

Task1FinalReportRJ

by Robert Jamieson

Submission date: 02-May-2023 07:16PM (UTC+0100)

Submission ID: 204381825

File name: 24101_Robert_Jamieson_Task1FinalReportRJ_499749_1000287443.docx (491.47K)

Word count: 4122

Character count: 24546

Robert Iwan Jamieson

The University of Wales Trinity Saint David
School of Applied Computing
Wales Institute of Science and Art

**Socket Programming and Concurrency combined
assessment – Task 1**

Socket Programming and Concurrency (ACCA6028)

Robert Iwan Jamieson (2004728)

Date of submission: 02.05.2023

Contents

List of figures	3
Chapter 1 Introduction	4
Chapter 2 – How the Transmission Control Protocol operates	5
2.1 Three way handshake	6
2.2 The Two Generals' Problem	7
Chapter 3 – The impact the Maximum Transmission Unit has on the transfer of data....	8
Chapter 4 - Synchronicity and asynchronicity in network protocols.....	11
Chapter 5 – Interoperability for networked applications	13
Chapter 6 Conclusion	15
Chapter 7 References	16

List of figures

Figure 1: Illustration of the three way handshake. Inspired by source: [14]. 6

Figure 2: OSI Model vs TCP/IP Model. Inspired by source: [17].....8

Figure 3: OSI model revealing what PDUs are referred to in each layer [24]. Inspired by sources: [12] [24]. 10

Figure 4: Illustration of how synchronicity and asynchronicity work. Source: [28].....11

Figure 5: TCP/IP model revealing protocols found within TCP/IP. Inspired by sources: [12], [38], [39] and [40].....13

Chapter 1 Introduction

This report first explores the topic of how the Transmission Control Protocol (TCP) [1] operates and helps to ensure the delivery of data. The three-way handshake [2] and Two Generals' Problem [3] are therefore discussed. The impact that the Maximum Transmission Unit (MTU) [4] has on the size of data that can be transmitted and related factors to it are discussed. Furthermore, their relative positions in the OSI model [5] will be mentioned.

After this, the topic of Synchronicity and asynchronicity [6] in network protocols is discussed, including how the I/O is affected. Examples of applications where each are suitable are described. Finally, the meaning of interoperability [7] in networked applications is explored, including how it is best achieved.

Throughout the report, special attention will be paid to how each of the topics impact on the design decisions programmers face when developing networked applications.

Chapter 2 – How the Transmission Control Protocol operates

The Transmission Control Protocol (TCP) is a communication standard and has widespread use when it comes to transferring data over a network [1]. It ensures that networked devices can send and receive data to one another [1], even over large geographical distances. It therefore allows for services, such as emails, browsing the web, messaging etc.

In order for all devices to be able to understand and thus communicate with each other, the TCP layers must be followed [8]. TCP is often accompanied by the Internet Protocol (IP) and together they are the TCP/IP [8]. The TCP/IP layers from top to bottom include the:

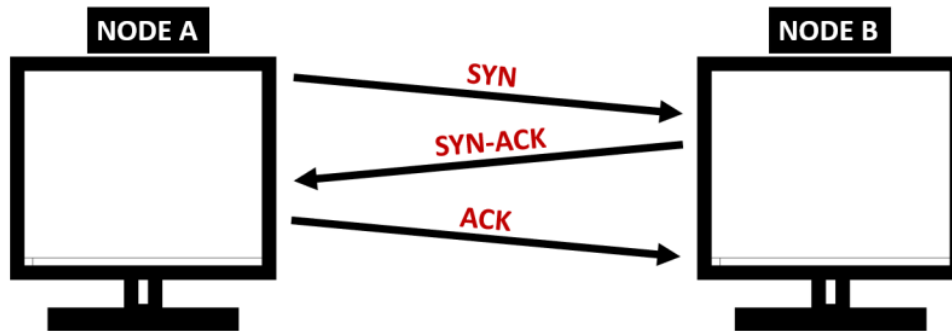
- **Application layer** – The application layer is the layer which the end user can interact with. It is here that data is displayed in a way that is readable to humans [9].
- **Transport layer** – The transport layer helps ensure that data is transported reliably and efficiently [9].
- **Internet layer** – The internet layer routes the data to its intended destination using the most efficient pathing [10]. Upon reaching the destination, it reorganizes the data back into its original form.
- **Network Interface layer** – The network interface layer determines how the data is physically transferred from one node to another [5].

The alternative to the TCP is the User Datagram Protocol (UDP) [11]. The UDP is used where a faster, more efficient connection is required, such as in multiplayer games or when streaming videos [11]. Its downside however is that whenever data is lost or not sent correctly in its travel time, it is not corrected [11]. Developers may make the design decision of using UDP in a multiplayer game instead of TCP because a faster connection is what is prioritized, rather than a more reliable one [12]. If a packet of data is corrupted, by the time it is re-transmitted and delivered, it would no longer be of use and may only serve to cause confusion or to annoy the player.

The TCP is a much more reliable protocol [11]. Although it is slower, it ensures that whenever data is lost or not sent correctly, it is re-transmitted until it is received as it was originally sent [11]. This gives developers of networked applications such as texting applications a reason to use TCP over UDP because users of the app will be able to have confidence that their messages are always being delivered.

2.1 Three way handshake

The three way handshake is TCP's way of ensuring that nodes have a more reliable method of communicating with each other over a network [2]. Before a node can transmit the data that they wish to be transmitted, first a connection must be established between both nodes



[13].

Figure 1: Illustration of the three way handshake. Inspired by source: [14].

As illustrated in Figure 1, imagine that Node A seeks to transmit data to Node B. The steps of doing so reliably [13] with the three way handshake is described below:

1. **Synchronize (SYN)** [14] – A SYN segment is sent from Node A to request a connection be made between it and Node B [14]. This request seeks for both nodes to use a synchronized sequence number because that is what is required for a connection to be established [14].
2. **Synchronize and Acknowledge (SYN-ACK)** [2] – After receiving the SYN segment from Node A, Node B replies with a SYN-ACK to Node A [2]. This reply details Node B's SYN segment so that both nodes have the necessary details to establish a connection and begin communicating with each other [2]. In this reply, Node B also sends an acknowledgement to Node A that confirms that Node A's SYN request was received [2].
3. **Acknowledge (ACK)** [14] – After receiving Node B's acknowledgement, an acknowledgement is sent by Node A to confirm that it has acknowledged the response issued by Node B [14].

The TCP makes use of what is called a Positive Acknowledgement Re-transmission (PAR) method [2]. It expects a PAR response from the node of which the request was sent to [2]. If for example Node B does not respond with a PAR to Node A's SYN segment, it is presumed

Robert Iwan Jamieson

that the request never reached Node B. Node A would therefore re-transmit the request again, seeking a response [2].

By using this process, the reliability of the connection is greatly improved [13] because both sides can be confident in most cases that their data is being retrieved. If a node did not receive a confirmation that the other node received their reply – they would have no way of knowing for certain that they did [2].

2.2 The Two Generals' Problem

Notably, although the three way handshake process makes the connections more reliable [13], it is still affected by the Two Generals' Problem [3]. This is because there can never be enough acknowledgements for nodes to be 100% certain that the receiving node received its transmission [3].

In the three way handshake method, as illustrated in Figure 1, Node B never acknowledges Node A's acknowledgement. Thus, Node A can't be sure whether or not Node B received its acknowledgement. However, if Node B did acknowledge Node A's acknowledgement, the issue would still remain because then that acknowledgement would need to be acknowledged. The problem of trying to acknowledge every acknowledgement is that it isn't possible and would create an infinite loop of replies [3].

Therefore, the three way handshake is limited in how reliable it can make a connection – it can only offer reliability to a certain extent. The Two Generals' Problem is thought of as “unsolvable” [3]. One proposed solution is to send multiple requests at once in hope that at least one of the requests is received [15]. This could give the sender more confidence, however it would still not confirm for a certainty that the receiver would receive the request and would be at the expense of efficiency and costs [15].

Chapter 3 – The impact the Maximum Transmission Unit has on the transfer of data

The Open Systems Interconnection (OSI) [16], in theory, serves the same purpose as the TCP/IP but is designed somewhat differently [5], as seen in Figure 2:

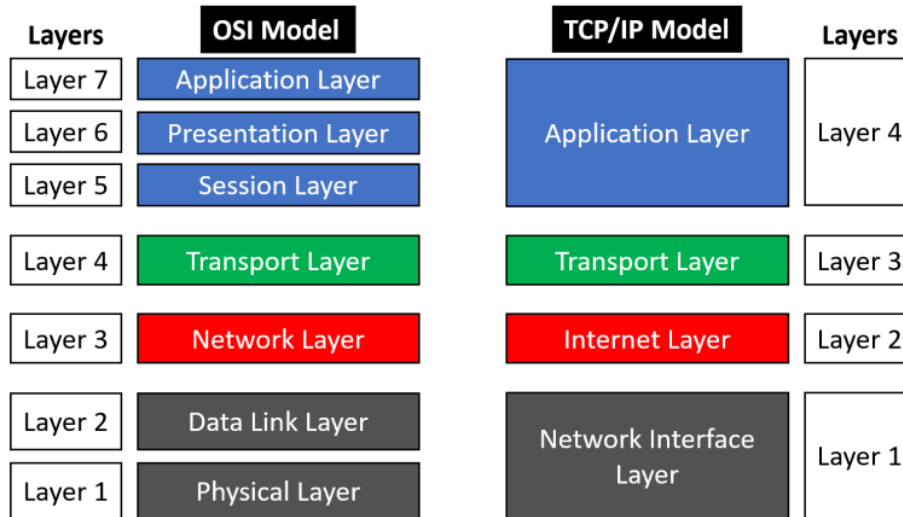


Figure 2: OSI Model vs TCP/IP Model. Inspired by source: [17].

The TCP/IP is a simpler, yet functional architecture unlike the OSI architecture. The OSI model is “conceptual” [16] and thus is not implemented in practise, but allows for a better understanding of how data is transmitted over a network [16]. Throughout this chapter, when referring to layers it is always the OSI model that is being referred to.

When data is transmitted, it can be transmitted to another node in the Local Area Network (LAN) or transmitted to a Wide Area Network (WAN) [18]. In the case of transmitting data from one node to another in a LAN, a frame, such as an ethernet frame is used [18]. Other frames, including the newer and faster Jumbo Ethernet Frames [4] can be used but standard **ethernet frames will be referred to in this chapter** [4]. Frames encapsulate data and transmit it to a designated Media Access Controller (MAC) address [19]. Frames are used at OSI’s level 2 data link layer [19].

Frames by themselves do not have the capability of transmitting data to nodes outside of a LAN [18]. Frames therefore rely on packets to communicate with WANs via IP addresses and therefore encapsulate them [18]. Packets are used at OSI’s level 3, network layer [19]. As a frame encapsulates a packet, it limits the maximum transmission rate of the packet [4]. The

maximum amount of data that a packet can transmit is known as the Maximum Transmission Unit (MTU) [4] and is also present in OSI's layer 3, the network layer [4].

Ethernet frames have a maximum size of 1518 bytes, when the header at 18 bytes is included [4]. This means that a packet's maximum transmission rate is 1500 bytes of data [4]. The impact that MTU has on the transfer of data depends. For example, if all nodes in the data transmissions path accept an MTU size of 1500 or more, then the data transmission will be faster and more efficient than if it were to be originally broken down into even smaller segments [20]. This is because these segments would have to be further re-organized when reaching the destined node and thus would affect the transmission speed [4].

However, as suggested the MTU for nodes (such as a router) that a packet travels through throughout a network can differ [4]. They may only allow a smaller MTU than what the sending node is transmitting [4]. Unfortunately, the MTU of other nodes cannot be foreseen [4]. Therefore, in order for the data to get through the bottleneck, it has to be segmented - a process known as fragmentation [21]. Segmentation of packets occurs in OSI's level 4, transport layer.

Although IP fragmentation allows for the successful transmission of data [4], one downside is that "each fragment has duplicated IP headers" [4]. Due to more headers, this means more bytes are transmitted [4]. IP fragmentation also causes further inefficiencies as it means that there is more processing to further segment and re-organize the packets of data [4].

IP fragmentation also has the issues of:

- Causing all fragments to be re-transmitted if just one fragment is corrupted [21].
- Causing fragmented data to have complications with security systems, including firewalls [21].

Due to IP fragmentation having many drawbacks, it should be "avoided where possible" [21]. To help avoid it, developers should make sure to configure all of their network devices to ensure that they all support the same MTU size. By not doing so, before the data is even transmitted out of the local network to a client it could already become fragmented. It is therefore vital that this is done for efficiency purposes.

Furthermore, although some companies may enjoy the use of the faster Jumbo Ethernet frames which can support a MTU of 9000 bytes per packet [4], it is “rare” [4] for networks to use it. The MTU of 1500 bytes “is used almost universally on access networks” [22]. To avoid unnecessary fragmentation, developers should not attempt to send a MTU size of greater than 1500 bytes outside of their own network. If a packet size of over 1500 bytes is transmitted to an external network, it will likely need to be fragmented – which would not be ideal, for the reasons mentioned.

As seen in Figure 3, the Protocol Data Units (PDUs) are present in all OSI layers but are referred to differently depending on the layer [23]. PDUs, including packets, frames and segments [24] have already been described above.

As data is transmitted throughout the layers, each PDU relays structured data to the next layer [24]. They have the same goal – which is to ensure that data is able to be transmitted from one node to another successfully [24].

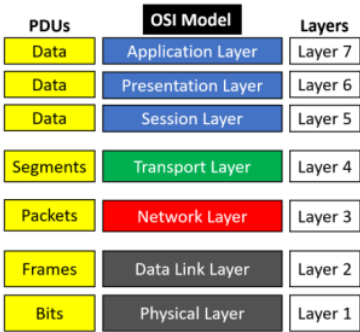


Figure 3: OSI model revealing what PDUs are referred to in each layer [24]. Inspired by sources: [12] [24].

Chapter 4 - Synchronicity and asynchronicity in network protocols

Synchronous and asynchronous are differing communication methods [6]. Whether an application makes use of synchronicity or asynchronicity depends on the applications function [6].

Synchronous communication allows for multiple parties of a networked application to receive real-time responses from the other party [6]. Synchronous communications **typically** use a “blocking architecture” [25] to deal with Input/Output (I/O) operations [26]. This means that whenever data is transmitted, the operation is actively awaiting and expecting an immediate response from the receiver [25].

Once the response is given, the thread is free to execute the next instruction [25]. Although, as a thread becomes idle, this can result in poor performance [27] and may make the application seem noticeably unresponsive [28] - particularly if it is a large operation. Despite of this, synchronous communication is simpler in its concept [6].

An example of synchronous communication is this: when typing a query in Google it can suggest multiple queries it predicts you want to search for before you even finish writing the full query. Google is therefore transmitting data to the user in real-time – it is responding immediately to the user as they are typing their query.

Other applications that benefit from synchronicity include:

- Online calls.
- Online chat rooms [29].

In contrast, although asynchronous communication still allows for both parties to communicate with each other, there is no expectation for the receiving party to immediately respond [6]. This is because asynchronous communications **typically** use a “non-blocking architecture” [25]. This means that I/O operations can run concurrently [26] and are each responded to whenever it is convenient to do so [25], as illustrated in Figure 4.

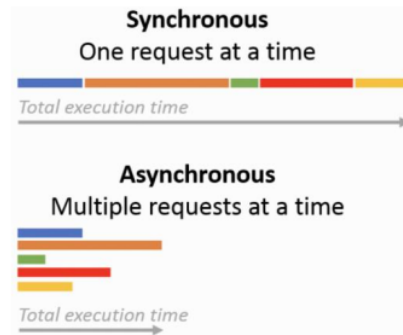


Figure 4: Illustration of how synchronicity and asynchronicity work. Source: [28].

Robert Iwan Jamieson

A single thread can be used for multiple operations whilst still not receiving a reply from its first operation [6] because it is not idly waiting for a response from any of its operations.

Due to this, asynchronicity is more robust and scalable [6] and can deal with large numbers of I/O operations better than synchronicity can – it is more efficient [28]. However, a big downside with asynchronous communication is that resolving bugs can be troublesome and the application can become very complex [30].

Signing up for an online give-away is an example of asynchronous communication. A hopeful customer transmits their details to a server and gets no immediate response. The server receives the data but does not necessarily immediately act upon it or reply to the user. It could be weeks until the give-away is concluded and only then the server may respond.

Other applications that benefit from asynchronicity include:

- E-mails [31].
- Online forums [29].

Asynchronous and synchronous communication can be used for the same task, such as requesting data from a database [28] [27]. With synchronous communication, an immediate reply is requested [25]. Thus, the operation must be completed before the thread can be used for another operation [25]. However, if asynchronous communication is used, the data could be downloaded in the background whilst the user is free to perform other operations [28].

Developers should consider the use of synchronous and asynchronous communication methods in networked applications carefully. In supported programming languages, developers can use both synchronous and asynchronous operations where necessary [25]. The best one to use is heavily dependent on the application's basis. Factors such as the applications purpose; how simple [6] the developer wants the application to be; whether an immediate response is necessary [6] and the efficiency [28] of the program are all key considerations.

Chapter 5 – Interoperability for networked applications

In this chapter the meaning of interoperability for network applications and how it is best achieved will be discussed.

When interoperability is achieved, it allows for differing networked devices and applications to communicate to one another seamlessly [7]. Interoperability is used a lot for networked applications, as well as notably when browsing the web [32]. When browsing from one website to another, the thought of the underlying practicalities of this may not occur to the end user due to how seamless it is. The same website can be viewed via different browsers such as Firefox and Edge and can be viewed on differing operating systems such as Windows, Apple, Android and Linux. This can all be contributed to the standardization of key protocols, such as the Hypertext Transfer Protocol (HTTP) [32].

When the web was still in its infancy, Microsoft and ZDNet AnchorDesk were both competing to create their own modified version of HTML [33].

Fortunately, it was eventually agreed that the two companies would cooperate to ensure that the internet was based “on a single standard” [33]. If this had not been the case, it is conceivable that many other companies would have developed their own version of HTML too. This could have led to the web being fractured and largely incompatible, with web developers having to choose which version of HTML they would support.

HTML is a markup language that was therefore standardized [34] – however the reasoning behind as to why it was standardized can also be applied to network protocols, such as the TCP/IP. The TCP/IP is a standard [35] that incorporates many other communication protocols throughout its layers – it “allows different protocols to interoperate” [36] and thus enables simpler communications. There are many protocols within the TCP/IP, as seen in (but not limited to) Figure 5.

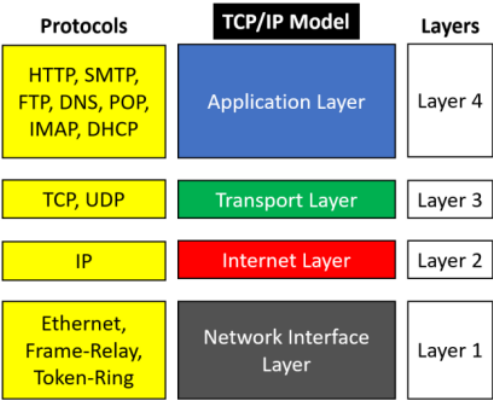


Figure 5: TCP/IP model revealing protocols found within TCP/IP. Inspired by sources: [12], [38], [39] and [40].

Robert Iwan Jamieson

Networked applications that do not comply with protocol standards lack the interoperability for it and other devices and applications to communicate to one another without compatibility issues [37].

Interoperability for networked applications is therefore best achieved by companies and organizations cooperating with each other to agree to standards [37]. It is organizations such as the Internet Engineering Task Force [32] that help ensure the continuity of interoperability, by maintaining protocols such as the TCP/IP [32].

When developing a networked application, developers should therefore use established protocols that are freely available for use. This is because it ensures that the application will be able to communicate with end users and vice versa without issues. Furthermore, the application would be easily scalable in the future if it needs to start communicating with third party applications, for example.

Chapter 6 Conclusion

This report set out to explore multiple topics. The TCP/IP is reliable because it re-transmits data if the data is not arrived as it was originally sent [11]. Furthermore, although the three-way handshake does an excellent job at ensuring data is transmitted successfully [2], it cannot be guaranteed without a doubt due to the Two Generals' Problem [3].

It was found that the MTU does have a major impact on the size of the data being transmitted – it is generally limited to 1500 bytes of data per packet [4]. Data being transmitted with a higher MTU than the receiving node will have to go through a process known as fragmentation [21]. IP fragmentation has many downsides and thus it is best to avoid it as much as possible [21].

Synchronicity and asynchronicity both differ. Whilst synchronous communication aims to communicate in real-time with another node, asynchronous communication doesn't [6]. The best to use in a networked application heavily depends on the applications purpose.

Finally, interoperability for networked applications refers to how seamlessly an application can communicate back and forth with clients, other nodes and applications [7]. To do so, an established protocol should be used so that the application can easily communicate with others that make use of that same protocol [37].

Chapter 7 References

- [1] Fortinet, “What is Transmission Control Protocol TCP/IP?,” Fortinet, [Online]. Available: <https://www.fortinet.com/resources/cyberglossary/tcp-ip>. [Accessed 24 Apr 2023].
- [2] javatpoint, “TCP Connection Termination,” javatpoint, [Online]. Available: <https://www.javatpoint.com/tcp-connection-termination>. [Accessed 24 Apr 2023].
- [3] Computer Geeks, “The Two Generals’ Problem (And What You Can Do About It),” Computer Geeks, 3 Feb 2020. [Online]. Available: <https://geeks.co.uk/2020/03/two-generals-problem/>. [Accessed 24 Apr 2023].
- [4] Network Direction, “How MTU and MSS Affect Your Network,” Network Direction, [Online]. Available: <https://networkdirection.net/articles/network-theory/mtu-and-mss/>. [Accessed 26 Apr 2023].
- [5] L. Williams, “TCP/IP Model: What are Layers & Protocol? TCP/IP Stack,” Guru99, 8 Apr 2023. [Online]. Available: <https://www.guru99.com/tcp-ip-model.html>. [Accessed 24 Apr 2023].
- [6] P. Gupta, “Synchronous vs. asynchronous communications: The differences,” TechTarget, 18 Aug 2021. [Online]. Available: <https://www.techtarget.com/searchapparchitecture/tip/Synchronous-vs-asynchronous-communication-The-differences>. [Accessed 26 Apr 2023].
- [7] S. Lewis, “interoperability,” TechTarget, Feb 2019. [Online]. Available: <https://www.techtarget.com/searchapparchitecture/definition/interoperability>. [Accessed 27 Apr 2023].
- [8] IBM, “TCP/IP protocols,” IBM, 24 Mar 2023. [Online]. Available: <https://www.ibm.com/docs/en/aix/7.2?topic=protocol-tcpip-protocols>. [Accessed 24 Apr 2023].
- [9] javatpoint, “TCP/IP model,” javatpoint, [Online]. Available: <https://www.javatpoint.com/computer-network-tcp-ip-model>. [Accessed 24 Apr 2023].

- [10] Vikyath Ram, "The Internet Layer in the TCP/IP Model," tutorialspoint, 17 Jun 2020. [Online]. Available: <https://www.tutorialspoint.com/The-Internet-Layer-in-the-TCP-IP-Model>. [Accessed 24 Apr 2023].
- [11] Cloudflare, "What is UDP?," Cloudflare, [Online]. Available: <https://www.cloudflare.com/en-gb/learning/ddos/glossary/user-datagram-protocol-udp/>. [Accessed 24 Apr 2023].
- [12] K. Koishigawa, "TCP vs. UDP — What's the Difference and Which Protocol is Faster?," freeCodeCamp, 28 Jun 2021. [Online]. Available: <https://www.freecodecamp.org/news/tcp-vs-udp/>. [Accessed 27 Apr 2023].
- [13] U. Samariya, "TCP 3-Way Handshake Process," tutorialspoint, 23 Nov 2021. [Online]. Available: <https://www.tutorialspoint.com/tcp-3-way-handshake-process>. [Accessed 24 Apr 2023].
- [14] L. Williams, "TCP 3-Way Handshake (SYN, SYN-ACK,ACK)," Guru99, 4 Mar 2023. [Online]. Available: <https://www.guru99.com/tcp-3-way-handshake.html>. [Accessed 24 Apr 2023].
- [15] jakub, "Two Generals' Problem," Finematics, 5 Feb 2018. [Online]. Available: <https://finematics.com/two-generals-problem/>. [Accessed 24 Apr 2023].
- [16] L. Williams, "TCP/IP vs OSI Model – Difference Between Them," Guru99, 1 Apr 2023. [Online]. Available: <https://www.guru99.com/difference-tcp-ip-vs-osi-model.html>. [Accessed 25 Apr 2023].
- [17] imperva, "OSI Model," imperva, [Online]. Available: <https://www.imperva.com/learn/application-security/osi-model/>. [Accessed 25 Apr 2023].
- [18] D. P. Barrera, "Ethernet frames and packets: what's the difference?," NETWORKWORLD, 18 Sep 2017. [Online]. Available: <https://www.networkworld.com/article/3225865/ethernet-frames-and-packets-whats-the-difference.html>. [Accessed 26 Apr 2023].

- [19] javatpoint, “Difference between Frames and Packets,” javatpoint, [Online]. Available: <https://www.javatpoint.com/frames-vs-packets>. [Accessed 26 Apr 2023].
- [20] M. Rouse, “Maximum Transmission Unit,” techopedia, 26 Jul 2016. [Online]. Available: <https://www.techopedia.com/definition/3170/maximum-transmission-unit-mtu>. [Accessed 26 Apr 2023].
- [21] S. Iveson, “IP Fragmentation in Detail,” Packet Pushers Interactive, 26 Nov 2019. [Online]. Available: <https://packetpushers.net/ip-fragmentation-in-detail/>. [Accessed 26 Apr 2023].
- [22] S. Hogg and J. Fruhlinger, “MTU size issues, fragmentation, and jumbo frames,” NETWORKWORLD, 20 Oct 2021. [Online]. Available: <https://www.networkworld.com/article/2224654/mtu-size-issues.html>. [Accessed 27 Apr 2023].
- [23] M. Rouse, “Protocol Data Unit,” techopedia, 6 Nov 2014. [Online]. Available: <https://www.techopedia.com/definition/25292/protocol-data-unit-pdu>. [Accessed 26 Apr 2023].
- [24] annieahujaweb2020, “Protocol Data Unit (PDU),” geeksforgeeks, 29 Apr 2022. [Online]. Available: <https://www.geeksforgeeks.org/protocol-data-unit-pdu/>. [Accessed 26 Apr 2023].
- [25] D. Bevans, “Asynchronous vs. Synchronous Programming: Key Similarities and Differences,” Mendix Technology BV, 1 Feb 2023. [Online]. Available: <https://www.mendix.com/blog/asynchronous-vs-synchronous-programming/>. [Accessed 28 Apr 2023].
- [26] luminousmen.com, “Asynchronous programming. Blocking I/O and non-blocking I/O,” luminousmen.com, 26 Apr 2020. [Online]. Available: <https://luminousmen.com/post/asynchronous-programming-blocking-and-non-blocking>. [Accessed 28 Apr 2023].

- [27] Microsoft, “Synchronous I/O antipattern,” Microsoft, 16 Dec 2022. [Online]. Available: <https://learn.microsoft.com/en-us/azure/architecture/antipatterns/synchronous-io/>. [Accessed 28 Apr 2023].
- [28] BuildFire, “Synchronous vs. Asynchronous Programming: What’s the Difference?,” BuildFire, [Online]. Available: <https://buildfire.com/synchronous-vs-asynchronous-programming/>. [Accessed 28 Apr 2023].
- [29] MKS075, “Difference between Synchronous and Asynchronous Transmission,” geeksforgeeks, 21 Feb 2023. [Online]. Available: <https://www.geeksforgeeks.org/difference-between-synchronous-and-asynchronous-transmission/>. [Accessed 26 Apr 2023].
- [30] sakshamkaushik, “Difference between Asynchronous and Non-blocking,” geeksforgeeks, 21 Feb 2023. [Online]. Available: <https://www.geeksforgeeks.org/difference-between-asynchronous-and-non-blocking/>. [Accessed 28 Apr 2023].
- [31] A. Tarnowski, “Synchronous vs. asynchronous communication (in the cloud and beyond),” predica, [Online]. Available: <https://www.predicagroup.com/blog/synchronous-vs-asynchronous-communication/>. [Accessed 26 Apr 2023].
- [32] New America, “Interoperability Is Fundamental to the Internet,” New America, [Online]. Available: <https://www.newamerica.org/oti/reports/promoting-platform-interoperability/interoperability-is-fundamental-to-the-internet/>. [Accessed 27 Apr 2023].
- [33] S. Khudairi, “WIP -- The Web Interoperability Pledge,” W3C, 18 Jul 1997. [Online]. Available: <https://www.w3.org/Promotion/WIP/>. [Accessed 27 Apr 2023].
- [34] w3schools, “HTML Introduction,” w3schools, [Online]. Available: https://www.w3schools.com/html/html_intro.asp. [Accessed 27 Apr 2023].

- [35] M. E. Shacklett, "TCP/IP," TechTarget, Jul 2021. [Online]. Available: <https://www.techtarget.com/searchnetworking/definition/TCP-IP>. [Accessed 27 Apr 2023].
- [36] LinkedIn, "How do you compare and contrast different TCP/IP models and standards?," LinkedIn, 27 Apr 2023. [Online]. Available: <https://www.linkedin.com/advice/0/how-do-you-compare-contrast-different-tcpip>. [Accessed 27 Apr 2023].
- [37] N. Cranford, "The role of network interoperability in telecommunications," RCR Wireless News, 19 Apr 2018. [Online]. Available: <https://www.rcrwireless.com/20180419/fundamentals/the-role-of-network-interoperability-in-telecommunications>. [Accessed 27 Apr 2023].
- [38] INFOSPARK.IN, "The TCP/IP Reference Model and Protocol Suite Explained," INFOSPARK.IN, [Online]. Available: <https://infospark.in/networking/the-tcp-ip-model-and-protocol-suite-explained/>. [Accessed 27 Apr 2023].
- [39] RingCentral, "Frame Relay," RingCentral, 28 Oct 2021. [Online]. Available: <https://www.ringcentral.com/gb/en/blog/definitions/frame-relay/>. [Accessed 27 Apr 2023].
- [40] Dummies, "Network Administration: TCP/IP Protocol Framework," Dummies, 26 Mar 2016. [Online]. Available: <https://www.dummies.com/article/technology/information-technology/networking/general-networking/network-administration-tcpip-protocol-framework-184682/>. [Accessed 27 Apr 2023].

Task1FinalReportRj

GRADEMARK REPORT

FINAL GRADE

66/100

GENERAL COMMENTS

Instructor

A good report, well-sourced.

Your implementation is not quite there as your report mentions, however displays some good understanding of concepts and some original code heading in the right direction.

Part 1 (report): 82

Part 2 (implementation): 50

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

