Relation of Various Exercises with the OSI Model

Written by Daniel Ferdetta, Breeda Gonzalez, and Ikemefuna Obinnakwelu

Professor: Dr. Illanovsky

CSIT 540 - Final Paper

December 21, 2022

## Table of Contents

[**Table of Contents**](#_buj32rbfizgw) **1**

[**Introduction :** Breeda](#_k6dqu5sb51vq) **2**

[**Exercise 1 : WebServer :** Ikemefuna](#_kvg2ixtr4i3q) **3**

[**Exercise 2 : UDPPinger :** Breeda](#_h51zgrmf6j4e) **5**

[**Exercise 3 : SMTP :** Daniel](#_cs0u26u7v5m9) **7**

[**Exercise 4 : ICMPping :** Daniel](#_z190iomap0qx) **9**

[**Exercise 5 : ProxyServer :** Breeda](#_cd686gsgbo5o) **10**

[**Exercise 6 : VideoStreaming :** Ikemefuna](#_4jmchvf1gw1) **12**

[**Exercise 7 : Traceroute :** Daniel](#_xequiih57vi2) **14**

[**Conclusion :** Ikemefuna](#_r4yrmol3jts4) **16**

[**Bibliography**](#_di6ueqg4h8tx) **17**

## **Introduction :** Breeda

This paper is a comprehensive review of the creation and lessons learned from the Computer Networks final project. The details on the networking layer that is in scope for each coding assignment is also included. The paper includes background on the seven exercises completed, impacted OSI layer, details on writing the code, explanations on coding decisions, and lessons learned after completion of the sections. It is important to note that the coding project instructions, including skeletons for how to write the code, were in Python. This group has written out our code in Java instead which means some explanations included are details on how the Python skeletons were translated into Java.

The exercises walked the group through the essentials of Computer Networking. There was a focus on the OSI layer which was being utilized by the respective coding assignment. As an introduction, the “Open Systems Interconnection (OSI) Reference Model is a conceptual framework that describes functions of the networking or telecommunication system independently from the underlying technology infrastructure. It divides data communication into seven abstraction layers and standardizes protocols into appropriate groups of networking functionality to ensure interoperability within the communication system regardless of the technology type, vendor, and model.” (Raza, 2018)

The seven layers, in order, are Physical, Data Link, Network, Transport, Session, Presentation, and Application. The physical layer is responsible for the physical equipment and hardware utilized in computer networks. This would include cables, hardware, and protocols such as Wifi or ethernet (Raza, 2018). Data Link is responsible for data transmission between nodes which would include the sublayers that help divide the data transmission. The Networking layer is responsible for data routing through the best path while considering different network characteristics such as traffic (Raza, 2018). The fourth layer is the Transport layer. This layer is responsible for “complete and reliable delivery of packets” (Raza, 2018).

The fifth layer is the Session layer- this layer is responsible for data exchange within the same network such as authentication requests and opening, closing, or authorizing sessions (Raza, 2018). The sixth layer is Presentation which marries the syntax with the semantics of the message; in other words, this layer is responsible for encryption and decryption among other activities (GeeksforGeeks, 2022). The final layer, the seventh layer, is the Application layer. This layer is responsible for the software level networking. This includes user functions such as email, file transfers, and directory services (Raza, 2018).

## **Exercise 1 : WebServer :** Ikemefuna

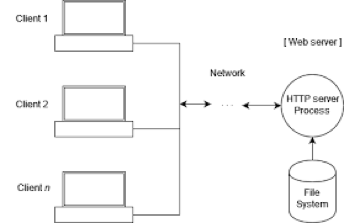
When it comes to the project, Exercise 1 focuses on the WebServer. A web server is software and hardware that uses HTTP (Hypertext Transfer Protocol) and other protocols to respond to client requests made over the World Wide Web. The main job of a web server is to display website content through storing, processing and delivering web pages to users. Besides HTTP, web servers also support SMTP (Simple Mail Transfer Protocol) and FTP (File Transfer Protocol), used for email, file transfer and storage. Web server hardware is connected to the internet and allows data to be exchanged with other connected devices, while web server software controls how a user accesses hosted files. The web server process is an example of the client/server model. All computers that host websites must have web server software. Web servers are used in web hosting, or the hosting of data for websites and web-based applications or web applications.

Web server software is accessed through the domain names of websites and ensures the delivery of the site's content to the requesting user. The software side also comprised several components, with at least an HTTP server. The HTTP server is able to understand HTTP and URLs. As hardware, a web server is a computer that stores web server software and other files related to a website, such as HTML documents, images and JavaScript files.

When a web browser, like Google Chrome or Firefox, needs a file that's hosted on a web server, the browser will request the file by HTTP. When the request is received by the web server, the HTTP server will accept the request, find the content and send it back to the browser through HTTP.

More specifically, when a browser requests a page from a web server, the process will follow a series of steps. First, a person will specify a URL in a web browser's address bar. The web browser will then obtain the IP address of the domain name -- either translating the URL through DNS (Domain Name System) or by searching in its cache. This will bring the browser to a web server. The browser will then request the specific file from the web server by an HTTP request. The web server will respond, sending the browser the requested page, again, through HTTP. If the requested page does not exist or if something goes wrong, the web server will respond with an error message. The browser will then be able to display the webpage.

Considerations in choosing a web server include how well it works with the operating system and other servers; its ability to handle server-side programming; security characteristics; and the publishing, search engine and site-building tools that come with it. Web servers may also have different configurations and set default values. To create high performance, a web server, high throughput and low latency will help.



*Figure 1.1*

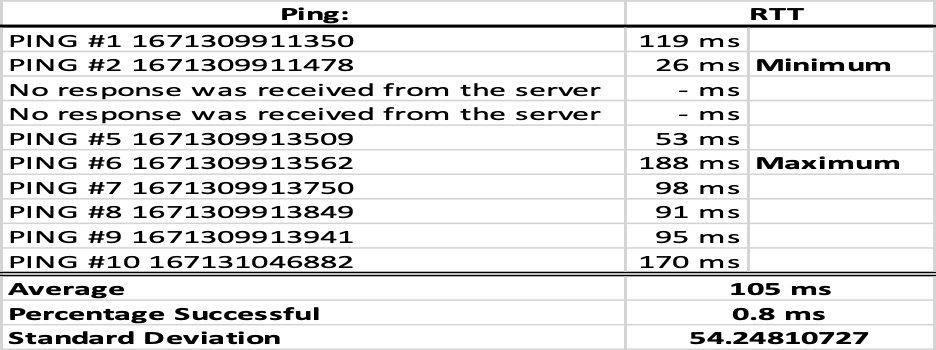
## **Exercise 2 : UDPPinger :** Breeda

This exercise was about the basics of socket programming. This required being able to send and receive datagrams using UDP sockets. Part of creating this process included how to set up a proper socket timeout. This exercise assisted in gaining a familiarity with ping applications and the usefulness in computing statistics such as packet loss rate. It also demonstrated the network layer (layer 3) of the OSI model.

As background for this exercise, ping is “packet internet or internetwork groper”. (Zola, 2021) The purpose of this is to allow a user to test and validate if a particular IP address exists. Ping also checks if the IP Address is accepting requests in computer network administration (Zola, 2021), i.e. check that the IP address is up and running and capable of accepting requests. If the server is shut down or if the server is handling other requests, it may not be capable of receiving new requests at this time. The technical way this is completed is through ICMP – the UDP datagrams or packets are sent by a host that wants to perform this check. Once the packet is sent, the UDP socket listens for all ICMP incoming messages. If a message is received, the socket exits with the information about the receiver. If there is no response within a set amount of time (ie the timeout function), then it exits as a failure (Falko, 2015). This exercise included prewritten code to function as the server which included a simulated fail-function built in to replicate the true to life scenario of packet failure during pings. To simulate the packet loss that would occur in a live example, the provided code by Kurose and Ross simulated a 30% packet fail rate.

Exercise 2 impacted the third layer (networking) layer. “ICMP (Internet Control Message Protocol) is located at the Network layer of the OSI model (or just above it in the Internet layer, as some argue), and is an integral part of the Internet Protocol suite (commonly referred to as TCP/IP)” (Parker, 2022). The networking layer is responsible for packets, which includes IP and ICMP as addresses for sending and receiving the packets. This exercise specifically created packets of data and sent them out. The already created server sent them back following ICMP protocol. This aligns with the 3rd networking layer. The networking layer is responsible for determining the best possible path which includes factors such as congestion, traffic controls, priority of service, and other network characteristics (Raza, 2018). The 3rd layer is also responsible for packaging datagrams, or logical addressing. Part of the function of the code design was to include relevant information in the datagram creation so that it could successfully be sent out.

The provided code offered an excellent launching point. The server code was written in python, while the expected client code was being written in Java. As a first step, Kurose and Ross assignment was searched online and a version with the server code in Java was identified (PingLab). This was utilized to build off of the client code. First, the client code needed to establish the basic parameters that were identified in both the server code and the exercise requirements. The number of pings to be sent to the server from the client was 10. The token timestamp and the max time out time were also set. The max timeout time was set because if no message was received back from the server, the socket should stop listening and try again. If this wasn’t set, the socket would continually listen and receive no response.



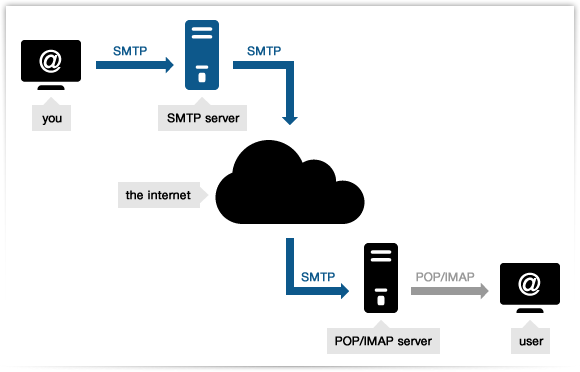
*Figure 2.1*

As an optional exercise to this assignment, there was an ask to calculate the minimum, maximum, and average RTT, along with the packet loss rate in a percentage. As shown in Figure 2.1, the packet loss rate was above the expected simulation of only 70% success rate. The spread between the RTT for each ping was 26ms to 188 ms. The standard deviation showed a decently large spread for the small sample size of data- the standard deviation was about 54.

## **Exercise 3 : SMTP :** Daniel

Exercise 3 is about SMTP (Simple Mail Transfer Protocol) and is a networking protocol used to send and receive electronic mail (email) over the Internet. It operates at the Application layer of the OSI model, which is the top layer of the seven-layer networking model used to describe how different networking protocols interact with each other. SMTP is responsible for the transfer of emails from one machine to another over a network, and it is used by email clients such as Microsoft Outlook and Mozilla Thunderbird to send and receive emails. When an email is sent, the email client communicates with an SMTP server using a series of commands and responses to transfer the email to the recipient's mail server.

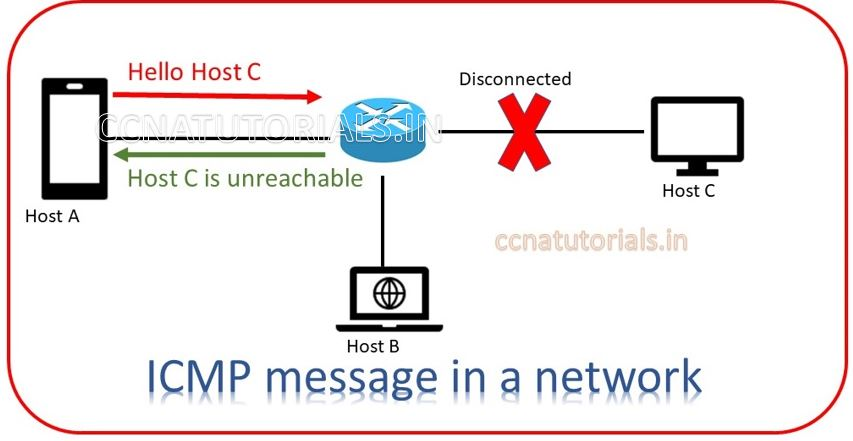
The SMTP protocol works by establishing a connection between the sender's and the recipient's mail servers, and then sending commands and responses back and forth to transfer the email. These commands and responses are text-based and are transmitted using TCP (Transmission Control Protocol) at the transport layer of the OSI model as pictured in Figure 3.1.

  
*Figure 3.1*

In Exercise 3, the user creates a socket and connects to a mail server and port. Following this, several commands are sent in search of replies. One noteworthy piece of information regarding SMTP is that commands are case-sensitive and require a precise syntax in order to work as intended.

## **Exercise 4 : ICMPping :** Daniel

Requirement number 4 touches on Internet Control Message Protocol (ICMP) and implementation of a ping application through the usage of ICMP requests and reply messages. In the seven-layer OSI model, this falls under level three which is the network layer. ICMP is most known for its usage through the “ping” command. The ping command sends ICMP requests to a desired host address, and is met with an echo reply. This reply indicates if the host is reachable and responding on the network. In addition, ICMP can provide information regarding a number of problems when discussing packet delivery. According to IBM, ICMP can inform hosts about maximum transmission unit limitations, packet expiry, destination reachability, and any routing problems.

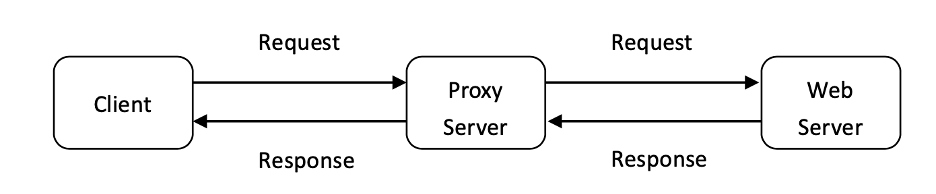
Through the previous wireshark assignments we had in the course, that ping command was used on multiple occasions, and some files were too large to be fit in a single packet. Due to this, ICMP can detect this and send an ICMP fragmentation needed packet. Fragmentation describes when one larger packet is broken up into two or more smaller pieces by the router, then reassembled after it is received. Packet expiry simply refers to a packet’s “Time-To-Live” field reaching zero. Destinations being unreachable are also detected, and an example of what that means can be pictured in Figure 4.1. According to 4.1, the source host A is able to reach the router of host B, but is unable to establish with host C who is seemingly disconnected or otherwise unreachable. The final feature of ICMP pings is searching for optimal routing paths, though this is often disabled as there are other protocols more tailor-made for this function.  
  
*Figure 4.1: ICMP message unable to connect to Host C*

Exercise 4 uses the ping command to determine reachability. The program pings a host and attempts to make a connection with that host. If it does not succeed it continues to try again. After enough attempts, the program moves on to the next IP and repeats the process. There was difficulty in maintaining a connection due to Motnclair’s internet interference, despite attempting through a Virtual Private Network (VPN), personal wireless connection, and a wired connection.

## **Exercise 5 : ProxyServer :** Breeda

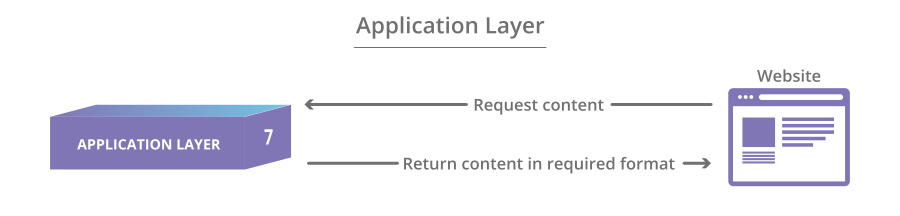
This exercise was to develop an understanding of how web proxy servers work as well as understanding of cache. This exercise involved creating a small web proxy server which only accepted GET-requests and caching the web pages. This exercise was not exclusive to HTML pages but included images as well. It is important to note that this assignment and skeleton code was provided in Python. The final server code was instead written in Java as that is the only language this student was familiar with. This assignment represented the seventh layer, the Application layer of the OSI model.

As a lesson learned from this exercise, I would say I should have learned python so I could start with the skeleton. I jest- but on a more serious note, I would say in future I would have spent more time ensuring that that I fully understand everything that was being supplied in the provided code before starting as my method of working on writing code as I read through the skeleton was not very effective and provided a lot of debugging opportunities. I did not manage to get the code running or ensure it was totally fulfilling the requests of the assignment because I did not know how to set up the proxy link on my internet browser. As such, I do not have screenshots showing completion. Part of this issue may come from my translations to Java- perhaps I was correctly setting up the proxy settings in my browser but did not have the coding correct.

**

*Figure 5. 1 Explanation of proxy server function in communication*

This exercise is part of the seventh layer of the OSI model- the application layer. The exercise involves creating a proxy server- this means that the request from the client goes first to the proxy server which then forwards the requests onwards. The server then responds to the initial request, and sends back a response. The response is received by the proxy server, which then forwards the request to the client which initially requested the information. This code specifically includes a function to cache the web pages that way on a second request. In this scenario, the client requests a web page, and the request is received by the proxy server. Because the proxy server has cached the web page, the proxy server responds directly to the client. Figure 5.1, from the assignment, visualizes this process. All of this is done at the application layer, from application to application. Figure 5.2 shows how exactly Exercise 5 relates to the application layer work.



*Figure 5. 2 Visual depiction of the application layer*

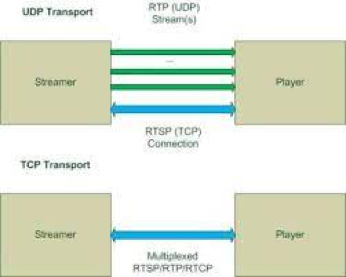
## **Exercise 6 : VideoStreaming :** Ikemefuna

The Internet Engineering Task Force (IETF) standardized the Real-Time Transport Protocol as a network-level protocol that transmits video and audio data when timeliness is essential. Your video platform would be better off using RTP. RTP streaming timestamps packets so they arrive in the correct order even if some are missing, which ensures whatever is there arrives promptly. The stream flows constantly this way, facilitating your conversation’s momentum instead of making you endure frustrating playback stalls and glitches. Because RTP streaming prioritizes speed instead of accuracy, it’s best applied to use cases like telecommunications to achieve as close to real-time transmission as possible, or for streaming to sizable audiences. Voice over IP (VoIP) applications frequently leverage RTP, as do major communications platforms like FaceTime, WhatsApp, Zoom, and Microsoft Teams (at least, their own unique versions of RTP). RTP serves as the foundational transmission method while each company adds its own playback features on a different protocol layer.

RTP isn’t strictly for audio-visual data, either. It’s applicable to many kinds of data streaming, including status updates, control information transport, active badges, and more. It’s also important to note what RTP streaming is not for. RTP isn’t designed to allocate computational resources, nor does it offer quality-of-service (QoS) features. Though labeled a “real-time” protocol, no existing end-to-end protocol can deliver data in true real time — it still depends on lower-layer protocols that govern switch and router resources. Essentially, RTP is fit for delivering content attached to data packets that bring it closer to real time, such as Timestamps, controls for synchronizing multiple streams, Sequence numbers (for detecting lost packets), payload identification, beginning and end of frame indications, and intra-media synchronization.

The transport layer protocol RTP streaming works best with is UDP. Because UDP doesn’t require recipient devices to let the sender know all data packets have arrived, it uses checksums instead of handshakes to validate data integrity; it's convenient for swift transmission instead of complete delivery. As such, RTP and UDP make an excellent pair; they both prioritize speed over accuracy, reducing latency at the expense of reliability. However, whatever packets are missing are likely unnoticeable to the human eye or ear. RTP and RTSP are often used interchangeably, so it’s understandable if you confuse them. They work closely together, but there is a distinct difference: RTP is the transport protocol that moves data from one endpoint to another, but RTSP Real-Time Streaming Protocol is the presentation-layer protocol that enables users to control the media server to play and pause video.

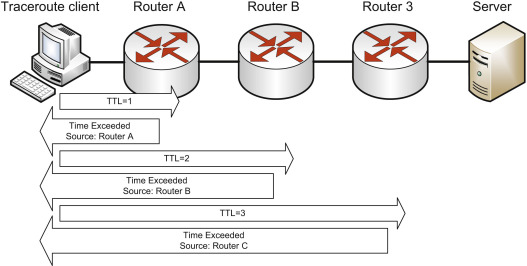
RTP also does not automatically provide encryption, which is where Secure Real-Time Transport Protocol bolsters (SRTP) comes in. SRTP bolsters media stream security, reducing the likelihood of successful cyber-attacks.



*Figure 6.1*

## **Exercise 7 : Traceroute :** Daniel

The final exercise, exercise 7, focuses on tracerouting. Tracerouting is a network diagnostic tool used to trace the path that a packet takes from the source to the destination. It is useful for identifying routing issues and understanding the topology of a network. In the OSI model, tracerouting also takes place at the Network layer (layer 3). At this layer, routing decisions are made based on the destination IP address of the packet.

When a packet is sent from the source to the destination, it is passed down through the layers of the OSI model. At the Network layer, the packet is examined to determine the next hop in the path to the destination. This is done by consulting the routing table, which contains a list of networks and their corresponding next hop addresses. If the destination IP address is not in the same network as the source, the packet is forwarded to the next hop. This process continues until the packet reaches its destination or an error occurs as pictured in Figure 7.1, where the destination was reached.

*Figure 7.1*

Tracerouting works by sending a series of packets with gradually increasing Time To Live (TTL) values. The TTL value is a field in the IP header that determines how many hops a packet can take before it is discarded.When a packet with a TTL of 1 is sent, it is only allowed to travel one hop before it is discarded. The first router in the path will send an error message back to the source, indicating that the packet has expired. The traceroute tool can then record the IP address of the router that sent the error message.

This process is repeated with gradually increasing TTL values until the destination is reached or the maximum number of hops is reached. Along the way, the traceroute tool can record the IP addresses and round-trip times of each router in the path. By analyzing the path and round-trip times, network administrators can identify bottlenecks and potential issues with the routing infrastructure. Tracerouting is an essential tool for troubleshooting and maintaining the performance of networks.

Exercise 7 attempts through a number of sources to build a packet, enter a loop with TTL at 1 up to the maximum number of hops, and sends a socket. Following this, the program then attempts to fetch a type. If no type can be found, then the program skips to the next socket and attempts the process again on the new source.

## **Conclusion :** Ikemefuna

In conclusion, computer networking refers to interconnected computing devices that can exchange data and share resources with each other. These networked devices use a system of rules, called communications protocols, to transmit information over physical or wireless technologies. Nodes and links are the basic building blocks in computer networking. A network node may be Data Communication Equipment (DCE) such as a modem, hub or switch, or Data Terminal Equipment (DTE) such as two or more computers and printers. A link refers to the transmission media connecting two nodes. Links may be physical, like cable wires or optical fibers, or free space used by wireless networks.

In a working computer network, nodes follow a set of rules or protocols that define how to send and receive electronic data via the links. The computer network architecture defines the design of these physical and logical components. It provides the specifications for the network’s physical components, functional organization, protocols, and procedures. Many computer networks are software-defined. Traffic can be routed and controlled centrally using a digital interface. These computer networks support virtual traffic management. All networking solutions come with in-built security features like encryption and access control. Third-party solutions like antivirus software, firewalls, and antimalware can be integrated to make the network more secure.

## **Bibliography**

Zola, Andrew. “What Is a Ping and How Does It Work?” *Networking*, TechTarget, 16 July 2021,

https://www.techtarget.com/searchnetworking/definition/ping.

Falko. “A Close Look at UDP Ping.” Cybrary, Cybrary, 1 Sept. 2015, https://www.cybrary.it/blog/0p3n/a-deep-look-into-udp-ping/.

“Programming Assignment: UDP Ping Lab.” *Ping Lab*, https://sites.cs.ucsb.edu/~ebelding/courses/176A/f05/prog\_assign2.html.

Parker, Jeff. “What Is ICMP? the Protocol, Port Number and Ping!” *PC & Network Downloads -* PCWDLD*.com*, 16 Feb. 2022, https://www.pcwdld.com/what-is-icmp-and-port.

Raza, Muhammad. “OSI Model: The 7 Layers of Network Architecture.” *BMC Blogs*, 29 June 2018, https://www.bmc.com/blogs/osi-model-7-layers/.

“Presentation Layer in OSI Model.” *GeeksforGeeks*, 2 Aug. 2022, https://www.geeksforgeeks.org/presentation-layer-in-osi-model/.

“Internet control message protocol (ICMP) and other layer 3 protocols,” *IBM*. [Online].

https://www.ibm.com/docs/en/zos-basic-skills?topic=nll-internet-control-message-protocol-icmp-other-layer-protocols.

“What Is SMTP.” *Smtp Mail Server - Professional SMTP Service Provider*,   
 https://serversmtp.com/what-is-smtp/.