing”. The “queuing” functionality in QoS technology works by internally queuing each packet the network device receives and then sorting them out by their priority or other factors and only then sending them.

CBWFQ (Class Based Weighted Fair Queuing) is one of the simplest ways queuing is achieved. It works by creating classes with certain requirements for the traffic, after the classes are created, they are applied to groups and finally the groups are given a minimum network bandwidth they are allowed to use up. This way of queuing makes sure that certain programs or activity in the network always get a minimum amount of bandwidth when network congestion is detected.

LLQ (Low Latency Queuing) works similarly to CBWFQ, but it also allows certain packets to bypass the queuing and be sent over immediately. This type of queuing is especially useful in applications that need an uninterrupted and stable traffic flow to work without problems. LLQ is most often used with the VoIP protocol where even the slightest network stutters or congestions can be felt by the users. However, LLQ has to be used with caution because an incorrectly configured queuing can impact the performance of other applications in the network.

QoS shaping gives a network administer the ability to limit bandwidth over an interface. According to Adeolu (no date), QoS shaping is often used to match one interfaces maximum bandwidth with another interface. It is also commonly used by ISPs (Internet Service Providers) because it lets them set the maximum bandwidth a certain client can use by limiting the bandwidth on certain interfaces.

QoS policing works similarly to shaping. However, when using policing to limit the bandwidth of an interface any packets being sent when the limit is reached are immediately dropped instead of being queued up.

QoS functionality is very important in this type of project, since e-sports events are extremely sensitive to small network delays. Even an extra added latency of a few milliseconds can be easily noticed by participants. John (2021) recommends that between the participant’s device and the server on a remote or local network, the network upload speed must be at least 1Mbps and the download speed about 3Mbps. The latency between those two points should also be no more than 150ms. Due to all of these reasons using QoS services is extremely important in this kind of network. However strictly only allowing the game traffic in the network could lead to various other issues. Therefore, it was decided that to ensure that the network can prioritize the game traffic while still allowing other programs to also use the network when needed, the Cisco CBWFQ QoS queuing service is going to be used. Using it the network will have the ability to guarantee that the game traffic always has a minimum amount of bandwidth when congestions occur, but also will allow other traffic to flow freely when the congestion clears up.

2.3.3 Local cache server

A cache server is a dedicated server used for locally storing certain files and data so that the users don't need to access the outside network to download them. O'Reilly (2016, chapter 8) mentions that the transfer of large amounts of data from various storage servers can, in many cases, be slow due too poor WAN infrastructure. This type of server helps reduce the bandwidth used up at the network edge and can therefore speed up the downloads of certain large files which would otherwise need to be downloaded from outside the local network. Cache servers usually work like proxy servers since all of the requests to the outside network first go through them. If a file that is already cached is requested, the server redirects the request to the server itself. Otherwise, if the request is for a file that the server does not have, the server redirects the request to the outside network to be downloaded. The downloaded file first goes into the cache server itself and only then goes to the machine which made the request. Because of this once another device makes a request for the same file, the file can then be downloaded from the cache server directly. This minimizes the bandwidth to the outside network since most files only need to be downloaded once and then can be accessed by everyone locally.

Cache servers are commonly used in various e-sports events. During the events, the servers help reduce the load on the network and improve network stability to outside servers. Depending on the events and their requirements, different software implementation can be used. However, the main ways to run cache server are by using either the “Squid” or “LanCache.NET” platforms.

Squid allows the caching of various internet webpages and supports HTTP, HTTPS, FTP and many other protocols. The Squid (no date) website states that by using this service it is possible to cache whole webpages and even files locally. “Squid” works on various UNIX based operating systems but also can

work with Windows. The service by default stores all its cached data for three days however the timeframe can be adjusted.
LanCache.NET is a service which specializes in the caching of game downloads. Since game downloads are usually big in size this service can drastically reduce the strain on the network by housing the downloads locally. According to the LanCache.NET (no date) website, the service supports downloading and caching of games from most major game distributors like: Steam (Valve), Origin (EA Games), Riot Games, Battle.net, Uplay (Ubisoft). By default, this service stores its downloads for two years. However, because of this, once the maximum disk capacity on the server is reached, the service overrides the oldest cached downloads with newer ones. LanCache.NET runs in a docker container, which allows it to be run on a variety of operating systems, while also minimizing the system resources needed to run it.

Because this project mainly focuses on e-sports events, it was decided that a LanCache.NET cache server will be used. The main reason it was chosen is because it supports downloading and caching game downloads from all the major distributors which could be used in an e-sports event. It is also free, open-sourced and constantly updated by the community. However, running it will require the installation of the Docker virtualization software on the server.

The official Docker (no date) website sates that Docker is a virtualization software with which it is possible to pack an entire virtual machine into a container that only has the main functionalities, libraries and other operating system parts needed for the virtual machine to run. Docker containers are independent from the OS they are run on and as such can be deployed on most operating systems.

Once a docker container has been tested and packed it will always run as originally intended since its environment will remain the same between deployments. In that sense docker containers are similar to regular virtual machines. However, compared to a regular virtual machine, a Docker container uses less resources and as such can run more efficiently on the host machine.

2.4 Network security

Security is one of the most important parts of any network. A secure network is almost always more stable and safer for end users when compared to a non-secured one. Because the network in this project would only be a separate segment and have a NAT translating its IPs to one address that eventually goes through whatever security measures the colleges network has, in the end most of the traffic will still have all the restrictions any other computer connected to the college has. However, ensuring as much security as possible in that separate segment is still important. According to Denise (2015), when considering network security, it is best practice to use the OSI (Open Systems Interconnection) model and go from the bottom layer to the top layer and check if your network has any major vulnerabilities. The OSI model separates the network into seven separate layers. However, since in e-sports events most users bring their own computers to the event, the network can effectively only be secured up to the transport layer, because anything above it usually deals with the software running on computers and any bugs or vulnerabilities found in them.

OSI Model
Application
Presentation
Session
Transport
Network
Data Link
Physical

Figure 6 OSI model

Physical layer

The physical layer deals with the physical connections and devices that are part of the network. The biggest security concern at this layer is someone physically tampering with the network devices or trying to connect unauthorized devices to the network without the administrators knowing about it.

Physically securing the network devices is a great way to ensure security at this layer. This can be done by monitoring the network equipment and keeping network devices in an area where non-authorized personnel cannot reach them. Optimally all of the network devices should be in a centralized area, where they are easily monitorable by the network administrators.

Data Link layer

The data link layer deals with frames and connects the physical layer to the network layer. In this layer all addressing is done via MAC addresses. The data link layer is mostly controlled by network switches. A lot of the security risks in this layer have to do with unauthorized devices trying to sniff out the network topology or trying to disrupt the traffic flow of the network by modifying the intended network topology. According to Callisma, et al. (2003) there are many ways to harden and secure a switch, one of the more common ways are by: authenticating users, securing ports and assigning switch port modes manually so they don’t dynamically change. Another common way to secure this layer is by segmenting the network into multiple VLAN domains so that certain attacks are localized in those segments and don’t spread to the rest of the network.

User authentication is a process which usually makes it so that a user must provide a password and username in order to connect to a device. This is especially useful for network devices since with authentication enabled only users that know the passwords to the devices can access them and use them to modify the network. Without any user authentication on the network devices anyone would have the ability to connect to the devices and start tampering with the network. This in turn could make the whole network unusable for everyone else or could even expose every user in the network to major security risks. On Cisco devices user authentication can be enabled by adding a username and password login system, which forces anyone trying to connect to the device to provide the correct login credentials. Adding an “enable” password is also a great way to strengthen the security, since the password is needed when wanting to access the global configuration mode in which it is possible to modify the switches configuration file.

Port security is another great way to provide extra security in the network. Port security is a process in which certain rules are applied to the switches' interfaces, which when broken block the interface from sending or receiving traffic. The main way port security works is by limiting unique MAC addresses each interface can have. This way of securing the network can prevent a lot of MAC table flooding attacks. MAC flooding attacks usually work by flooding the switches' MAC table with a large amount of new unique MAC addresses. Once the MAC table is full the switch starts acting like a hub. If this happens, any traffic destined to only one device gets broadcasted through the entire broadcast domain and sent out to every other device. This can lead to various eavesdropping attacks where users with malicious intent are able to receive packets not originally destined to them. However, this way of securing the network mainly works in environments where devices stay connected to the same ports. In e-sports environments users commonly hop from one place to another, changing the interfaces or even switches they are connected to. Because of this it is best to allow a big enough number of unique MAC addresses per interface. If the maximum number of unique MAC addresses on the interface is reached an administrator can simply reset the interface to allow for more unique address when needed.

Manually assigning switch port modes helps with securing the logical topology of the network. Switch port modes define if the port is an access port or a trunk port. Access ports are meant for connecting end devices to network devices, while trunk ports are meant for connecting network devices to one another. By default, most network devices automatically change their switch port modes depending on what device is connected to each interface. This however can lead to certain STP attacks. If a malicious user attaches a rogue switch to the network, he can cause traffic loops or even trick other users in to connecting to his switch. On top of that he might also be able to perform certain network snooping attacks by listening to the traffic usually only meant for network devices. Manually setting the switch port modes can prevent these malicious users from being able to perform these types of attacks. On Cisco devices each interface can be assigned to either the access or trunk mode. When a switch or similar network device is attached to an access port, the port automatically shuts down, blocking the user from performing any attacks. Doing this prevents them from messing with the logical topology of the network and harming other users.

Segmenting the network into multiple VLAN domains is also a common way to increase security in this layer, since by doing so an attack in one section of the network can be controlled and not spread to the other parts of the network. The most common type of attack this prevents is the broadcast storm attack. A broadcast storm can occur when too much traffic is being broadcasted in one broadcast domain. When this happens, the switch has a hard time keeping up with all of the frames being sent and starts dropping some of them. This in turn can lead to network slowdowns for the end users. Even though broadcast storm attacks are usually started by malicious users, they can also occur naturally when too much broadcast traffic is being sent in a big enough broadcast domain. The best way to prevent broadcast storms is by reducing the size of the broadcast domains. This is usually done by segmenting one domain into smaller ones by creating multiple VLANs.

DHCP snooping lets the network administrators specify where an authorized DHCP server is located. Without DHCP snooping enabled a malicious user could connect his own DHCP server to the network and trick the computers of other users in to using his server. When this happens, the attacker can redirect the other users into certain spoofed pages and trick them in to revealing their credentials on those sites. With DHCP spoofing enabled only certain ports in the network can send DHCP advertisements, preventing the attackers from being able to advertise their own fake DHCP servers.

Network layer

The network layer sends packets between routing devices, so the packets reach their desired destination. The main network devices in this layer are routers. Packets in this layer are addressed by their IP addresses. Network risks in this layer occur when unauthorized devices gain access to the routers or when they are able to access certain devices they aren’t supposed to. The security in this layer is handled by routers and firewall devices.

Creating ACLs or access control lists can help protect the network in this layer. ACLs work as rules that filter out certain traffic going through an interface. If the

ACL specifies that the traffic isn’t allowed, the router automatically drops it. This can be very useful when wanting to limit certain users from accessing parts of the network. One example of this is letting only certain IP addresses connect to network devices. This can add an extra security layer for protection against malicious users that want to access the network devices and modify their configuration files.

Transport layer

The transport layer is used to encapsulate the packets from the network and layer and prepare them for the session layer. In the transports layer addressing is done via ports. Network security risks in this layer happen when the packets from the network layer aren’t properly secured during their transportation. To secure the information in the packets various encryption methods are used so that people capturing packets cannot reconstruct the messages.

Using SSH for remote access of the network devices is a great way to ensure that the networks’ management traffic is sent out in an encrypted format. There are two main protocols that deal with remote access traffic: Telnet and SSH. It is always preferred to use the SSH protocol over the outdated Telnet protocol since the Telnet protocol does not encode its messages and sends them as plain text. SSH on the other hand encodes the information using the SHA256 algorithm. The algorithm makes sure that any attacker sniffing the tracking isn’t able to read what was being sent over through the network.

Securing the other layers would require access to the users’ computers. However, because in e-sports events users bring their own devices it is not possible to secure them properly, unless the users themselves make sure their computers have a working antivirus software. Also, since there is a large number of different applications and devices in an e-sports event the security measures must be flexible enough so that they don’t cause certain programs to stop working. That being said, the network in this project will still be used with a large number of users so security will be important.

3 PROJECT IMPLEMENTATION

Using the information gained from the background study part, it is finally possible to start the implementation and configuration process. However, the order in which the devices are prepared and configured is important. Doing everything in a logical and sound order helps keep track of what's done and what still needs to be added. Due to this, the implementation process will start from the topology of the network and move on to the actual configuration of the devices.

3.1 Logical layout of the network

The very first step when creating a network is to create its logical layout. A logical network diagram shows how each device in the network must be connected. Creating a logical network diagram is also important since it can help troubleshoot problems by making it more clear how the network is laid out.
The logical network diagram for this project was created with the limitations of the available network hardware in mind.

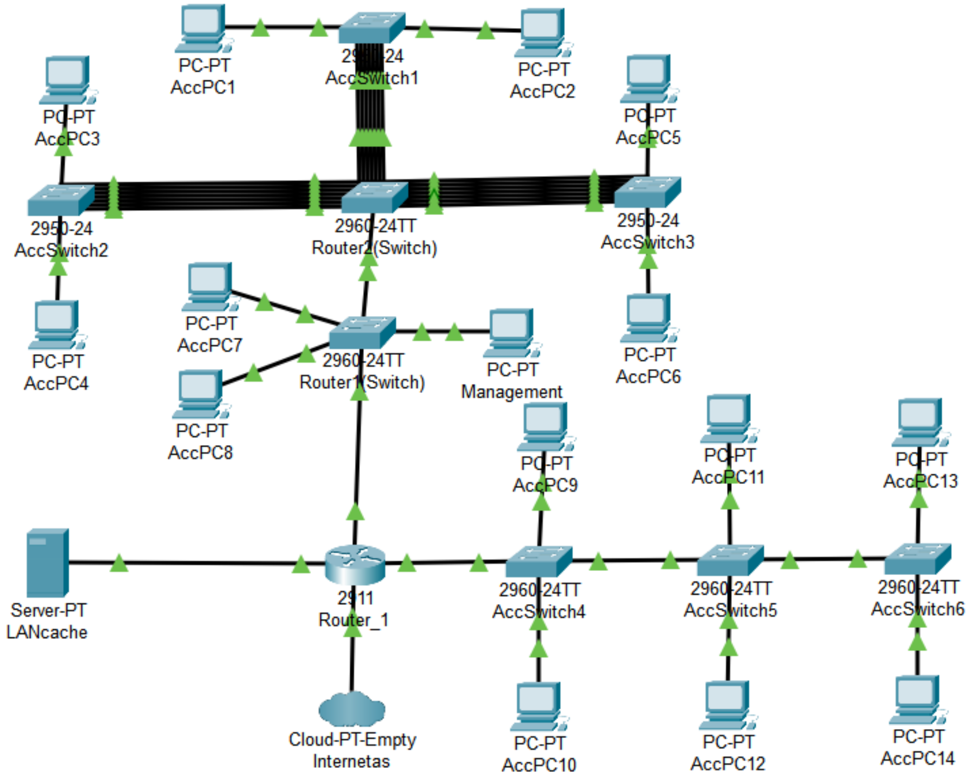


Figure 7 Logical scheme of the network

This network layout uses the expanded star topology. The layout was chosen because it maximizes the available network bandwidth with the available hardware. The Cisco 2911 router used in the network has three 1Gbps interfaces. Because of this it was chosen as the network edge device. One of its interfaces connects to the colleges' main network while the other interfaces are used for the internal network connections. One of these connections are to the cache server, while the other one is connected to a Cisco 2960 switch. The Router1(Switch) and Router2(Switch) are the EtherSwitch SM-ES2-24-P modules found in the Cisco 2911 routers. The modules are connected to the routers internally and have a 1Gbps bandwidth connection to the routers.
According to Cisco (2014), the Cisco 2960 switches have two 1Gbps interfaces. These interfaces are used to connect the switches to each other and to connect them to the router.

Because the Cisco 2950 switches only have 100Mbps interfaces it was decided to aggregate seven interfaces from each switch into an EtherChannel group. Even though each end user needs a minimum bandwidth of 4Mbps a common problem in e-sports events are large unexpected updates of games. With this in mind, seven interfaces were chosen to create the EtherChannel groups so that in the worst-case scenario the users have enough bandwidth available to download the updates in a reasonable amount of time. To connect all of these switches to the rest of the network one switch had to be used as a central switch. The Router_2's EtherSwitch module was chosen since it has it's 1Gbps interface connected to Router_1's EtherSwitch modules 1Gbps interface.

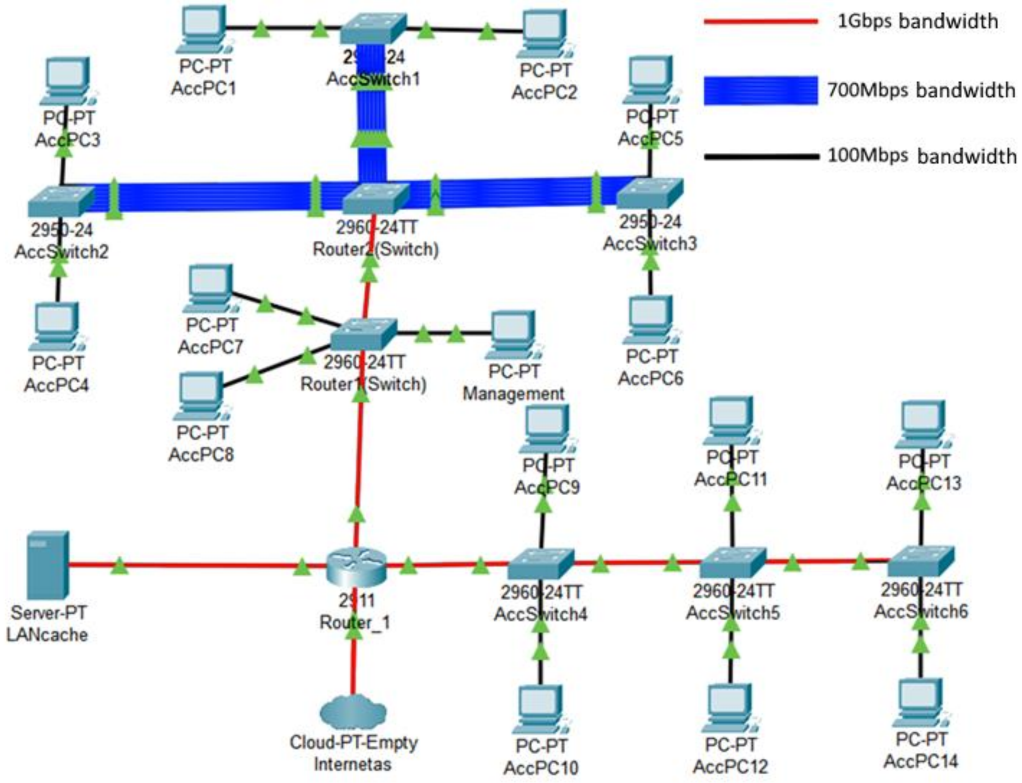


Figure 8 Maximum bandwidth in the logical scheme of the network

With this network layout, all of the available 1Gbps interfaces are used as efficiently as possible, and in those places where they aren’t available, EtherChannel links with a maximum of 700Mbps bandwidth are created to ensure enough bandwidth to the users. All the other 100Mbps are used to connect each of the end users to the network.

3.2 Physical layout of the network

After the networks logical layout has been planned out, it is possible to start the planning for the network’s physical layout. A poorly planned out physical layout can cause confusion when setting up the network and increase the time needed to track down and troubleshoot a problem.

The projects network needs to be able to be easily and quickly deployable. It also must be able to adapt to various physical layouts. Therefore, it was decided a physical layout template should be created. The switches are planned to be placed as close as possible to the tables of the end users. In these types of events, it is common for the user to bring their own device and cable. Cable lengths can vary a lot however the physical layout should still try to minimize the cable length the users need to have. Because of this it was decided that each user should not need a cable longer than 10m. To accomplish this, tables need to be put in rows and the switches need to be put in the middle of them. If each row is ten or less tables in length no user should need a cable length of more than 10m. The rows should also be placed in the shape of a circle or square so that the administrators of the network have the ability to oversee each device. Since only the switches need to be accessible to the users all other network devices should be put in the middle of the layout.

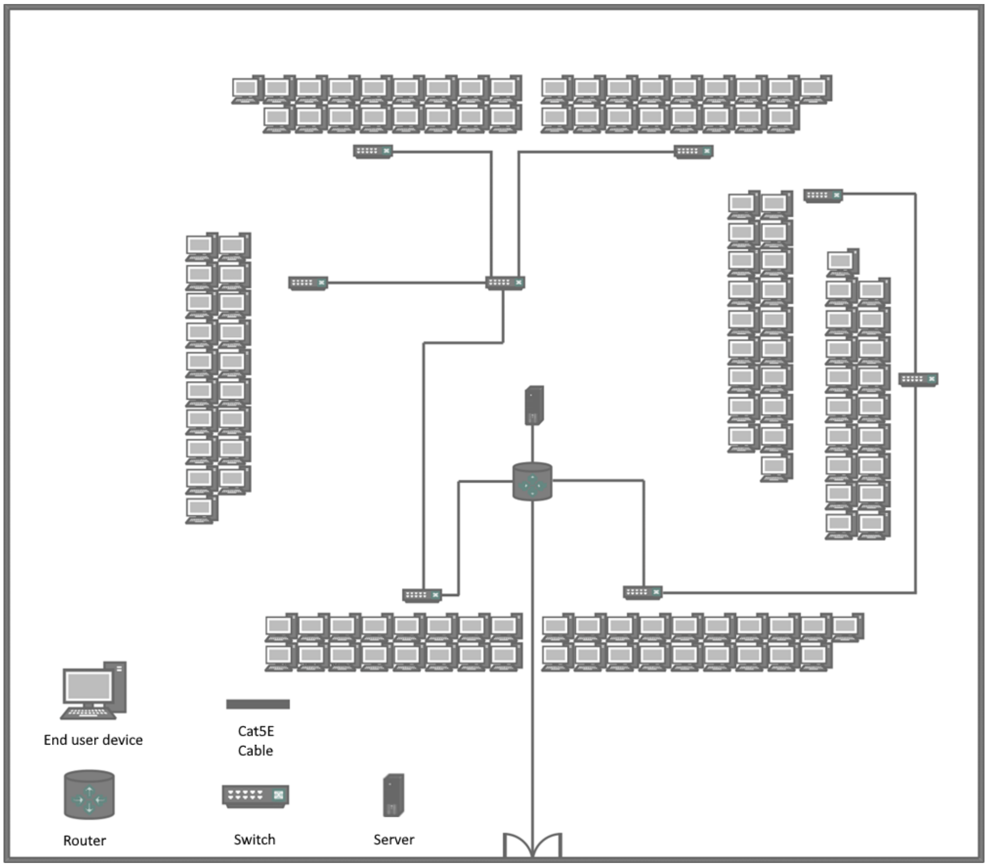


Figure 9 Physical scheme of the network

Since the place where the events might take place can change this layout might not workout every time. It is however a good reference on how to plan out the layout and modify it in places when needed.

3.3 Preparing the cache server

The cache server is an important part of this project. The server must be able to have a large enough storage capacity for all of the files, it also must have a network interface card capable of providing a high enough bandwidth. Thankfully, the HP P4500 G2 is capable of fulfilling these requirements.

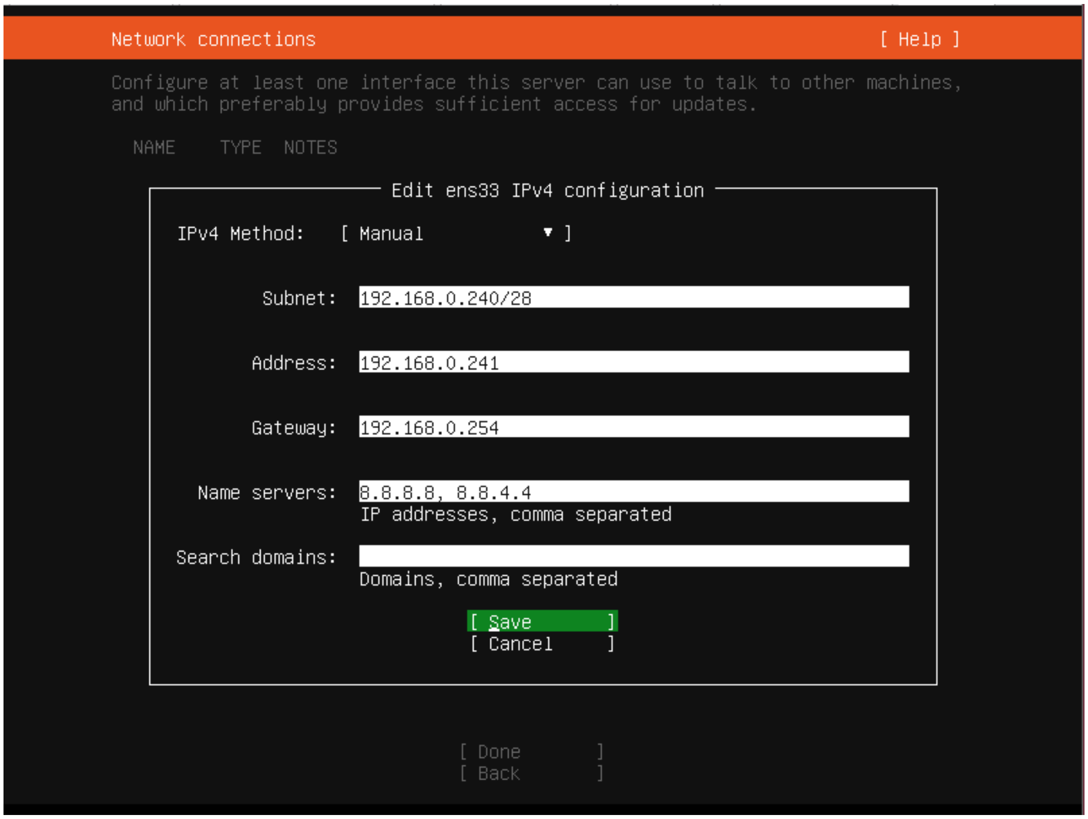
However, to make sure the server is capable enough to be use in the project the official LanCache.NET server recommendations were compared with the servers' specifications. These recommendations are made for an event with around 250 devices in mind.

Table 6 Minimum requirements comparison for the cache server

	Server specifications	**"LanCache.NET" recommendations**
Processor	E5520	-
Processor cores	4 physical, 8 logical	4 physical, 8 logical
Processor frequency	2.26GHz (Turbo 2.53GHz)	2.40GHz
RAM type	DDR3 ECC	-
RAM storage	6GB	2GB
Physical storage	12TB	500GB
Network interfaces	2 x 1Gbps	1 x 1Gbps

As can be seen from the table the server should have enough resources for this type of event. It is also worth mentioning that the server used as a reference in the LanCache.NET recommendation on average utilized only 20% of the processors processing power. Because of this even if the processor in the HP P4500 G2 server has a lower base clock, it shouldn't be a problem.

After making sure the server is capable of handling the projects needs, an operating system was installed into the server. The Ubuntu Server 18.04.4 LTS operating system was chosen. While installing the operating system most of the default configurations were left. Only the IPv4 configuration settings were changed.



```
Network connections                                        [ Help ]

Configure at least one interface this server can use to talk to other machines,
and which preferably provides sufficient access for updates.

  NAME    TYPE  NOTES

        Edit ens33 IPv4 configuration

  IPv4 Method:   [ Manual         ▾ ]

        Subnet:  192.168.0.240/28

       Address:  192.168.0.241

       Gateway:  192.168.0.254

  Name servers:  8.8.8.8, 8.8.4.4
                 IP addresses, comma separated

Search domains:
                 Domains, comma separated

                          [ Save     ]
                          [ Cancel   ]

                         [ Done      ]
                         [ Back      ]
```

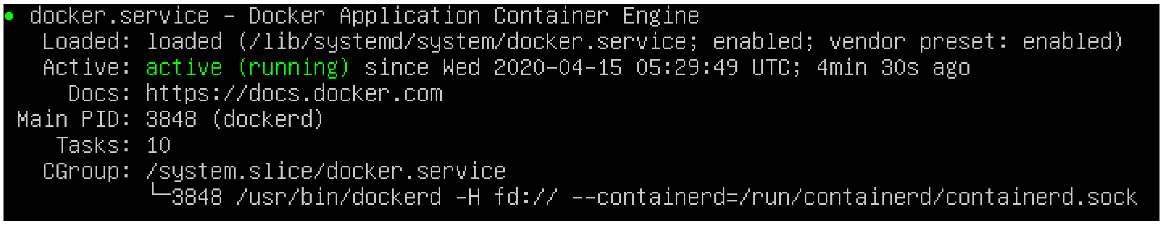
Figure 10 IP configuration on the Ubuntu installation

After configuring the IPv4 address and mask, the configuration was saved and the operating system continued its installation process. The password and username of the server was set to “cacheserver”. The password however should be changed to a more secure one when deploying in actual event.

Since being able to directly hook up a monitor and keyboard to the server might not always be possible SSH was also installed into the server. The “openssh-server” package was used. After installation, the **“systemctl enable ssh”** and **“systemctl start ssh”** commands were used to enable the SSH service.

3.4 Configuring Docker

To start working with Docker on Linux, the Docker repository needs to be added to the APT (advanced packing tool) list. After the addition it is possible to download the latest Docker version and containers straight from their official servers by using the **apt-get** command. Before all that the Docker GPG key needs to be imported. After the key has been imported, the **“sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable”** command can be used to add the Docker repository. Once the command has executed the **“sudo apt-get update”** needs to be used to refresh the now updated APT list. Finally using the **“sudo apt-get install -y docker-ce”** command is used to automatically download and install the Docker platform. After the installation, it was checked whether the Docker process is up and running.



```
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2020-04-15 05:29:49 UTC; 4min 30s ago
     Docs: https://docs.docker.com
 Main PID: 3848 (dockerd)
    Tasks: 10
   CGroup: /system.slice/docker.service
           └─3848 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

Figure 11 The status of the Docker service

Because Docker doesn’t have a GUI (graphical user interface), it can sometimes be hard to work with it. However, there are plenty of Docker containers which can add simple to use GUIs. For this project, the “Portainer” GUI container was chosen. To download the latest version of the container, the **“docker pull portainer/portainer”** command was issued. Once a container has downloaded, it can be run by using the **“docker run”** command with the containers name at the end. There are many flags which can be added to the “docker run” command to change the way the container runs.

-d flag can be used to run the container in the background.

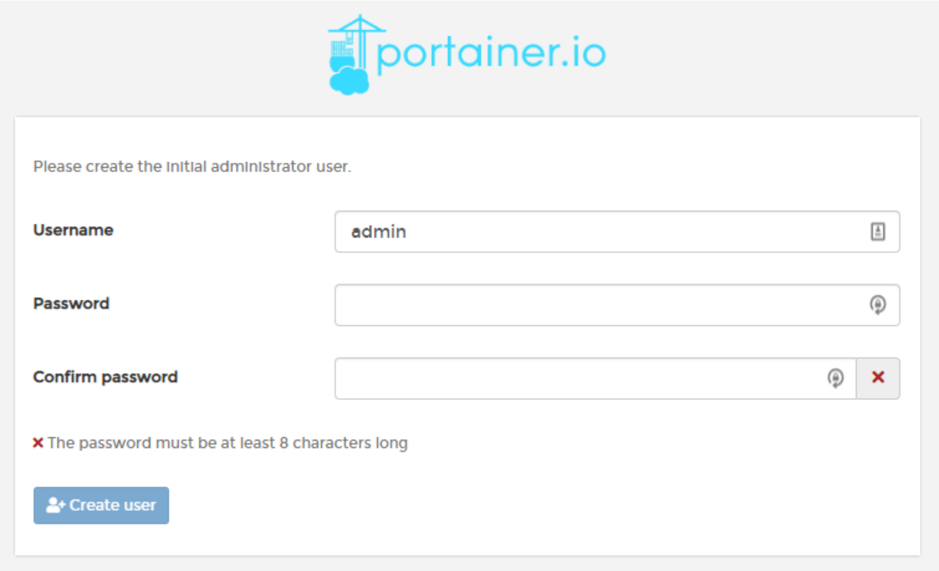
-p lets a custom IP address and port be specified for the container.

--restart specifies the action the container takes once it has unexpectedly shut down.

--name specifies the name the container will have.

-v flag is used to specify the mounting point of the container.

The container was run using the following command **<u>“docker run -d --restart unless-stopped --name Portainer -p 192.168.0.241:9000:9000 -v /var/run/docker.sock:/var/run/docker.sock portainer/portainer”</u>**. Using an internet browser, the 192.168.0.241:9000 IP address was entered to open the Portainer GUI.



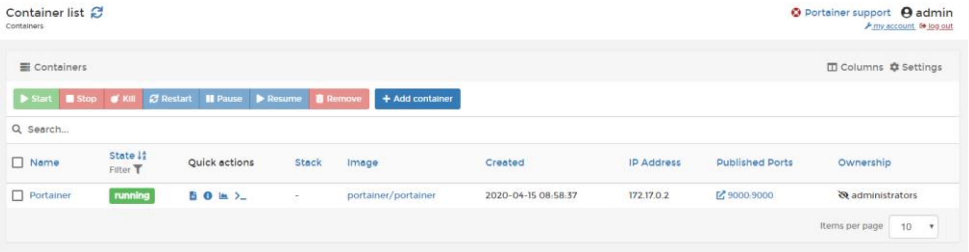
Please create the initial administrator user.

Username	admin
Password	
Confirm password	

✗ The password must be at least 8 characters long

Figure 12 Portainer login window

A secure password was entered and the “Create user” button was pressed to create the admin account. After signing in with the admin account the list of currently running containers was checked to see if everything is working as expected.



Name	State Filter	Quick actions	Stack	Image	Created	IP Address	Published Ports	Ownership
Portainer	running		-	portainer/portainer	2020-04-15 08:58:37	172.17.0.2	9000:9000	administrators

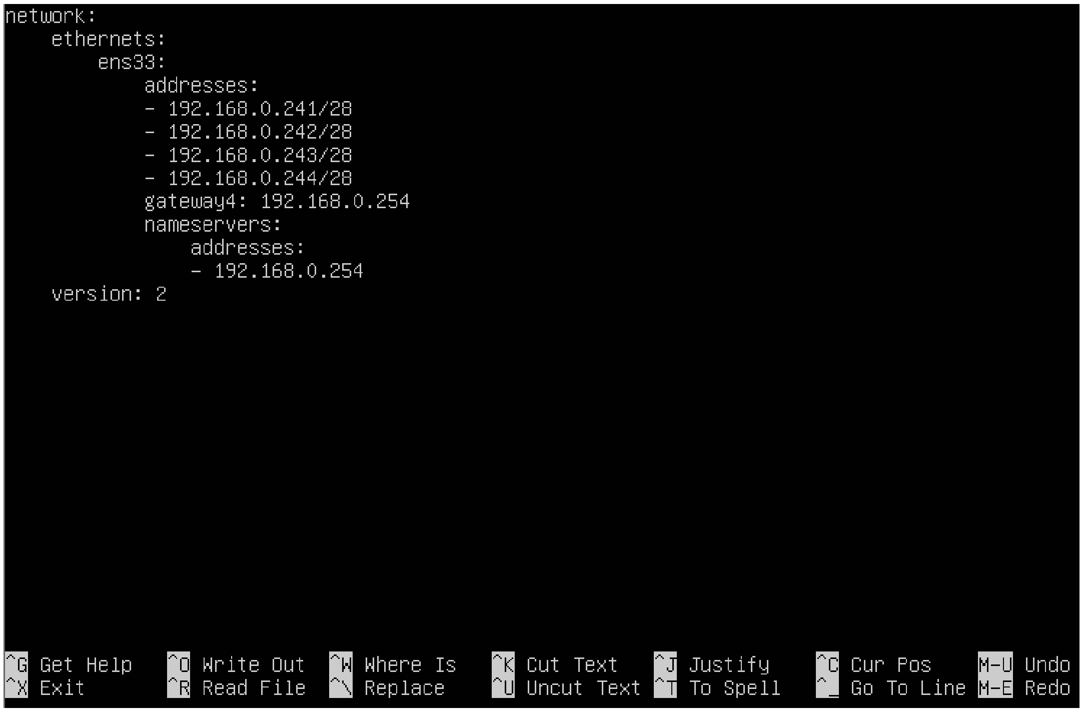
Figure 13 The active containers list after configuring Portainer

As was expected, only the Portainer container was running. With Portainer installed, configured and running it was possible to start the configuring the LanCache.NET container.

3.5 Configuring LanCache.NET

Before the cache server can start being configured the necessary Docker containers need to be downloaded. LanCache.NET consists of multiple modular components separated in multiple Docker containers. To download the main container the **“docker pull lancachenet/monolithic”** command is used and to start it the **“docker run -d --restart unless-stopped --name LanCache -v /cache/data:/data/cache -v /cache/logs:/data/logs -p:80:80 lancachenet/monolith”** is issued. With this command LanCache.NET is run on port 80 of the server.

However, because some game launchers use a relatively short HTTP protocol timeout window, LanCache.NET recommends adding at least four IP addresses to the pool that the server can pull from to download the games. Following these recommendations, the netplan configuration file was modified to have four addresses.



```
network:
    ethernets:
        ens33:
            addresses:
            - 192.168.0.241/28
            - 192.168.0.242/28
            - 192.168.0.243/28
            - 192.168.0.244/28
            gateway4: 192.168.0.254
            nameservers:
                addresses:
                - 192.168.0.254
    version: 2
```

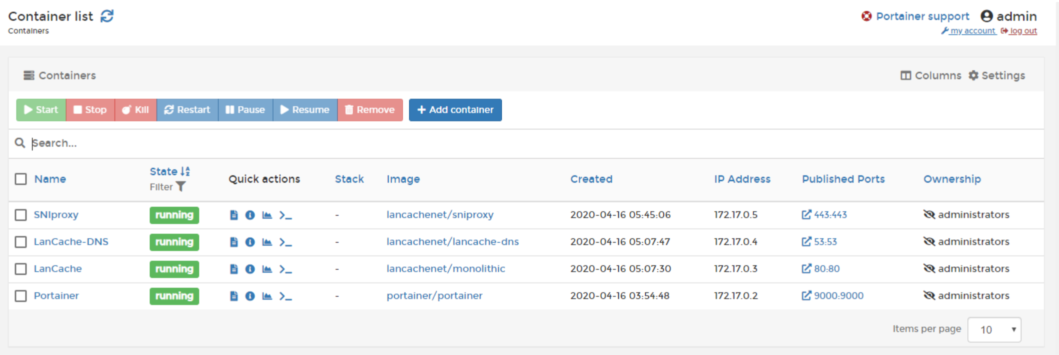
Figure 14 Netplan configuration file of the server

To apply the changes made to the file, the **“netplan apply”** command was issued.

For the LanCache.NET container to work as a cache server, HTTP requests, coming from inside the local network and destined to the outside network, need to be redirected to the server. According to the LanCache.NET recommendations, this is best achieved by changing the DNS server address on end user devices to the IP address of the cache server. However, this means that the cache server must now also act as a DNS server. Thankfully LanCache.NET has a separate Docker container configured for this purpose. To download and install the container the **“docker pull lancache-dns”** and **„docker run -d --restart unless-stopped --name LanCache-DNS -p 192.168.1.241:53:53/udp -e USE_GENERIC_CACHE=true -e LANCACHE_IP="192.168.1.241 192.168.1.242 192.168.1.243 192.168.1.244" lancachenet/lancache-dns“** commands are used with the previously added IP addresses. Once installed and run the container checks if it has received requests to a game publishers’ server. If the destination of a request matches the IP addresses of a game publishers’ server, the request is forwarded to the main container.

Many of the game launchers use HTTP and HTTPS requests to connect a user to their servers. According to LanCache.NET (no date) “When running a LAN Cache and overriding DNS entries, there are some services (including the Origin launcher) which will try and use HTTPS to talk to one of the hostnames that are being overridden”. Due to this LanCache.NET recommends using the “SNI Proxy” Docker container which redirects HTTPS request to the internet whole letting the HTTP request go to your cache server. To download the container the **“docker pull lancachenet/sniproxy”** command is used. To install it the **“docker run -d --restart unless-stopped --name SNIproxy -p 443:443 lancachenet/sniproxy”** command is entered.

To verify that all of the containers have been download and ran successfully, the Portainer GUI was used. Using the GUI the list of currently running containers was checked.

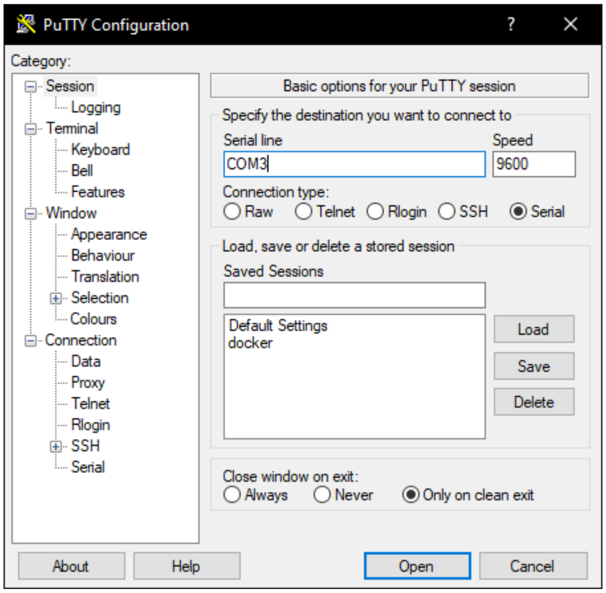


Name	State Filter	Quick actions	Stack	Image	Created	IP Address	Published Ports	Ownership
SNIproxy	running		-	lancachenet/sniproxy	2020-04-16 05:45:06	172.17.0.5	443:443	administrators
LanCache-DNS	running		-	lancachenet/lancache-dns	2020-04-16 05:07:47	172.17.0.4	53:53	administrators
LanCache	running		-	lancachenet/monolithic	2020-04-16 05:07:30	172.17.0.3	80:80	administrators
Portainer	running		-	portainer/portainer	2020-04-16 03:54:48	172.17.0.2	9000:9000	administrators

Figure 15 Portainer active container list after LanCache.NET was configured

3.6 Configuring the network devices

The network devices used in this project don’t have a graphical interface, only console ports. Because of this a computer had to be used to configure the devices. The console ports on the network devices used the RJ45 interface so a RJ45-to-USB adapter was also needed. Once the devices were connected to the computer a terminal emulating program (PuTTY) was used to interface with the devices. PuTTY was chosen because it is free, easy to use and tested by many people. Once the program was launched the correct serial port was selected, COM3 in this case, and the computer was able to interface with the network devices.



Specify the destination you want to connect to

Serial line: COM3

Speed: 9600

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

Saved Sessions

Default Settings
docker

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 16 PuTTY user interface

3.6.1 Configuring VLANs

Because the network in this project has a lot of devices constantly communicating with one another there is a high possibility that they could create a “broadcast storm” scenario. If this happens a significant part of the network’s resources could be quickly used up. To avoid this, it was decided to separate the network into multiple VLAN domains. Two VLANs per switch were decided to be used for a total of fifteen VLAN domains, fourteen for end users and one for administrators.

Table 7 List of VLANs in the network

VLAN name	VLAN ID	Switch	Number of interfaces in the VLAN
AccPC1	11	AccSwitch1	9
AccPC2	12		8
AccPC3	13	AccSwitch2	9
AccPC4	14		8
AccPC5	15	AccSwitch3	9
AccPC6	16		8
AccPC7	17	Router1(Switch)	9
AccPC8	18		9
Management	30		5
AccPC9	19	AccSwitch4	12
AccPC10	20		12
AccPC11	21	AccSwitch5	12
AccPC12	22		12
AccPC13	23	AccSwitch6	12
AccPC14	24		12

To help with this task two VTP (VLAN trunking protocol) groups were also created. Using VTP it is possible to configure the VLAN database on one device and have that device send it’s VLAN database information to all the other devices in the same VTP group. VTP works in a client, server mode, where the server device propagates it’s VLAN database to all of the client devices.

To create a VTP group the **“vtp domain”** command was entered in the global configuration mode of a switch with the name of the group. Next the **“vtp password”** command was used with a secure password for the group. Lastly for the switches used as VTP servers the **“vtp mode server”** command was used and for the clients the **“vtp mode client”** command was issued.

After that, the interfaces of the switches were manually changed to access mode with the **“switchport mode access vlan”** command with the VLAN ID. After all the commands were entered all of the VLAN groups were assigned to their interfaces.

3.6.2 Configuring the physical interfaces

Since the projects network has a router connected to the colleges outside network, some of the router interfaces need to have IP addresses assigned to them. The GigabitEthernet0/0 interface on Router_1 is connected to the colleges network, while the GigabitEthernet0/2 interface is connected to the cache server.

Table 8 IP configuration of the physical interfaces

Device	**Interface**	**Address of the interface**
Router_1	GigabitEthernet0/2	192.168.0.246
	GigabitEthernet0/0	10.4.13.69/24

To assign IP addresses to the interfaces the **“ip address”** command was used with an address and subnet mask. By default, Cisco routers have all of their interfaces shutdown. Because of this after assigning the IP addresses the **“no shutdown”** command was issued on the interfaces.

3.6.3 Configuring the virtual interfaces

Because there are fifteen VLAN groups in the network, they need to be assigned to virtual interfaces to be able to communicate with one another. These interfaces will be created using the dot1q standard. Before creating these types of interfaces, it is best to temporarily disable the physical interface. After the interface has been shutdown the virtual interface configuration mode is entered. Once there the **“encapsulation dot1q”** command is used with the VLAN ID that will correspond to the virtual interface. After that, the gateway address is assigned to the interface. These steps are repeated for all of the other VLAN groups which need to be connected to the router.

Table 9 IP configuration of the virtual interfaces

Device	**Physical interface**	**Virtual interface**	**Address of the interface**
Router_1	GigabitEthernet0/1	GigabitEthernet0/1.19	192.168.0.142/28
		GigabitEthernet0/1.20	192.168.0.174/28
		GigabitEthernet0/1.21	192.168.0.158/28
		GigabitEthernet0/1.22	192.168.0.190/28
		GigabitEthernet0/1.23	192.168.0.206/28
		GigabitEthernet0/1.24	192.168.0.222/28
		GigabitEthernet0/1.34	192.168.1.14/30
		GigabitEthernet0/1.35	192.168.1.18/30
		GigabitEthernet0/1.36	192.168.1.22/30

For the VLAN groups that are only connected to the SM-ES2-24-P EtherSwitch module SVI(switch virtual interfaces) need to be created. They can be created by entering the "**interface vlan"** command with the VLAN ID used for the interface in the global configuration mode. After that, the gateway IP addresses are assigned to each interface.

Table 10 IP configuration of the VLAN interfaces

Device	**Virtual interface**	**Address of the interface**
Router_1	Vlan11	192.168.0.14/28
	Vlan12	192.168.0.30/28
	Vlan13	192.168.0.46/28
	Vlan14	192.168.0.62/28
	Vlan15	192.168.0.78/28
	Vlan16	192.168.0.94/28
	Vlan17	192.168.0.110/28
	Vlan18	192.168.0.126/28
	Vlan30	192.168.0.238/28
	Vlan31	192.168.1.2/30
	Vlan32	192.168.1.6/30
	Vlan33	192.168.1.10/30
	Vlan37	192.168.1.26/30
	Vlan38	192.168.1.30/30

Finally, the EtherChannel groups needed for the Cisco 2950 switches are created. Each switch uses seven physical interfaces to create one virtual logical interface. To create these groups all of the interfaces used for the EtherChannel are selected and the **“channel-group 1 mode active”** command is entered. Before this step it is also important to check that all of the interfaces in the group have identical configurations.

Table 11 EtherChannel configuration

First device	Interfaces of the first device	Second device	Interfaces of the second device
Router2(Switch)	FastEthernet0/1	AccSwitch1	FastEthernet0/18
	FastEthernet0/2		FastEthernet0/19
	FastEthernet0/3		FastEthernet0/20
	FastEthernet0/4		FastEthernet0/21
	FastEthernet0/5		FastEthernet0/22
	FastEthernet0/6		FastEthernet0/23
	FastEthernet0/7		FastEthernet0/24
	FastEthernet0/8	AccSwitch2	FastEthernet0/18
	FastEthernet0/9		FastEthernet0/19
	FastEthernet0/10		FastEthernet0/20
	FastEthernet0/11		FastEthernet0/21
	FastEthernet0/12		FastEthernet0/22
	FastEthernet0/13		FastEthernet0/23
	FastEthernet0/14		FastEthernet0/24
	FastEthernet0/15	AccSwitch3	FastEthernet0/18
	FastEthernet0/16		FastEthernet0/19
	FastEthernet0/17		FastEthernet0/20
	FastEthernet0/18		FastEthernet0/21
	FastEthernet0/19		FastEthernet0/22
	FastEthernet0/20		FastEthernet0/23
	FastEthernet0/21		FastEthernet0/24

3.6.4 Configuring DHCP

Because the network has a Cisco 2911 router, it was decided to have a DHCP server running on it. Due to there being fifteen VLAN groups, the DHCP server will need to have a separate pool for each group. To create a DHCP pool on a Cisco router the **“ip dhcp pool”** command with the name of the pool was entered in the global configuration mode. Next the **“network”** command with an IP address and subnet mask was entered. After that using the **“default-router”** command the gateway was specified. Lastly the **“dns-server”** command with the address of the cache server was issued along with the “domain-name” command used for defining the name of the domain.

Table 12 DHCP configuration

Router	Address and mask	Gateway	Domain name	DNS
Router_1	192.168.0.0/28	192.168.0.14	AccPC1	192.168.0.241
	192.168.0.16/28	192.168.0.30	AccPC2	192.168.0.241
	192.168.0.32/28	192.168.0.46	AccPC3	192.168.0.241
	192.168.0.48/28	192.168.0.62	AccPC4	192.168.0.241
	192.168.0.64/28	192.168.0.78	AccPC5	192.168.0.241
	192.168.0.80/28	192.168.0.94	AccPC6	192.168.0.241
	192.168.0.96/28	192.168.0.110	AccPC7	192.168.0.241
	192.168.0.112/28	192.168.0.142	AccPC8	192.168.0.241
	192.168.0.144/28	192.168.0.158	AccPC9	192.168.0.241
	192.168.0.160/28	192.168.0.126	AccPC10	192.168.0.241
	192.168.0.128/28	192.168.0.174	AccPC11	192.168.0.241
	192.168.0.176/28	192.168.0.190	AccPC12	192.168.0.241
	192.168.0.192/28	192.168.0.206	AccPC13	192.168.0.241
	192.168.0.208/28	192.168.0.222	AccPC14	192.168.0.241
	192.168.0.224/28	192.168.0.238	Management	192.168.0.241

3.6.5 Routing

Because the network consists of multiple switches and only one router, all of the routes needed for communications in the inside network are gathered automatically. However, to allow the network to communicate with the rest of the colleges network a route must be added. A default route was added using the **“ip route”**.

Table 13 Routing configuration

Router	Address and mask	Next hop address
Router_1	0.0.0.0/0	10.4.13.1

3.6.6 Configuring NAT

NAT (network access translation) needs to be configured because the projects network and the colleges network are separated and use different IP ranges. To not cause IP address conflicts with the colleges network a many to one translation will be used. Specifically, NAT overload or simply PAT (port address translation). Using this type of NAT, a senders IP address is changed to the address of the outgoing interface on the router. A port is also assigned to the now modified address to make it possible to distinguish it from other addresses. The original address and the translated one are then stored inside the routers NAT table, so it is later possible to find out which request needs to go to which address. To enable this type of NAT translation first the inside and outside interfaces need the be selected. The **“ip nat inside”** command is used for the inside interface and the **“ip nat outside”** is used for the outside one. Next an ACL (access control list) is created with list of addresses that need to be translated.

Finally, the **"ip nat inside source list x interface y overload"** command, where x is the number of the ACL list and y is the outside interface, is entered o enable the NAT translations.

Table 14 NAT translations configuration

IP NAT in interface	IP NAT out interface
Vlan11	GigabitEthernet0/0
Vlan12	
Vlan13	
Vlan14	
Vlan15	
Vlan16	
Vlan17	
Vlan18	
GigabitEthernet0/1.19	
GigabitEthernet0/1.20	
GigabitEthernet0/1.21	
GigabitEthernet0/1.22	
GigabitEthernet0/1.23	
GigabitEthernet0/1.24	
Vlan30	

3.6.7 Remote access of the network devices

During an e-sports event it might not always be possible to go directly to a network device and connect to it via its console port. However, being able to monitor and access the network devices is still important. Because of this it was decided to enable remote access on the devices. Accessing Cisco devices remotely is possible using either the Telnet or SSH (secure shell) protocol. For this project the SSH protocol was chosen since it encrypts its messages before sending them.

In total it was decided to create five different users that the administrators can use. This was chosen because in an event with one hundred people, one person wouldn't be enough to make sure the network is running stabile. To enable remote access on a Cisco device a user with a password needs to be created. This can be done by using the **"username x password y"** command, where x is the username and y is the password. Next the **"login local"** command Is issued to force devices wanting to connect via the VTY lines to provide a username and password. By using the **“hostname”** command each network device was assigned a unique name. An enable password was also added by using the “enable secret” command with the password. The **“transport in ssh”** and **“transport out ssh”** commands were entered on the VTY lines to force them to use SSH instead of Telnet. Lastly In the global configuration mode the **“ip ssh version 2”** command was issued and a 1024-bit key was generated.

After configuring the VTY lines, it was also necessary to assign each network device an IP address that could be used for connecting to it. This was done by creating virtual interfaces. The **“ip default-gateway”** command with an address of the gateway was also used so that all of the network devices could be reached from anywhere in the network.

Table 15 IP addresses and enable passwords for the remote access of the network hardware

Device	IP address for the remote interface	enable password
AccSwitch1	192.168.1.1/30	5l4pt4z0d151
AccSwitch2	192.168.1.5/30	5l4pt4z0d152
AccSwitch3	192.168.1.9/30	5l4pt4z0d153
AccSwitch4	192.168.1.13/30	5l4pt4z0d154
AccSwitch5	192.168.1.17/30	5l4pt4z0d155
AccSwitch6	192.168.1.21/30	5l4pt4z0d156
Router_1(Switch)	192.168.1.25/30	5l4pt4z0d157
Router_2(Switch)	192.168.1.29/30	5l4pt4z0d158
Router_1	-	5l4pt4z0d159

Table 16 Username and password configuration for the remote access of the network devices

Username	Password
LANAdmin1	P45l4pt151
LANAdmin2	P45l4pt152
LANAdmin3	P45l4pt153
LANAdmin4	P45l4pt154
LANAdmin5	P45l4pt155

The passwords shown here are only used as examples. In an actual network they should be longer and more complex. They should also be unique and should not share repeating parts with one another.

3.6.8 Configuring QoS

Since the bandwidth at the link connecting the projects inside network to the colleges outside network is only 1Gbps, congestion can occur during peak traffic moments. To avoid or mitigate this a QoS was decided to be implemented. The QoS will be able to prioritize game traffic during moments of network congestion, while still allowing the traffic to flow normally when the congestion ends. Games usually communicate to outside servers through various UDP (user datagram protocol) and TCP (transmission control protocol) ports. However, predicting the ports for every single game that might be played in an event is impossible. Because of this the games that are most likely to be played in an e-sports event were decided to have at least half of the network bandwidth during congestion. Guaranteeing them half of the bandwidth, makes sure that other games, not on the list, still have a stable enough connection during moments of network congestion. Since being able to download or view information on webpages is also important, some bandwidth was also assigned to HTTP, HTTPS, and FTP protocols.

In total four different groups are created, one for game traffic, another one for web traffic, third one for downloads, and the last one for miscellaneous traffic. For QoS configurations Cisco uses the Modular QoS Command-Line Interface. To start the **“class-map”** command is used in the global configuration mode to define what type of traffic is in each class.

Next once the type of traffic has been defined, the **“policy-map”** command is used to group all of the previously created classes into one group. Lastly the **“service*policy”** command is used on the interfaces which will use these QoS settings.

Table 17 QoS configuration

policy-map	class-map	Application	TCP port	UDP port	Minimum bandwidth %
gamelan	games	Overwatch	1119, 3724, 6113	5060, 5062, 6250, 3478-3479, 12000-64000	50
		Counter-Strike: Global Offensive	27015-27030, 27036-27037	4380, 27000-27031, 27036	
		Street Fighter	27015-27030, 27036-27037	4380, 27000-27031, 27036	
		Dota 2	27015-27030, 27036-27037	4380, 27000-27031, 27036	
		Rocket League	27015-27030, 27036-27037	4380, 7000-9000, 27000-27031, 27036	
		Fortnite	5222, 5795-5847	5222, 5795-5847	
		Call of Duty: Black Ops	3074, 28910, 29900-29901, 29920	3074-3075	
		League of Legends	2099, 5222-5223, 8088, 8393-8400	5000-5500, 8088	
	web	HTTP, HTTP	80. 443	80, 443	10
	ftp	FTP	21	21	15

3.6.9 Securing the network

Since during e-sports events it is common for people to bring their own devices, some of them can have malicious software on them. Since modifying the devices of participants is impossible, various ways to secure the network at the first four levels of the OSI model will be used to mitigate certain attacks the network might experience.

To protect the network physically most of the network devices will be placed in the middle of the event where the administrators can watch over them. Some type of barrier or restriction should also be placed so that certain individuals cannot access the hardware without being seen crossing the barrier.

At the data link layer various switch configurations will be applied. One of these is to split up the network into smaller chunks by using multiple VLAN groups. Once the network is segmented, if there are malicious attacks coming from a certain segment of the network, it can be easily isolated, so it does not affect the rest of the network. The process with which the VLANs were created is described in chapter 3.6.1 named “Configuring VLANs”.

Switch port modes were also manually set on all of the interfaces to secure logical topology of the network. Doing so prevented certain participants from connecting network devices that could change the logical topology of the network in a way that could harm other users or the network itself. Using the **“switchport mode access”** command all of the interfaces intended for the end users were set to access mode. Similarly, using the **“switchport mode trunk”** command all of the interfaces used for connecting network devices to other network devices were set to the trunking mode. Additionally, the **“spanning-tree bpduguard enable”** command was also issued on the access ports to disable them if bpdu packets are being sent through them. All other unused interfaces were shutdown.

To make sure no one runs a fake DHCP server, DHCP snooping was also enabled. To enable and configure DHCP snooping the **“ip dhcp snooping”** command was used in the global configuration mode. This command enables

DHCP snooping on the switch. After that, the **“ip dhcp snooping vlan”** command was entered with all of the end user VLANs on that particular switch. Finally, the **“ip dhcp snooping trust”** command was used on the trusted interfaces leading up to the DHCP server. With all of this no rouge DHCP servers can advertise themselves to the network.

A maximum amount of IP addresses per interface was also implemented. Since during an event there could be a lot of different participants changing the places where their computers are at, a limit of fifteen unique addresses per interfaces was added. This was done by firstly enabling port security on a particular interface by using the **“switchport port-security”** command. Next the **“switchport port-security maximum 15”** command was used to limit the maximum number of unique MAC address on that port to only fifteen. This was then repeated on all of the access ports intended for the end users. In the event that an interface has used up all of the MAC addresses, an administrator can just reset the interface, so the unique MAC address counter starts again from zero.

Authentication on the network devices was added so that only users that know the required passwords and usernames have the ability to access the network devices and make changes to their configuration files. The steps and process for doing so was described in the 3.6.7 chapter titled “Remote access of the network devices”.

To strengthen the security of the network devices an ACL was created allowing only certain users to access the network devices. The ACL only lets the computers in the management VLAN to connect to the network devices remotely, any other user trying to connect from anywhere else in the network will be immediately denied. To create the ACL the **“access-list permit 30 192.168.0.224 0.0.0.15”** command was used. The command specifies that a number access list with the number 30 is created and permits traffic from the 192.168.0.224 0.0.0.15 subnet. The subnet is used by network administrators for management. After the list has been created it was applied to the VTY lines. Using the **“line vty 0 15”** command all of the VTY lines were selected. After that, the **“access-class 30 in”** command was entered to apply the ACL to the lines.

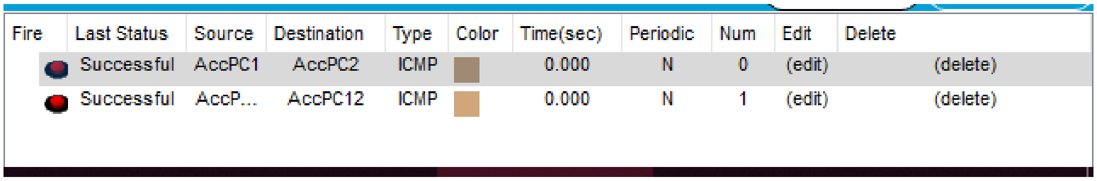
For the transport layer, only SSH traffic was allowed to be used for accessing network devices remotely. Using the Telnet protocol would allow anyone with a network traffic sniffing program to see the passwords and usernames needed to connect to network devices, since Telnet does not encrypt its messages. The steps and process for doing so was described in the 3.6.7 chapter titled “Remote access of the network devices”.

4 TESTING

The configuration and testing of the network was done by using the Cisco Packet Traces simulation program. The configuration files are included in the appendixes. The testing of the cache server was done by using two computers, one acting as the server and the other one acting as the client. The configurations done in Packet Tracer and on the cache server are analogous to the ones that would be found in the final network.

4.1 Device connectivity

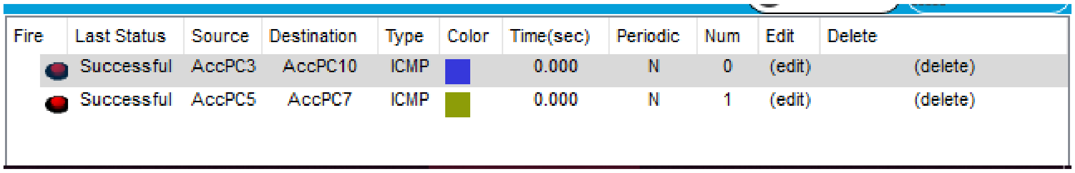
To test weather all of the devices in the network are able to communicate with one another, four tests were made. The first one checked weather two devices connected to the same switch but assigned to different VLAN groups can ping each other. The second test checked weather devices connected to different switches can communicate. The third test checked weather devices in the network can reach the cache server. The last test checked weather devices in the inside network can reach devices on the outside network.



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	AccPC1	AccPC2	ICMP	🟫	0.000	N	0	(edit)	(delete)
●	Successful	AccP...	AccPC12	ICMP	🟫	0.000	N	1	(edit)	(delete)

Figure 17 First PING test

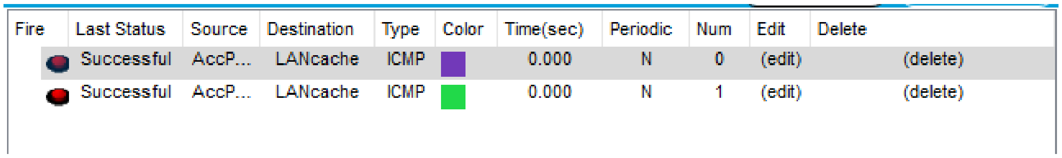
In the first test two different devices pinged another pair of devices that were connected on the same switch but were assigned to different VLAN groups. The test passed successfully and the devices were able to ping one another.



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	AccPC3	AccPC10	ICMP	🟦	0.000	N	0	(edit)	(delete)
●	Successful	AccPC5	AccPC7	ICMP	🟩	0.000	N	1	(edit)	(delete)

Figure 18 Second PING test

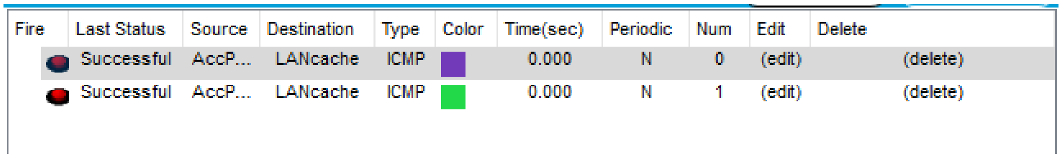
The second test checked if devices connected to different switches can ping each other. Just like in the first test, two random devices were chosen from each switch. This test also passed.



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	AccP...	LANcache	ICMP	🟪	0.000	N	0	(edit)	(delete)
●	Successful	AccP...	LANcache	ICMP	🟩	0.000	N	1	(edit)	(delete)

Figure 19 Third PING test

On the third test the connectivity to the cache server was tested. Two random devices in the network tried pining the server and both were successful in doing so.



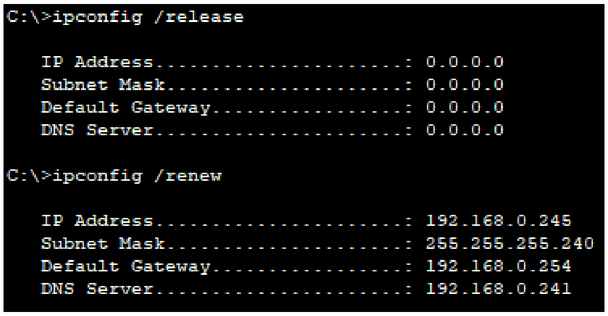
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	AccP...	LANcache	ICMP	🟪	0.000	N	0	(edit)	(delete)
●	Successful	AccP...	LANcache	ICMP	🟩	0.000	N	1	(edit)	(delete)

Figure 20 Fourth PING test

Finally, the last test checked whether devices on the inside network can reach devices on the outside network. Again, two random devices were tested and both of them passed like in the previous tests.

4.2 Testing DHCP

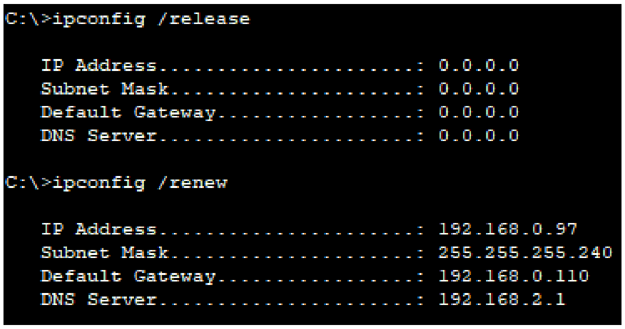
To test weather the DHCP service is running correctly, two random devices on the network were tested. Both of them had their IP information deleted and then were checked if they can get an IP assigned to them via the DHCP server.



```
C:\>ipconfig /release

   IP Address......................: 0.0.0.0
   Subnet Mask.....................: 0.0.0.0
   Default Gateway.................: 0.0.0.0
   DNS Server......................: 0.0.0.0

C:\>ipconfig /renew

   IP Address......................: 192.168.0.245
   Subnet Mask.....................: 255.255.255.240
   Default Gateway.................: 192.168.0.254
   DNS Server......................: 192.168.0.241
```

Figure 21 First DHCP test

The first device was able to successfully get a new IP address after the old one was released.



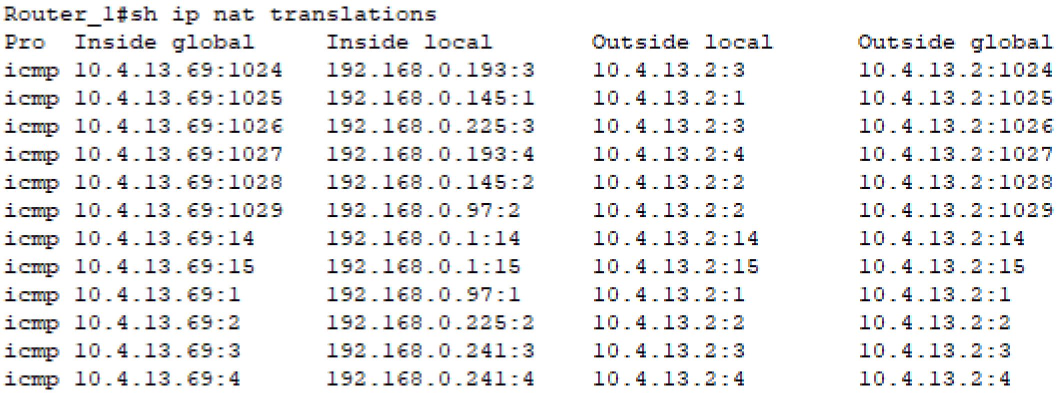
```
C:\>ipconfig /release

   IP Address......................: 0.0.0.0
   Subnet Mask.....................: 0.0.0.0
   Default Gateway.................: 0.0.0.0
   DNS Server......................: 0.0.0.0

C:\>ipconfig /renew

   IP Address......................: 192.168.0.97
   Subnet Mask.....................: 255.255.255.240
   Default Gateway.................: 192.168.0.110
   DNS Server......................: 192.168.2.1
```

Figure 22 Second DHCP test

The second device was also able to get a new IP address assigned to it, after the first one was released.

4.3 NAT translations

The test to check weather NAT translations work, was done on Router_1. The 10.4.13.2 address in this test was an address belonging to a device on the outside network. Different devices pinged the outside network device and then the NAT translation table was checked on the router.



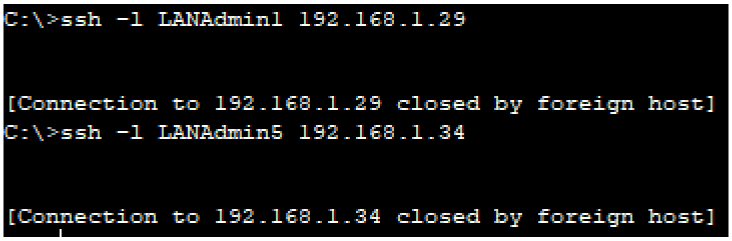
```
Router_1#sh ip nat translations
Pro  Inside global      Inside local       Outside local      Outside global
icmp 10.4.13.69:1024    192.168.0.193:3    10.4.13.2:3        10.4.13.2:1024
icmp 10.4.13.69:1025    192.168.0.145:1    10.4.13.2:1        10.4.13.2:1025
icmp 10.4.13.69:1026    192.168.0.225:3    10.4.13.2:3        10.4.13.2:1026
icmp 10.4.13.69:1027    192.168.0.193:4    10.4.13.2:4        10.4.13.2:1027
icmp 10.4.13.69:1028    192.168.0.145:2    10.4.13.2:2        10.4.13.2:1028
icmp 10.4.13.69:1029    192.168.0.97:2     10.4.13.2:2        10.4.13.2:1029
icmp 10.4.13.69:14      192.168.0.1:14     10.4.13.2:14       10.4.13.2:14
icmp 10.4.13.69:15      192.168.0.1:15     10.4.13.2:15       10.4.13.2:15
icmp 10.4.13.69:1       192.168.0.97:1     10.4.13.2:1        10.4.13.2:1
icmp 10.4.13.69:2       192.168.0.225:2    10.4.13.2:2        10.4.13.2:2
icmp 10.4.13.69:3       192.168.0.241:3    10.4.13.2:3        10.4.13.2:3
icmp 10.4.13.69:4       192.168.0.241:4    10.4.13.2:4        10.4.13.2:4
```

Figure 23 Results of the NAT translations test

Using the “show ip nat translations” command the NAT translations table was gotten. According to the table the translations work as intended.

4.4 Testing remote access

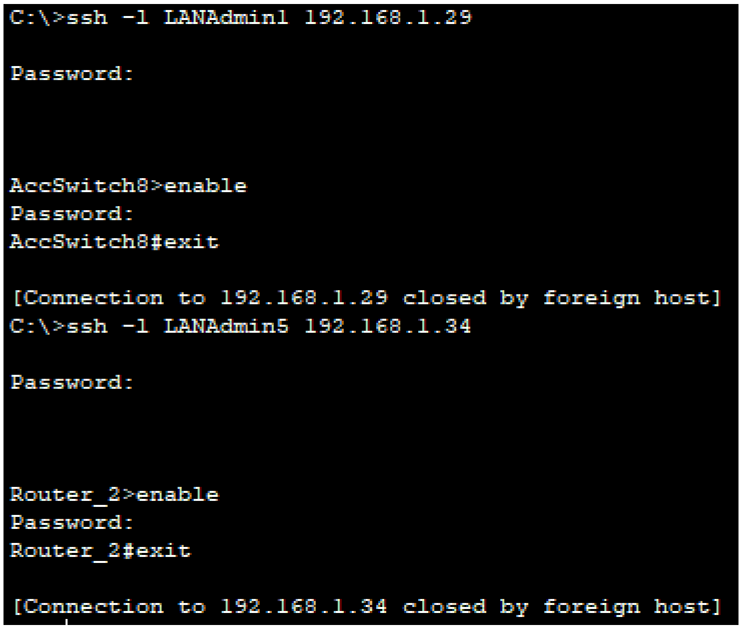
To test weather accessing the network devices remotely is possible, two different devices tried connecting one of the switches. The remote access configuration is the same on all switches, so these results are analogous to all other devices. The test was done by trying to connect to a switch from a device outside the management VLAN (VLAN 25) and then from devices belonging to the management VLAN (VLAN 30).



```
C:\>ssh -l LANAdmin1 192.168.1.29


[Connection to 192.168.1.29 closed by foreign host]
C:\>ssh -l LANAdmin5 192.168.1.34


[Connection to 192.168.1.34 closed by foreign host]
```

Figure 24 First remote access of the network devices test

The first test was done with a device not belonging to the management VLAN. The switch immediately closed the connection after the attempt was made.



```
C:\>ssh -l LANAdmin1 192.168.1.29

Password:



AccSwitch8>enable
Password:
AccSwitch8#exit

[Connection to 192.168.1.29 closed by foreign host]
C:\>ssh -l LANAdmin5 192.168.1.34

Password:



Router_2>enable
Password:
Router_2#exit

[Connection to 192.168.1.34 closed by foreign host]
```

Figure 25 Second remote access of the network devices test

The second tried connecting to the same switch but this time from devices belonging to the management VLAN. The computers were able to remotely connect to the switch successfully.

4.5 Testing the cache server

To test weather the cache server works, a few games were downloaded and installed from different game launchers and automatically cached. After the games were installed, they then were immediately uninstalled. After this step once again, the games were downloaded and installed and the time needed was checked between the first download and the second one.

Table 18 Results of the cache server test

Launcher	Game	Download size	First download time	Second download time
Steam	Neverwinter	132Mb	0:1:15s	0:0:13s
EpicGames	Enter The Gungeon	238Mb	0:2:48s	0:0:29s
Steam	Worms Crazy Golf	516Mb	0:6:36s	0:0:46s
Origin	Command & Conquer: Red Alert 2	1.54Gb	0:18:21s	0:5:53s

The test showed that in some cases the second download took almost eight times less then the first download. But overall, the second download on all of the tested games took less then the first one. The cache server statistics were also checked using the Portainer GUI to see visually how after the first download, the games were served directly from the cache server.

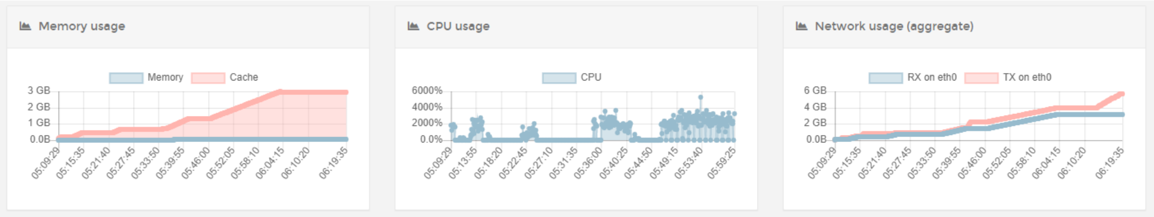


Figure 26 Statistics of the running cache server

4.6 Testing the QoS

To test weather the QoS configuration worked, multiple packets with different ports were sent to the outside network. The packets were sent from devices located in the inside network. After the packets were sent the “show policy-map interface” command was used with outside interface.

```
Router_1#show policy-map interface gi0/0
GigabitEthernet0/0

Service-policy output: gamelan

Class-map: games (match-any)
20 packets, 92424 bytes
5 minute offered rate 2425 bps, drop rate 0 bps
Match: access-group name csgo
20 packets, 92424 bytes
5 minute rate 2425 bps
Match: access-group name overwatch
0 packets, 0 bytes
5 minute rate 0 bps
Match: access-group name dota2
0 packets, 0 bytes
5 minute rate 0 bps
Match: access-group name fortnite
0 packets, 0 bytes
5 minute rate 0 bps
Match: access-group name streetfighter
0 packets, 0 bytes
5 minute rate 0 bps
Match: access-group name cod2
0 packets, 0 bytes
5 minute rate 0 bps
Match: access-group name rocketleague
0 packets, 0 bytes
5 minute rate 0 bps
Match: access-group name lol
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
Output Queue: Conversation 265
Bandwidth 50 (%)
Bandwidth 500000 (kbps)Max Threshold 64
(packets)
(pkts matched/bytes matched) 0/0
(depth/total drops/no-buffer drops) 0/0/0
Class-map: ftp (match-any)
17 packets, 748 bytes
5 minute offered rate 34 bps, drop rate 0 bps
Match: protocol ftp
17 packets, 748 bytes
5 minute rate 34 bps
Queueing
Output Queue: Conversation 266
Bandwidth 15 (%)
Bandwidth 150000 (kbps)Max Threshold 64
(packets)
(pkts matched/bytes matched) 0/0
(depth/total drops/no-buffer drops) 0/0/0

Class-map: web (match-any)
18 packets, 792 bytes
5 minute offered rate 36 bps, drop rate 0 bps
Match: access-group name web
18 packets, 792 bytes
5 minute rate 36 bps
Match: protocol https
0 packets, 0 bytes
5 minute rate 0 bps
Match: protocol http
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
Output Queue: Conversation 267
Bandwidth 10 (%)
Bandwidth 100000 (kbps)Max Threshold 64
(packets)
(pkts matched/bytes matched) 0/0
(depth/total drops/no-buffer drops) 0/0/0

Class-map: class-default (match-any)
182 packets, 6260 bytes
5 minute offered rate 300 bps, drop rate 0 bps
Match: any
Queueing
Flow Based Fair Queueing
Maximum number of Hashed Queues 256
Bandwidth 750000 (kbps)Max Threshold 64
(packets)
(total queued/total drops/no-buffer drops) 0/0/0
```

Figure 27 Results of the QoS test

All of the different class maps had matches when packets matched the access list. However, because the test was done using Cisco Packet Tracer it was not possible to accurately test how the network acts during high load moments.

5 CONCLUSION

The network needed to host e-sports events in Kaunas college was successfully designed. The network devices used in the project were all from the available hardware in the college that is normally used for teaching purposes. Having this in mind the network was designed in a way to maximize the capabilities of the hardware, while minimizing bottlenecks in the network. However, the network devices used were not ideal. All of the switches can only operate in the second OSI layer and so cannot route layer three traffic between multiple VLAN domains. On top of that, some of the switches only had 100Mbps interfaces and so limited the maximum bandwidth available in some segments of the network. To deal with this multiple EtherChannel groups were configured on some of the devices. The EtherChannel links should provide enough bandwidth for those segments to be able to operate without the users noticing any major slowdowns.

A logical and physical network scheme was also created. Using the schemes, deploying and tracking down problems in the network should be easier. From the logical scheme it was determined that the network can support up to 141 devices. The devices were configured in a way that minimizes possible network bottlenecks and security risks.

QoS was configured to prioritize game traffic during times of network congestion, while allowing the traffic to flow normally when the congestion clears up. The games that the QoS prioritizes currently only has a handful of popular tournament games, however if the need arises the list can be easily expandable.

A cache server was also created and configured to minimize congestion at the networks' edge. The server automatically downloads games that people are requesting, but it also can be pre-filled with all of the expected games before the event starts. However, the cache server also could use some upgrades. Currently the cache server is only made up of HDD type disks. The speed of HDDs is not optimal for the needs of the network and SSDs would be much more useful in this situation.

For the security, all of the interfaces on the switches were manually set so that rouge switches can't modify the logical layout of the network. The ability to deploy rouge DHCP servers was also removed by making the switches only send out DHCP request from the permitted ports. A way for administrators to access network devices remotely was also added to allow them to monitor or modify devices during an event. To secure the network devices an ACL was created, letting only people in the management VLAN to access them. On top of that each device was protected with an enable password and a simple login.

In conclusion, the network accomplishes all of the requirements it needs to host e-sports events with over one hundred users. The QoS and cache server should minimize congestion of the network, while the way the network was set up should provide enough bandwidth to allow adequate traffic flow. Since the computers used in these kinds of events cannot be known, the security aspect of the network has a few weak spots. However, the configuration of the network devices should minimize most of them.

REFERENCES

Adeolu, O. No date. Traffic Policing vs. Traffic Shaping. Blog. Available at: https://www.routerfreak.com/traffic-policing-vs-traffic-shaping/ [Accessed 23 March 2021]

Alison, Q. 2008. A Guide to Network Topology. Blog. Available at: https://www.itprc.com/a-guide-to-network-topology/ [Accessed 24 March 2021]

Barreiros, M. & Lundqvist, P. 2016. QOS-Enabled Networks, 2nd Edition . Wiley. Available at: https://ebookcentral.proquest.com/lib/xamk-ebooks/reader.action?docID=4205854[Accessed 23 March 2021]

Bradley, M. 2021. What Causes Network Lag and How to Fix It. Blog. Available at: https://www.lifewire.com/lag-on-computer-networks-and-online-817370 [Accessed 23 March 2021]

Callisma Inc Staff, Flannagan, ME, Riley, C. & Syngress. 2003. The Best Damn Cisco Internetworking Book Period . Burlington: Syngress. Available at: https://ebookcentral.proquest.com/lib/xamk-ebooks/reader.action?docID=294395&query=The+Best+Damn+Cisco+Internetworking+Book+Period [Accessed 25 March 2021]

Cisco. 2006. Cisco Catalyst 2950 Series Switches with Cisco Standard Image and Enhanced Image. WWW document. Available at: https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-2950-series-switches/prod_qas09186a008009258e.html [Accessed 26 March 2021]

Cisco. 2014. Cisco Catalyst 2960-S and 2960 Series Switches with LAN Lite Software Data Sheet. WWW document. Available at: https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-2960-series-switches/product_data_sheet0900aecd806b0bd8.html [Accessed 26 March 2021]

Cisco. 2017. Cisco 800M Series ISR Software Configuration Guide. WWW document. Available at: https://www.cisco.com/c/en/us/td/docs/routers/access/800M/software/800MSCG/QoS.html#35645 [Accessed 26 March 2021]

Denise, G. B. 2015. Strengthening the different layers of IT networks. Blog. Available at: https://www.welivesecurity.com/2015/06/30/strengthening-the-different-layers-of-it-networks/ [Accessed 27 March 2021]

Docker. No date. Docker homepage. WWW document. Available at: https://www.docker.com/ [Accessed 27 March 2021]

Jim, D. 2012. Dissecting Cisco's FabricPath Ethernet Technology. Blog. Available at: https://www.cio.com/article/2397679/dissecting-cisco-s-fabricpath-ethernet-technology.html [Accessed 27 March 2021]

Joel, K. 2012. 4 Types of Port Channels and When They're Used. Blog. Available at: https://ourtechplanet.com/common-types-of-port-channels/ [Accessed 27 March 2021]

John, D. 2021. How Much Speed You Need for Online Gaming. Blog. Available at: https://www.highspeedinternet.com/resources/how-much-speed-do-i-need-for-online-gaming [Accessed 27 March 2021]

LanCache.NET. No date. LAN Party game caching made easy. WWW document. Available at: https://lancache.net/ [Accessed 28 March 2021]

O’Reilly, J. 2016. Network Storage . Morgan Kaufmann. Available at: https://ebookcentral.proquest.com/lib/xamk-ebooks/reader.action?docID=4718032 [Accessed 28 March 2021]

Router-Switch. No date. SM-ES2-24-P Datasheet. PDF document. Available at: https://www.router-switch.com/pdf/sm-es2-24-p-datasheet.pdf [Accessed 28 March 2021]

Squid. No date. Squid: Optimising Web Delivery. WWW document. Available at: http://www.squid-cache.org/ [Accessed 28 March 2021]

Sterbenz, JPG, Chapin, AL, Escobar, J., Krishnan, R., Qiao, C. & Touch, JD 2001. High Speed Network Design: A Systematic Approach to High-Bandwidth Low-Latency Communication . New York: John Wiley & Sons, Inc. Available at: http://ebookcentral.proquest.com/lib/xamk-ebooks/detail.action?docID=120274 [Accessed 28 March 2021]

LIST OF FIGURES

LIST OF TABLES

Switch AccSwitch1 configuration (1/2), analogous to AccSwitch2 and AccSwitch3 configuration

```
hostname AccSwitch1
enable secret 5 $1$mERr$aKi.yADhvNZ1wkhT1Q34L.
ip ssh version 2
ip ssh time-out 10
ip domain-name kauko.com
username LANAdmin1 privilege 1 password 0 P45l4pt151
username LANAdmin2 privilege 1 password 0 P45l4pt152
username LANAdmin3 privilege 1 password 0 P45l4pt153
username LANAdmin4 privilege 1 password 0 P45l4pt154
username LANAdmin5 privilege 1 password 0 P45l4pt155
spanning-tree mode pvst
spanning-tree extend system-id
interface Port-channel1
switchport mode trunk
interface FastEthernet0/1
switchport access vlan 11
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/2
switchport access vlan 11
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/3
switchport access vlan 11
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/4
switchport access vlan 11
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/5
switchport access vlan 11
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/6
switchport access vlan 11
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/7
switchport access vlan 11
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/8
switchport access vlan 11
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/9
switchport access vlan 11
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/10
switchport access vlan 12
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/11
switchport access vlan 12
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/12
switchport access vlan 12
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/13
switchport access vlan 12
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/14
switchport access vlan 12
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/15
switchport access vlan 12
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/16
switchport access vlan 12
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/17
switchport access vlan 12
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/18
switchport mode trunk
channel-group 1 mode active
interface FastEthernet0/19
switchport mode trunk
channel-group 1 mode active
interface FastEthernet0/20
switchport mode trunk
```

Switch AccSwitch1 configuration (2/2), analogous to AccSwitch2 and AccSwitch3 configuration

```
channel-group 1 mode active
interface FastEthernet0/21
switchport mode trunk
channel-group 1 mode active
interface FastEthernet0/22
switchport mode trunk
channel-group 1 mode active
interface FastEthernet0/23
switchport mode trunk
channel-group 1 mode active
interface FastEthernet0/24
switchport mode trunk
channel-group 1 mode active
interface Vlan1
no ip address
shutdown
interface Vlan31
mac-address 0001.c9e6.9001
ip address 192.168.1.1 255.255.255.252
ip default-gateway 192.168.1.2
access-list 30 permit 192.168.0.224 0.0.0.15
line con 0
line vty 0 4
access-class 30 in
login local
transport input ssh
transport output ssh
line vty 5 15
no login
transport input none
transport output none
```

Switch Router2(Switch) configuration (1/1)

```
hostname Router2(Switch)
enable secret 5 $1$mERr$aahwvwmECPnKEL5iS/VxO.
ip ssh version 2
ip ssh time-out 10
ip domain-name kauko.com
username LANAdmin1 privilege 1 password 0 P45l4pt151
username LANAdmin2 privilege 1 password 0 P45l4pt152
username LANAdmin3 privilege 1 password 0 P45l4pt153
username LANAdmin4 privilege 1 password 0 P45l4pt154
username LANAdmin5 privilege 1 password 0 P45l4pt155
spanning-tree mode pvst
spanning-tree extend system-id
interface Port-channel1
switchport mode trunk
interface Port-channel2
switchport mode trunk
interface Port-channel3
switchport mode trunk
interface FastEthernet0/1
switchport mode trunk
channel-group 1 mode active
interface FastEthernet0/2
switchport mode trunk
channel-group 1 mode active
interface FastEthernet0/3
switchport mode trunk
channel-group 1 mode active
interface FastEthernet0/4
switchport mode trunk
channel-group 1 mode active
interface FastEthernet0/5
switchport mode trunk
channel-group 1 mode active
interface FastEthernet0/6
switchport mode trunk
channel-group 1 mode active
interface FastEthernet0/7
switchport mode trunk
channel-group 1 mode active
interface FastEthernet0/8
switchport mode trunk
channel-group 2 mode active
interface FastEthernet0/9
switchport mode trunk
channel-group 2 mode active
interface FastEthernet0/10
switchport mode trunk
channel-group 2 mode active
interface FastEthernet0/11
switchport mode trunk
channel-group 2 mode active
interface FastEthernet0/12
switchport mode trunk
channel-group 2 mode active
interface FastEthernet0/13
switchport mode trunk
channel-group 2 mode active
interface FastEthernet0/14
switchport mode trunk
channel-group 2 mode active
interface FastEthernet0/15
switchport mode trunk
channel-group 3 mode active
interface FastEthernet0/16
switchport mode trunk
channel-group 3 mode active
interface FastEthernet0/17
switchport mode trunk
channel-group 3 mode active
interface FastEthernet0/18
switchport mode trunk
channel-group 3 mode active
interface FastEthernet0/19
switchport mode trunk
channel-group 3 mode active
interface FastEthernet0/20
switchport mode trunk
channel-group 3 mode active
interface FastEthernet0/21
switchport mode trunk
channel-group 3 mode active
interface FastEthernet0/22
switchport mode trunk
shutdown
interface FastEthernet0/23
switchport mode trunk
shutdown
interface GigabitEthernet0/1
switchport mode trunk
interface GigabitEthernet0/2
ip dhcp snooping trust
switchport mode trunk
interface Vlan38
mac-address 0006.2a30.5a01
ip address 192.168.1.29 255.255.255.252
ip default-gateway 192.168.1.30
access-list 30 permit 192.168.0.224 0.0.0.15
line con 0
line vty 0 4
access-class 30 in
login local
transport input ssh
transport output ssh
line vty 5 15
no login
transport input none
transport output none
```

Switch Router1(Switch) configuration (1/2)

```
hostname Router1(Switch)
enable secret 5 $1$mERr$B9UzHiw6SQspyp9IxZu.u.
ip ssh version 2
ip ssh time-out 10
ip domain-name kauko.com
username LANAdmin1 privilege 1 password 0 P45l4pt151
username LANAdmin2 privilege 1 password 0 P45l4pt152
username LANAdmin3 privilege 1 password 0 P45l4pt153
username LANAdmin4 privilege 1 password 0 P45l4pt154
username LANAdmin5 privilege 1 password 0 P45l4pt155
spanning-tree mode pvst
spanning-tree extend system-id
interface FastEthernet0/1
switchport access vlan 17
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/2
switchport access vlan 17
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/3
switchport access vlan 17
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/4
switchport access vlan 17
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/5
switchport access vlan 17
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/6
switchport access vlan 17
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/7
switchport access vlan 17
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/8
switchport access vlan 17
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/9
switchport access vlan 17
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/10
switchport access vlan 18
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/11
switchport access vlan 18
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/12
switchport access vlan 18
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/13
switchport access vlan 18
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/14
switchport access vlan 18
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/15
switchport access vlan 18
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/16
switchport access vlan 18
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/17
switchport access vlan 18
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/18
switchport access vlan 18
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/19
switchport access vlan 30
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
```

Switch Router1(Switch) configuration (2/2)

```
interface FastEthernet0/20
switchport access vlan 30
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/21
switchport access vlan 30
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/22
switchport access vlan 30
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/23
switchport access vlan 30
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/24
switchport mode trunk
shutdown
interface GigabitEthernet0/1
switchport mode trunk
interface GigabitEthernet0/2
switchport mode trunk
interface Vlan1
no ip address
shutdown
interface Vlan37
mac-address 0004.9ab0.b201
ip address 192.168.1.25 255.255.255.252
ip default-gateway 192.168.1.26
access-list 30 permit 192.168.0.224 0.0.0.15
line con 0
line vty 0 4
access-class 30 in
login local
transport input ssh
transport output ssh
line vty 5 15
no login
transport input none
transport output none
```

Switch AccSwitch4 configuration (1/2), analogous to AccSwitch5 and AccSwitch6 configuration

```
hostname AccSwitch4
enable secret 5 $1$mERr$aQoLVRniNMZslQ1uUrOE8/
ip ssh version 2
ip ssh time-out 10
ip domain-name kauko.com
username LANAdmin1 privilege 1 password 0 P45l4pt151
username LANAdmin2 privilege 1 password 0 P45l4pt152
username LANAdmin3 privilege 1 password 0 P45l4pt153
username LANAdmin4 privilege 1 password 0 P45l4pt154
username LANAdmin5 privilege 1 password 0 P45l4pt155
ip dhcp snooping vlan 25,30
no ip dhcp snooping information option
spanning-tree mode pvst
spanning-tree extend system-id
interface FastEthernet0/1
switchport access vlan 19
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/2
switchport access vlan 19
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/3
switchport access vlan 19
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/4
switchport access vlan 19
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/5
switchport access vlan 19
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/6
switchport access vlan 19
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/7
switchport access vlan 19
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/8
switchport access vlan 19
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/9
switchport access vlan 19
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/10
switchport access vlan 19
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/11
switchport access vlan 19
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/12
switchport access vlan 19
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/13
switchport access vlan 20
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/14
switchport access vlan 20
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/15
switchport access vlan 20
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/16
switchport access vlan 20
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/17
switchport access vlan 20
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/18
switchport access vlan 20
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/19
switchport access vlan 20
switchport mode access
```

Switch AccSwitch4 configuration (2/2), analogous to AccSwitch5 and AccSwitch6 configuration

```
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/20
switchport access vlan 20
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/21
switchport access vlan 20
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/22
switchport access vlan 20
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/23
switchport access vlan 20
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface FastEthernet0/24
switchport access vlan 20
switchport mode access
spanning-tree portfast
spanning-tree bpduguard enable
interface GigabitEthernet0/1
switchport mode trunk
interface GigabitEthernet0/2
switchport mode trunk
interface Vlan1
no ip address
shutdown
interface Vlan34
mac-address 0003.e429.8d01
ip address 192.168.1.13 255.255.255.252
ip default-gateway 192.168.1.14
access-list 30 permit 192.168.0.224 0.0.0.15
line con 0
line vty 0 4
access-class 30 in
login local
transport input ssh
transport output ssh
line vty 5 15
no login
transport input none
transport output none
```

Router Router_1 configuration (1/3)

```
hostname Router_1
enable secret 5 $1$mERr$PUGIkgRUt81n9HpUmAQeu/
ip dhcp pool AccPC1
network 192.168.0.0 255.255.255.240
default-router 192.168.0.14
dns-server 192.168.2.1
domain-name AccPC1
ip dhcp pool AccPC2
network 192.168.0.16 255.255.255.240
default-router 192.168.0.30
dns-server 192.168.2.1
domain-name AccPC2
ip dhcp pool AccPC3
network 192.168.0.32 255.255.255.240
default-router 192.168.0.46
dns-server 192.168.2.1
domain-name AccPC3
ip dhcp pool AccPC4
network 192.168.0.48 255.255.255.240
default-router 192.168.0.62
dns-server 192.168.2.1
domain-name AccPC4
ip dhcp pool AccPC5
network 192.168.0.64 255.255.255.240
default-router 192.168.0.78
dns-server 192.168.2.1
domain-name AccPC5
ip dhcp pool AccPC6
network 192.168.0.80 255.255.255.240
default-router 192.168.0.94
dns-server 192.168.2.1
domain-name AccPC6
ip dhcp pool AccPC7
network 192.168.0.96 255.255.255.240
default-router 192.168.0.110
dns-server 192.168.2.1
domain-name AccPC7
ip dhcp pool AccPC8
network 192.168.0.112 255.255.255.240
default-router 192.168.0.126
dns-server 192.168.2.1
domain-name AccPC8
ip dhcp pool AccPC9
network 192.168.0.128 255.255.255.240
default-router 192.168.0.142
dns-server 192.168.2.1
domain-name AccPC9
ip dhcp pool AccPC10
network 192.168.0.144 255.255.255.240
default-router 192.168.0.158
dns-server 192.168.2.1
domain-name AccPC10
ip dhcp pool AccPC11
network 192.168.0.160 255.255.255.240
default-router 192.168.0.174
dns-server 192.168.0.241
domain-name AccPC11
ip dhcp pool AccPC12
network 192.168.0.176 255.255.255.240
default-router 192.168.0.190
dns-server 192.168.0.241
domain-name AccPC12
ip dhcp pool AccPC13
network 192.168.0.192 255.255.255.240
default-router 192.168.0.206
dns-server 192.168.0.241
domain-name AccPC13
ip dhcp pool AccPC14
network 192.168.0.208 255.255.255.240
default-router 192.168.0.222
dns-server 192.168.0.241
domain-name AccPC14
ip dhcp pool Management
network 192.168.0.224 255.255.255.240
default-router 192.168.0.238
dns-server 192.168.0.241
domain-name Management
no ip cef
no ipv6 cef
username LANAdmin1 password 0 P45l4pt151
username LANAdmin2 password 0 P45l4pt152
username LANAdmin3 password 0 P45l4pt153
username LANAdmin4 password 0 P45l4pt154
username LANAdmin5 password 0 P45l4pt155
license udi pid CISCO2911/K9 sn FTX1524ZZ7T-
ip ssh version 2
ip ssh time-out 10
ip domain-name kauko.com
spanning-tree mode pvst
class-map match-any web
match access-group name web
match protocol https
match protocol http
class-map match-any games
match access-group name csgo
match access-group name overwatch
match access-group name dota2
match access-group name fortnite
match access-group name streetfighter
match access-group name cod2
match access-group name rocketleague
match access-group name lol
class-map match-any ftp
match protocol ftp
policy-map gamelan
class games
bandwidth percent 50
class ftp
bandwidth percent 15
class web
bandwidth percent 10
class class-default
```

Router Router_1 configuration (2/3)

```
fair-queue
interface GigabitEthernet0/0
ip address 10.4.13.69 255.255.255.0
ip nat outside
service-policy output gamelan
duplex auto
speed auto
interface GigabitEthernet0/1
no ip address
service-policy output gamelan
duplex auto
speed auto
interface GigabitEthernet0/1.19
encapsulation dot1Q 19
ip address 192.168.0.142 255.255.255.240
ip nat inside
interface GigabitEthernet0/1.20
encapsulation dot1Q 20
ip address 192.168.0.158 255.255.255.240
ip nat inside
interface GigabitEthernet0/1.21
encapsulation dot1Q 21
ip address 192.168.0.174 255.255.255.240
ip nat inside
interface GigabitEthernet0/1.22
encapsulation dot1Q 22
ip address 192.168.0.190 255.255.255.240
ip nat inside
interface GigabitEthernet0/1.23
encapsulation dot1Q 23
ip address 192.168.0.206 255.255.255.240
ip nat inside
interface GigabitEthernet0/1.24
encapsulation dot1Q 24
ip address 192.168.0.222 255.255.255.240
ip nat inside
interface GigabitEthernet0/1.34
encapsulation dot1Q 34
ip address 192.168.1.14 255.255.255.252
interface GigabitEthernet0/1.35
encapsulation dot1Q 35
ip address 192.168.1.18 255.255.255.252
interface GigabitEthernet0/1.36
encapsulation dot1Q 36
ip address 192.168.1.22 255.255.255.252
interface GigabitEthernet0/2
ip address 192.168.0.246 255.255.255.248
ip nat inside
duplex auto
speed auto
interface FastEthernet0/3/0
switchport mode trunk
switchport nonegotiate
interface Vlan11
mac-address 00e0.a378.e101
ip address 192.168.0.14 255.255.255.240
ip nat inside
interface Vlan12
mac-address 00e0.a378.e102
ip address 192.168.0.30 255.255.255.240
ip nat inside
interface Vlan13
mac-address 00e0.a378.e103
ip address 192.168.0.46 255.255.255.240
ip nat inside
interface Vlan14
mac-address 00e0.a378.e104
ip address 192.168.0.62 255.255.255.240
ip nat inside
interface Vlan15
mac-address 00e0.a378.e105
ip address 192.168.0.78 255.255.255.240
ip nat inside
interface Vlan16
mac-address 00e0.a378.e106
ip address 192.168.0.94 255.255.255.240
ip nat inside
interface Vlan17
mac-address 00e0.a378.e107
ip address 192.168.0.110 255.255.255.240
ip nat inside
interface Vlan18
mac-address 00e0.a378.e109
ip address 192.168.0.126 255.255.255.240
ip nat inside
interface Vlan30
mac-address 00e0.a378.e10a
ip address 192.168.0.238 255.255.255.240
ip nat inside
interface Vlan31
mac-address 00e0.a378.e108
ip address 192.168.1.2 255.255.255.252
interface Vlan32
mac-address 00e0.a378.e10b
ip address 192.168.1.6 255.255.255.252
interface Vlan33
mac-address 00e0.a378.e10c
ip address 192.168.1.10 255.255.255.252
interface Vlan37
mac-address 00e0.a378.e10d
ip address 192.168.1.26 255.255.255.252
interface Vlan38
mac-address 00e0.a378.e10e
ip address 192.168.1.30 255.255.255.252
ip nat inside source list 100 interface GigabitEthernet0/0 overload
ip classless
ip route 0.0.0.0 0.0.0.0 10.4.13.1
ip flow-export version 9
access-list 100 permit ip 192.168.0.0 0.0.0.255 any
ip access-list extended owerwatch
permit tcp any any eq 1119
```

Router Router_1 configuration (3/3)

```
permit tcp any any eq 3724
permit tcp any any eq 6113
permit udp any any eq 5060
permit udp any any eq 5062
permit udp any any eq 5250
permit udp any any range 3478 3479
permit udp any any range 12000 64000
ip access-list extended csgo
permit tcp any any range 27015 27030
permit tcp any any range 27036 27037
permit udp any any eq 4380
permit udp any any eq 27036
permit udp any any range 27000 27031
ip access-list extended streetfighter
permit tcp any any range 27015 27030
permit tcp any any range 27036 27037
permit udp any any eq 4380
permit udp any any range 27000 27031
permit udp any any eq 27036
ip access-list extended overwatch
permit tcp any any eq 1119
permit tcp any any eq 3724
permit tcp any any eq 6113
permit udp any any eq 5060
permit udp any any eq 5062
permit udp any any eq 6250
permit udp any any range 3478 3479
permit udp any any range 12000 64000
ip access-list extended dota2
permit tcp any any range 27015 27030
permit tcp any any range 27036 27037
permit udp any any eq 4380
permit udp any any range 27000 27031
permit udp any any eq 27036
ip access-list extended fortnite
permit tcp any any eq 5222
permit tcp any any range 5795 5847
permit udp any any eq 5222
ip access-list extended cod2
permit tcp any any eq 28910
permit tcp any any range 29900 29901
permit tcp any any eq 29920
permit udp any any range 3074 3075
permit tcp any any eq 3074
ip access-list extended rocketleague
permit tcp any any range 27015 27030
permit tcp any any range 27036 27037
permit udp any any eq 4380
permit udp any any range 27000 27031
permit udp any any eq 27036
ip access-list extended lol
permit tcp any any eq 2099
permit tcp any any range 5222 5223
permit tcp any any range 8393 8400
permit udp any any range 5000 5500
permit udp any any eq 8088
ip access-list extended web
permit tcp any any eq www
permit tcp any any eq 443
permit udp any any eq 443
ip access-list extended ftp
permit tcp any any eq ftp
access-list 30 permit 192.168.0.224 0.0.0.15
line con 0
line aux 0
line vty 0 4
access-class 30 in
login local
transport input ssh
transport output ssh
line vty 5 15
no login
transport input none
transport output none
```