

Assignment

by Kieran John Ayres

FILE	2962_KIERAN_JOHN_AYRES_ASSIGNMENT_349477_1490837388.DOCX (2.29M)		
TIME SUBMITTED	12-MAY-2021 11:40PM (UTC+0100)	WORD COUNT	3041
SUBMISSION ID	152952140	CHARACTER COUNT	15958



Prifysgol Cymru

Y Drindod Dewi Sant

University of Wales

Trinity Saint David

University of Wales Trinity Saint David

School of Computing

04/03/2021/ACCA6024

Assignment 2

By Kieran John Ayres

Table of Contents

Table of Contents

Table of Contents.....	2
Introduction	3
Task 1	Error! Bookmark not defined.
Transmission Control Protocol.....	4
Maximum Transmission Unit.....	5
Synchronicity and Asynchronicity in Network Protocols	7
Network Interoperability	7
Relevance to Programmers	Error! Bookmark not defined.
Task 2	Error! Bookmark not defined.
Features	Error! Bookmark not defined.
Design Decisions	Error! Bookmark not defined.
How it Adheres to Best Practice?	Error! Bookmark not defined.

Introduction

The focal point of this essay is to explore core concepts, in depth, in relation to socket programming and concurrency, topics include; TCP and the process in which it ensures data is delivered, the MTU and how that impacts transfer protocols. It will also explore and discuss Synchronicity and Asynchronicity in network protocols and what applications are these relevant to. Then it will discuss the meaning of interoperability in socket programming. The second section of the assignment will be to develop a network application and then, in response, to produce a short report to justify the decisions that were chosen when developing a networked application.

Transmission Control Protocol

Transmission Control Protocol, or TCP as it's more commonly known is a network protocol which is a standard for exchanging data between applications on a geographical scale [1]. TCP/IP (as it's also known) will enable the link between the network and the applications, or client and server. A standard network programming paradigm can be utilised to form the connection [1]. To do this, the devices are connected to a network, this could be a different network, as long as it's able to transmit packets of data to the corresponding network, to do this, a network device, such as a router can be used [1]. It ensures a guarantee of delivery, it is an "end-to-end protocol" [2] where each machine on the network has a state and a number which can trace the inbound packets of data in the order they were sent. When a packet has been sent, an ACK packet will be sent back to confirm to the device sending the data that it has received this packet and is awaiting, for the next packet in the sequence [2]. When the connection of TCP is formed. The three-way handshake occurs. Moreover, in doing this the two systems communicating establish a connection to one another. System A will send a sequence request to system B. System B will then send an acknowledgement (ACK) back, the ACK number will always be the sequence number plus one. Following this, system B will then send a sequence request back and then the client will receive the ACK, and then sent an ACK number with another sequence number and then the connection is established overall, ready for communication. The sequence numbers are important as they assist in ordering packets of data which may be not in the correct order, it can also assist in stopping any data duplication [3]. In performing the three-way handshake, the system is reliable as it then allows the systems to confidently transmit data [2]. It is also reliable as there is a time constraint into which data can be sent, once a packet is sent across the network, TCP places a copy of that packet within a queue, the timer then begins and should the acknowledgement not be received in this time, then the packet from the queue would then be sent over the network again [3]. Furthermore, TCP has an extra depth of reliability as errors in the data delivery do not stop the correct delivery occurring through internet system errors as it can recover from this [3] [4]. TCP is also built with the correct set of protocol features which ensures the correct delivery of data, the two generals' problem is a common problem in computer science and TCP meets requirements to correct the proposed impossibility of the two generals' problem. The two generals' problem is as follows; there are two armies, led by two generals, these generals are positioned with their respective armies on two different corners of a battlefield, and they need to communicate to arrange an attack on an enemy who they can only overcome on the basis of them both attacking at the exact same time. This is where the problem arises as they general A has to synchronically plan the attack with general B, to communicate this, general A has to send one of the messengers across the battlefield, detailing everything that is essential to know for the attack (location, position etc). This proposes the question that, is it enough to understand the detail of the attack? No it isn't, when general B receives this message, he sends a message back to general A, an acknowledgement to confirm, which general A would then confirm and so on and so forth, resulting in an infinite exchange. However, it is essential to the communication that they confirm because without confirmation they would assume that the messenger had been intercepted, which means the plan can't continue, or that general B has received the message, but general acknowledgement has been intercepted resulting in again, the plan not being executable. In this instance, it is difficult to understand whether the plan is confirmed, or the message has been intercepted [5]. This concludes that it doesn't have a guaranteed delivery of data whereas, the TCP has a guaranteed delivery through methods such as the retransmission of unreceived packets or the three-way handshake [3] [2].

Maximum Transmission Unit

The Maximum Transmission Unit, or MTU, is a value that TCP uses to configure the largest packet size, in bytes, which the network can send. It has a noticeable impact on the network's ability to send data. Firstly, it is pre-configured, if it isn't it will automatically enable at a default value [6] Usually this would be 1500 bytes. An MTU is a packet, with a large portion of the packet occupied by the portion of data [7]. As a developer, the size of the MTU is something to consider, larger packets of data are, in general, more efficient due to there being a smaller number of packets. However, should one of these packets fail then the speed in which they re-transmit will be longer as the size of the data packet is much larger. By using smaller data packets, if there is an error, it will re-transmit increasingly faster as the packet which would be re-sent would be significantly smaller. There are other things to consider, such as the Protocol Data Unit and it's relevant within the OSI model. A figure can be shown below of the OSI model.

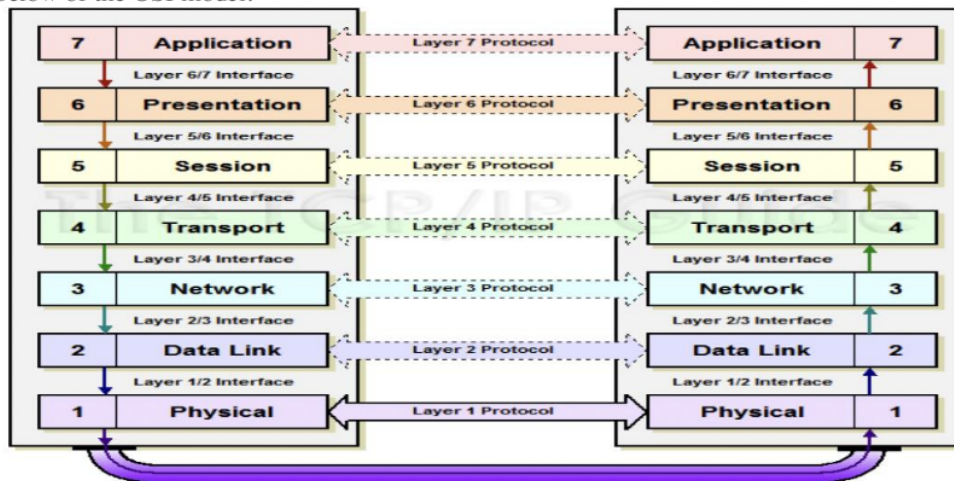


Figure 1 - OSI Model [8].

The idea of the OSI model was to maintain a degree of simplicity by each layer having a dedicated task in the processing of the data [8]. The OSI model is a architecture of seven layers that can be followed to understand the process of data when travelling through a network, every layer has communication with the next layer. This will be a form of message to the next layer to continue the process of sending the data over the network. This is advantageous as debugging can be done at the layer it occurred. The bottom of the stack, it begins with the physical layer (hubs, cables etc). The second layer is the data-link, this is the source and destination addresses. Layer three is the network layer, where the IP addresses are and the routing. The next layer is the transport layer, this is where the transport protocols are added, such as TCP. The fifth layer is then the session layer, which can determine the start and stop of the sessions, layer six is the presentation layer. At this layer, the data is formatted in a form that the application can understand but this also can encrypt, or decrypt data. The final layer is then the application layer, which is where the application and the user can communicate [8]. For example using SMTP (Simple Mail Transfer Protocol) [9]. On the example of SMTP and sending an email, it processes in this same order, that email goes through each layer of the OSI model, this is called encapsulation. The application layer creates the data that you created, the email addresses etc. and gets ready to send it using

SMTP(Simple Mail Transfer Protocol). Presentation, data is then formatted to a way in which the model understands such as ASCII. Session, will be receiving mail server has started. Transport layer then decides what protocol to use, (TCP to deliver every packet) then the source is added. Network layer, the IP addresses are added to the file (source and destination). Data-Link layer collects the MAC address from the router and the source mac address from the host and then adds these to the data. Physical, the data is then sent out through ethernet [8] [9]. There are four Protocol Data Units, or PDUs. It's the unit of data, that is passed through the network, this is essential to the network data transmission and passes through a specific layer of the OSI model, there are four layers they pass through, these are; physical, data-link, network and transport. The first layer it would pass through is the transport layer, this would be in the form of TCP header which contains details of the application. It then goes through the network layer, this is where the IP header would be attached to the TCP header where the application is, which grants the data a destination to reach, MAC addresses will also be added at this stage for this purpose. At this stage is where we add a "layer 2 data link frame" [10]. The data size of the PDU will be dependent on the MTU that was defined in the setup. This will determine whether fragmentation needs to occur. In some cases, when a packet is too large for a network, fragmentation occurs, this is the process of dividing large packets into smaller packets, in doing this the packets are small enough to continue the data transmission. This can affect efficiency due to the time it requires for fragmentation to complete [10]. Fragmentation occurs at the IP layer (OSI Model). So for example, if the MTU was configured at 1200 bytes, router A receives this and then needs to send this to Router B. However, Router B has an MTU of 800 bytes. Fragmentation would occur by dividing the 1200-byte packet into two equal sizes. At 600 bytes each, which would mean it could then be transferred to router B as it would then meet the MTU. Router B could then route them to the final destination. Defragmentation or Reassembly occurs with the targeted destination, they know to reassemble packets as there is an id or a unique identifier within those packets so the destination can identify those and then reassemble them from fragmentation [7] [11]. As developers, it is important that well designed networks are in place for when the processing of data occurs, it's important there are clear MTU's set so that the data can be as efficient as possible.

Synchronicity and Asynchronicity in Network Protocols

Firstly, synchronisation is achieved in networking predominantly for efficiency.

Synchronisation in terms of network would mean that they transmit data at the same frequency. Asynchronous differs in that it doesn't utilise a timing signal, so the devices are not synchronised during transmission [12] Asynchronous network protocols are also slower in comparison in the delivery of data [13].

Synchronous network protocols may include; TCP [14], BSC (Binary Synchronous Communication) and SDLC (Synchronous Data Link Control) [15]. Asynchronous protocols may include

As developers, synchronisation can be utilised to assist the code in avoiding problems. This is due to when developing multithreaded code, a singular thread may operate one section of code while others may operate another part of the application, problems could arise by not synchronising code. These include losing updates, an example of this is the ABA problem, If there were 2 threads, A and B, these could be operating simultaneously and would be accessing the same area of memory in preparation to update it. In this assumption, there is a integer equalling to five, and both threads would need to add an integer, A adding nine and B adding eleven to it to get the result of twenty-five. Each thread would go through the process at the same time. They would read, perform calculation and update the data. However, as B would perform this exact process as well, it would add eleven to five resulting in sixteen and in the process, updating thread A's value of fourteen. This would lose the updates [16]. These can be stopped by blocking threads through the use of mutexes, they would block from critical sections in the code.

Network Interoperability

Network interoperability can be interpreted differently depending on the situation. However, overall it is the idea that two networks can operate and communicate with each other while providing a consistent quality of service expected by whoever may be using that network [17]. Moreover, it is important that the network functions without any intervention from the end-user. It is especially important that the information exchange is direct [18]. The reason that this is an essential part to networked applications is that it grants connectivity between two systems. Interoperability is a global reaching technology, with it being a driving force behind networking applications. It allows any user with a mobile device to call any other mobile device through the Global System for Mobile Communication (GSM) [17]. It is beneficial for all users. The end-user would benefit from this as they can now have an open communication with any other device that wants to have communication, by having a fully, well designed interoperable network, the implementation can then have the best hardware available in terms of what achieves the best performance. By meeting these requirements, suppliers can sell their hardware anywhere [17]. There are level of network interoperability, similar to the OSI model, it has a physical level, a network level where functions for control would be, there's the application level, where the interface is [17].

There are some common ways in which interoperability, the first being through a network interface and the second is by utilising a network protocol, such as TCP. This can be done through a number of steps, first a specification must be created to achieve network interoperability, a business would need to have an interoperability standard set out for their specific application which outlines everything it would have to implement to achieve full network interoperability. This process would also include a cautious approach to the network design and planning as it must achieve interoperability. This should be thoroughly checked to ensure it covers all aspects, such as the requirements, architecture, and the protocols chosen for implementation. By having these as close to the design as possible allows the implementation to be far simpler. The implementation, once done, should be thoroughly tested in all environments to ensure it meets the requirements and it should then be put through a standards organisation, such as the ETSI, which is the European Telecommunication Standards Institute [17], this institute hosts events to allow businesses to have their applications thoroughly tested in a development environment to ensure they meet the standards before full deployment [17].

Conclusion

In conclusion, this assignment set out to research thoroughly. It has been a challenging assignment, a lot has been learnt through the research, such as, the different layers of the OSI model and how the PDUs transfer through them. Asynchronous and Synchronous applications, why they are used and what protocols are typically used for them. It has also investigated network interoperability and how this is best achieved, this is done by setting standards, designing and planning carefully and then running intensive tests at network standard events will allow the application to meet the standards.

References

- [e. a. Lydia Parziale, "Part 1. Core TCP/IP protocols," in *TCP/IP Tutorial and Technical Overview*, N/A, IBM Corp. , N/A, pp. 3-12.
1]
- [e. a. Fu-Hau Hsu, "TRAP: A Three-Way Handshake Server for TCP Connection Establishment," *Applied Sciences*, vol. 6, no. 358, p. 14, 2016.
2]
- [J. Postel, "Transmission Control Protocol," Defence Advanced Research Projects Agency
3 , Arlington , 1981.
4]
- [S. S. Pooja Parahar, "SOCKET PROGRAMMING," *Journal of Inventions in Computer Science and Communication Technology (JICSCT)*, vol. 3, no. 4, p. 5, 2017.
5]
- [N/A, "the "two-generals problem"," N/A, N/A, N/A.
6]
- [IBM, "Determining the maximum transmission unit," N/A N/A N/A. [Online]. Available:
7 [https://www.ibm.com/docs/en/zos/2.1.0?topic=overview-determining-maximum-](https://www.ibm.com/docs/en/zos/2.1.0?topic=overview-determining-maximum-transmission-unit)
8 [transmission-unit](https://www.ibm.com/docs/en/zos/2.1.0?topic=overview-determining-maximum-transmission-unit). [Accessed 08 05 2021].
- [D. Murray, "Large MTUs and Internet Performance," in *2012 IEEE 13th International Conference on High Performance Switching and Routing*, Belgrade, 2012.
9]
- [S. D. V. D. Sumit Kumar, "THE OSI MODEL: OVERVIEW ON THE SEVEN LAYERS OF COMPUTER NETWORKS," *International Journal of Computer Science and Information Technology Research*, vol. 2, no. 3, p. 6, 2014.
10]
- [V. Riabov, "SMTP (Simple Mail Transfer Protocol)," in *The Handbook of Computer Networks*, Bakersfield, John Wiley & Sons, 2007, p. 2.
11]

- [N/A, "Protocol Data Units - CompTIA Network+ N10-007 - 1.3," Professor Messer, N/A
1 N/A N/A. [Online]. Available: <https://www.professormesser.com/network-plus/n10-007/protocol-data-units/>. [Accessed 08 May 2021].
0]
- [I. S. Institute, "Darpa Internet Program Protocol Specification," Defense Advanced
1 Research Projects Agency, Marina Del Rey , 1981.
1]
- [D. Communications, "Part 11: Asynchronous and Synchronous Protocols," N/A N/A N/A.
1 [Online]. Available: https://www.angelfire.com/journal/brownr/b/datacomm/dc_011.htm.
2 [Accessed 09 05 2021].
2]
- [N/A, "Advantages and Disadvantages of Synchronous and Asynchronous Transmission,"
1 Raysync, 09 07 2020. [Online]. Available: [https://source.raysync.io/news/advantages-and-](https://source.raysync.io/news/advantages-and-disadvantages-of-synchronous-and-asynchronous-transmission)
3 [disadvantages-of-synchronous-and-asynchronous-transmission](https://source.raysync.io/news/advantages-and-disadvantages-of-synchronous-and-asynchronous-transmission). [Accessed 12 05 2021].
3]
- [Axway, "About Synchronous Communication," N/A N/A N/A. [Online]. Available:
1 [https://docs.axway.com/bundle/TransferCFT_36_UsersGuide_allOS_en_HTML5/page/C](https://docs.axway.com/bundle/TransferCFT_36_UsersGuide_allOS_en_HTML5/page/Content/appl_integration/synch_comm_tcpip_intro_1.htm#:~:text=Use%20synchronous%20communication%20(TCP%2FIP))
4 [ontent/appl_integration/synch_comm_tcpip_intro_1.htm#:~:text=Use%20synchronous%20](https://docs.axway.com/bundle/TransferCFT_36_UsersGuide_allOS_en_HTML5/page/Content/appl_integration/synch_comm_tcpip_intro_1.htm#:~:text=Use%20synchronous%20communication%20(TCP%2FIP))
] [communication%20\(TCP%2FIP\)](https://docs.axway.com/bundle/TransferCFT_36_UsersGuide_allOS_en_HTML5/page/Content/appl_integration/synch_comm_tcpip_intro_1.htm#:~:text=Use%20synchronous%20communication%20(TCP%2FIP)). [Accessed 09 05 2021].
]]
- [H. Austerlitz, "Standard Hardware Interfaces," in *Data Acquisition Techniques Using PCs*
1 (*Second Edition*), New York, Elsevier, 2003, p. N.A.
5]
- [B. CS, "The ABA Problem in Concurrency," Baeldung CS, 19 10 2020. [Online].
1 Available: <https://www.baeldung.com/cs/aba-concurrency>. [Accessed 09 05 2021].
6]
- [P. S. Paulo Teixeira de Sousa, "Telecommunications Network Interoperability,"
1 *TELECOMMUNICATION SYSTEMS AND TECHNOLOGIES*, vol. 11, no. N/A, p. 13,
7 N/A.
7]
- [S. Y. D. e. al, "Understanding Interoperability," in N/A, Norfolk, 2011.
1]
- [N/A, "TCP/IP," in *Securing HP NonStop Servers in an Open Systems World* , N/A,
1 Elsevier, 2006, p. N/A.
9]
- [G. Thomas, "INTRODUCTION TO THE TRANSMISSION CONTROL PROTOCOL,"
2 N/A, vol. 1, no. 5, p. 4, 2000.
0]

Assignment

GRADEMARK REPORT

FINAL GRADE

46/100

GENERAL COMMENTS

Instructor

A modest report is provided, but it's clearly quite rushed - the quality drops as it progresses. Some sources used, but their academic credibility is mixed.

The implementation provided is, unfortunately, a modest way from complete. An attempt made, nonetheless, displaying some understanding of key concepts.

Report (100): 62

Implementation (100): 30

Module average: 41

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10
