



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Socket Programming & Concurrency Assignment 1 - Report

By

Samuel Groves

Supervisor: Andrew Green

Second Marker: John Williams

May 2022

Contents

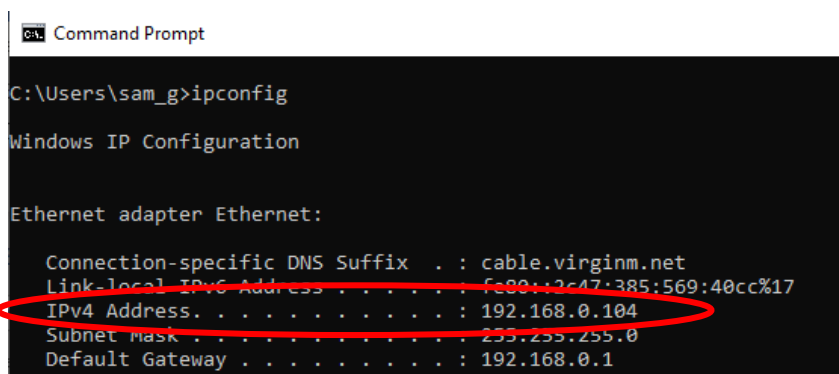
1. Transmission Control Protocol.....	3
1.1 How does TCP operate?	4
1.2 Three-way handshake and the “Two Generals’ Problem”	4
2. Maximum Transmission Unit (MTU) and the transfer of data	6
3. Synchronous and Asynchronous - Network Protocols.....	7
3.1 Synchronous.....	7
3.2 Asynchronous.....	7
3.3 The difference between Synchronous and Asynchronous	8
4. Interoperability	9
5. References	11

1. Transmission Control Protocol

Transmission control protocol (TCP) is a communication standard that enables programs and devices to exchange messages over a network. TCP organises data so that it can be transmitted between a server and a client/s. TCP also ensures data that is communicated over a network has integrity. This allows structured and reliable communication over the network without data loss.

TCP works with the internet protocol (IP). The internet protocol is a method of sending data from one device to another across the internet. Each device has a unique IP address that can be used to establish communication and exchange data with other devices.

There are different versions of IP addresses now available on machines. The most common (and somewhat outdated) is an IPv4 address. An IPv4 address is a 32-bit number that uniquely identifies a network interface on a machine. An example of an IPv4 address can be shown below.



```
Command Prompt

C:\Users\sam_g>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : cable.virginm.net
    Link-local IPv6 Address . . . . . : fe80::2c47:385:569:40cc%17
    IPv4 Address. . . . . : 192.168.0.104
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.1
```

The IPv4 address shown within the image is a private IP address. A public IP address can also be found.



This is the public IPv4 address used to connect to the internet on a machine. Due to the limited number of IPv4 addresses, a different version of IP addresses have been allocated to machines. IPv6 is the most recent. An IPv6 address is a 128-bit alphanumeric value used to uniquely identify a device. This can be shown below

```
Command Prompt

C:\Users\sam_g>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix . . . : cable.virginm.net
    Link-local IPv6 Address . . . . . : fe80::2c47:385:569:40cc%17
    IPv4 Address. . . . . : 192.168.0.184
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.1
```

1.1 How does TCP operate?

TCP splits large information or data into packets [1]. Packets are small amounts of data that are structured and sent to a device within a network. A data packet will contain small amounts of the overall information being sent across the network. Once all of the packets have reached their required destination, they are then reconstructed to show the information in its entirety.

Constructing packets increase data transfer efficiency and enables multiple pathways to a specific destination. These pathways are the most efficient method of data transfer as they optimise the use of transport capacity available along the devices which minimises the time it takes for data to pass across the network.

TCP instructs the packets how to be sent and reconstructed. TCP also establishes the initial connection of devices within a network [2]. This is completed by the implementation of multiple layers. There are four general layers to TCP model [3].

1. **Network Access Layer** – This layer handles the physical infrastructure that lets devices communicate with one another over the internet (using IP). The network access layer also includes the infrastructure that makes these network communications/connections possible.
2. **Internet Layer** – Sometimes referred to the network layer, controls the flow of data and the route that data is sent (network traffic). This ensures that data sent is accurate and follows the most efficient route to reach the end user which increases the speed at which data is obtained. The internet layer is also responsible for reconstructing the data packet once it has reached its destination.
3. **Transport Layer** – The transport layer ensures reliable data connection between communicating devices. This layer divides the data and information into packets. It also acknowledges the packet has been received by the recipient and the sender.
4. **Application Layer** – This is the final layer within the TCP model. The application layer allows a user to access the network. It is where the end-user will have the ability to see and interact with the data sent or received (such as email).

These layers are specific to TCP

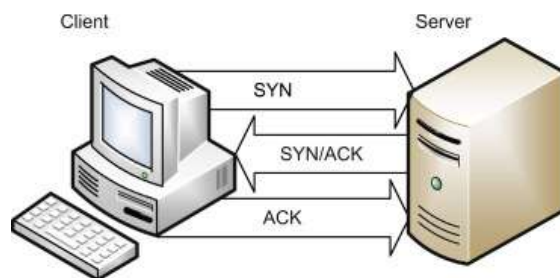
1.2 Three-way handshake and the “Two Generals’ Problem”

The three-way handshake is a method which is used within a TCP/IP network to create a connection and validate a communication between a local host (client) and a server. This method contains a

three-step process to initiate and negotiate the parameters within a TCP socket connection within a network before data is transmitted.

To conclude the reason why the three-way handshake exists, it is to create a TCP socket connection to send and transmit reliable data between devices [4]. As the name suggests, there are three different steps to this process.

1. A connection between the server and the client is established. The server must use open ports to accept and initiate new potential connections. The client must connect to the server by sending a data packet known as a synchronised sequence number (SYN). This is a request that the server will acknowledge if it is open and ready for new connections.
2. Once the SYN packet has been received by the server, it will respond to the client with an acknowledgement sequence number (ACK) packet which is a form of confirmation to the client. This packet also contains a newly generated SYN packet which indicates the client that the server is requiring its own acknowledgement.
3. The client receives the ACK packet along with the server-side SYN packet. The client responds with a new ACK packet to tell the server that their communication has been acknowledged. Once this process has been completed, the connection has been created. The host and the server can now communicate.



[5]

The “Two Generals’ problem” relates to the inability for devices to understand if their message has been received at the end of data exchanges [6]. Verification is provided to the device sending data that they have successfully sent a data packet/s, but there is no verification to the sender that the recipient has received their message. This is the core of the “Two General’s problem”. There is no guarantee that a data packet has been received by the recipient.

This could be due to network issues, packet errors or undisclosed disconnections. To combat this, servers should have the ability to manage disconnections. Specifically related to the three-way handshake, if some stages are not completed (SYN or ACK data packet has not been sent/received) this may lead to half-opened connections between the client and server [7]. One side is not aware that the connection has been closed.

TCP does not solve the “Two Generals’ problem” but there are methods which can decrease the chance of this occurring. The three-way handshake could send multiple SYN or ACK data packets at each stage in hopes that the client or server will obtain at least one of the messages. This can cause issues with the server or client as multiple packets could increase the network traffic leading to high load being placed on the devices (this may create slower network speeds and higher latency).

2. Maximum Transmission Unit (MTU) and the transfer of data

MTU relates to the maximum amount of information that can be contained within a packet. Furthermore, it is the size of the largest protocol data packet that can be communicated in a single network [8].

Once a packet has reached the specified MTU, it will be split into multiple packets. This is to ensure data loss does not occur (within the packet itself) and that the communicating devices within a network obtain all the data or information they require.

The default standard of MTU is 1500 bytes but this can be increased or decreased [9]. If the maximum size of the payload (packet) is increased more data can be sent within a packet and data transfer speeds can increase (improving the overall performance). This may result in a larger risk of packet loss and intermittent connection if a device within the network cannot manage the increased MTU.

Decreasing the MTU may resolve packet loss and fragmentation problems within the network but could lead to a decrease in overall performance as more data packets have to be sent. It is important to note that when using networked devices, the MTU should be assessed and configured so that the entire network is **optimised**.

Some networks may require large protocol data units (PDU) therefore, the MTU may need to be increased to create an optimised network. Some networks require small amounts of PDU, therefore the MTU may be decreased to again, create an optimised network. The responsibilities of the network should be assessed to designate the MTU allowed within a packet.

IP fragmentation may occur when a packet being sent is larger than the MTU. IP fragmentation is the breaking down of a packet into smaller pieces so that they can pass a link with a smaller MTU [10]. Different devices within a network may have different MTU allowances. IP fragmentation ensures that the recipient with a smaller MTU can still receive the 'large' packets sent by the **sender**.

We can assess how the forms of PDU change over the open systems interconnection model (OSI). This is a universal standard of communication functions for computing systems and telecommunication.

OSI Layer	Purpose	PDU State
7. Application	Services that are used with end user applications.	Data
6. Presentation	Formatting and encryption/decryption of data so that it can be viewed by the user.	Data
5. Session	Establishes or ends connections between hosts.	Data
4. Transport	Reliable transmission of data segments between network points including acknowledgement and segmentation.	Segment
3. Network	Reads the IP address from the packet to obtain destination (structuring, routing, and traffic control). IP Fragmentation occurs here.	Packet
2. Data Link	Reliable transmission of data frames between two connected nodes.	Frame
1. Physical	Transmission of raw bit stream through the physical medium.	Bits

Commented [AG4]: Accurate

Commented [AG5]: Accurate

Commented [AG6]: This could be clearer but is accurate

3. Synchronous and Asynchronous - Network Protocols

There are a large range of networks available today. Each network created will have specific requirements including the protocols used, their general purpose, location, and their type of communication. Synchronous and asynchronous communications are two different types of methods which can be used within a network. The type of communication will depend on the purpose of the network.

3.1 Synchronous

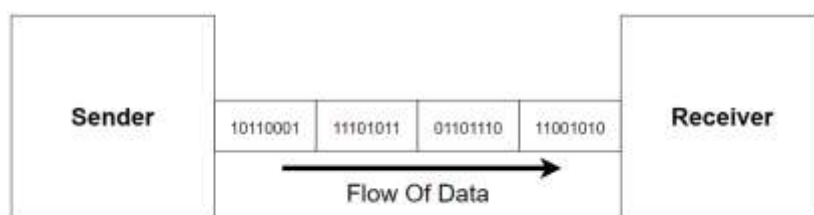
Synchronous communications occur in real time. This is where at least two individuals are exchanging information at the same time with each other. Within networking, this is where at least two devices are exchanging information across a network at the same time [12].

To understand synchronous communications, we must look at some examples of networks which use this form of communication protocol. Video conferencing requires synchronous communications [13]. Using this type of communication protocol allows a group of participants (within a video conference) to converse and collaborate through video, audio, and screen sharing in real time.

Using this communication protocol, video conferencing networks allow users to meet virtually instead of physically. This is beneficial for members who are not able to meet physically. This could be due to the geographical location of participants. Synchronous communications have increased exponentially in recent years due to physical office restrictions created during the COVID-19 pandemic. Another example of a network that uses synchronous communications are instant messaging services.

The largest disadvantage for synchronous communications are the availability of participants. Specifically, an office that communicates with global clients. Some may not be able to converse in real time due to time zones [14]. This might suggest that asynchronous communication could be the better option in this scenario. It is fully dependent on the requirements of the network, and the purpose.

Synchronous transmission of data is therefore a frequent stream of data in a time synchronised format. Data is transmitted at fixed intervals in the form of frames or blocks.



The flow of data transmission within a synchronous network is shown within the diagram. There is no interruption between the flow of data from the sender to the receiver. This means that there is a constant flow of data between the sender and recipient and there are no gaps within the data that is being sent.

3.2 Asynchronous

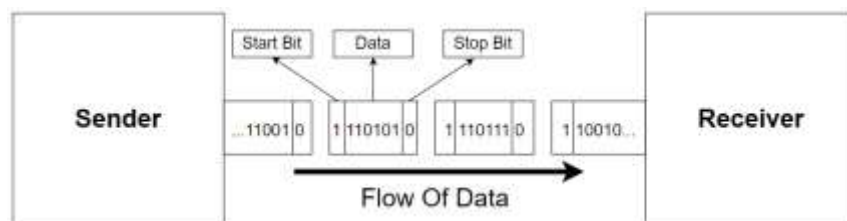
Asynchronous communications contain a time lag between sending and receiving data [15]. In the context of communication within a network, this is where a time lag occurs when a message is sent

and when the person receives and interprets it. Asynchronous communication is not a planned or scheduled event, but it can contain a string of back-and-forth exchanges between devices (users) within a network with the time delay **implemented**.

Email networks use this type of communication. One-way communications can be received by an audience and viewed at their convenience or through back-and-forth exchanges. The key to asynchronous communications within a network is the flexibility of the users [16]. asynchronous is also time-zone friendly due to the unrequired demand to respond to messages or communications **instantly**.

Due to the time delay applied to asynchronous communications, it is not suited for time sensitive communications within a network. Sending and responding to data/information must be extremely clear and detailed as there is no real-time exchanging of data.

Asynchronous transmission of data within a network works on start and stop bits [17]. Each character contains a stop and start bit and there are irregular time intervals between them. This creates a gap between the data due to this feature.



The diagram shows how an asynchronous network transmits data. The gap between the flow of data is created by the start and stop bits as shown. These gaps are the irregular streams of data that will be sent from the sender to the receiver. '1' is the start bit, followed by the data, then '0' is the stop bit. This is one complete data packet sent from an asynchronous network. This data is sent in the form of a byte or character and does not require synchronisation.

3.3 The difference between Synchronous and Asynchronous

There are a large range of differences between synchronous and asynchronous transmission within networks. These can be shown below.

Synchronous	Asynchronous
Real-time communication (fast transmission).	Time delay between sending and receiving data (slow transmission).
Frequent stream of data in the form of frames or blocks.	Irregular stream of data in the form of start and stop bits.
Immediate response from device or user within the network.	Flexible response time from device or user within the network.
Accuracy of received data depends on the receiver's ability to count the data streams (bits) accurately	Data is re-synchronised through start and stop bits, but limited synchronicity could create timing errors.
Bit-oriented	Character-oriented
No gap present between data.	Gap between data .

Commented [AG8]: Does not have to be delayed, just does not promise to be instant

Commented [AG9]: Good example

Commented [AG10]: I do not like tables used to make a comparison.

4. Interoperability

Network interoperability is the process of sending and receiving data amongst interconnected networks/applications through continuous means. This relates to the sharing and using of data between interconnected devices. The sending and receiving of data is at a quality level expected by the end user without negative impact [18].

Interoperability allows the sharing of interconnected devices within a network without any restrictions attached. Interoperability can be implemented within local area networks (LAN) or wide area networks (WAN) [19]. Interoperability allows management of resources across multiple different devices. This can improve the efficiency of data and resources shared as they can be found through any device with the network.

Interoperability is the real-time exchange of data between different systems communicating in a network. Sent data can be instantly understood and presented within interoperable networks. Networks that host interoperability may vary. There are four accepted levels of interoperability available to networks.

Level	Description
1 Foundational	A foundational level of interoperability allows data exchanges from one system to be received by another. It does not require the receiver to be able to fully interpret the data presented.
2 Structural	A structural level of interoperability establishes the data exchange format or structure within a network. This is to enable the movement of data from one system to another. Through this level, the meaning of the data, and the purpose, are preserved without any alterations occurring.
3 Semantic	A semantic level of interoperability provides the ability for two or more systems to exchange data and information. Within this level, systems or devices can fully interpret the data received. Semantic interoperability structures the data exchange to complete this.
4 Organisational	An organisational level of interoperability considers all organisational requirements and considerations (such as legal and social). This level refers to the ability for organisations to exchange data and information whilst addressing the organisational requirements and considerations. These requirements and considerations will also impact the design of the system and a network.

[20]

Many different industries use interoperability within their network. Healthcare systems have implemented interoperability to optimise and standardise the quality of medical care. Information and data such as patient records can be accessed by a range of different medical practices through different geographical locations. The ability to access this data and information can prove detrimental in many healthcare related scenarios.

There are challenges to implementing interoperability within a network. Complex features may need to be added to the existing architecture to achieve full network compatibility. Depending on the level of interoperability, the requirements will change. The highest level (organisational) will require the most complex features due to the range of features and design considerations required. This is usually the most expensive solution.

In order to achieve network interoperability between two or more networks, they must confirm/follow a common protocol standard or define a standard interface that all networks wishing

to interoperate must adhere [21]. A gateway can then be added so that data and information can be processed and translated between the network/s.

Some examples of standards are HTML, SQL, and XML. The internet uses a range of countless standards together to create an all-accessible platform that is continually expanding based on user contribution. In this case, users are not restricted to any single communication method or data structure.

Clear rules of communication and how the data is formatted and transferred should be applied to best achieve interoperability within a network. Networks that wish to achieve interoperability should contain the budget to do so.

Before the implementation of standards, the architecture, requirements, and protocols of the network must be assessed to understand the level of interoperability achievable and required. Conformance and interoperability testing should then be performed to ensure the network, and devices within the network, conform with the standards and protocols.

5. References

1. Why Do We Break Data into Packets? – Tech Evaluate [Internet]. Techevaluate.com. 2022 [cited 17 April 2022]. Available from: <https://www.techevaluate.com/why-do-we-break-data-into-packets/>
2. What Is TCP/IP? [Internet]. Avg.com 2022 [cited 17 April 2022]. Available from: <https://www.avg.com/en/signal/what-is-tcp-ip#:~:text=There%20are%20four%20layers%20of,when%20the%20data%20is%20received.>
3. Dominguez J. An Overview of each layer within the TCP/IP model [Internet]. Giac.org. 2002 [cited 17 April 2022]. Available from: <https://www.giac.org/paper/gsec/2233/overview-defense-in-depth-layer-tcp-ip-model/103817>
4. Shalom N. What is a Three-Way Handshake? - Definition from Techopedia [Internet]. Techopedia.com. 2020 [cited 28 April 2022]. Available from: <https://www.techopedia.com/definition/10339/three-way-handshake>
5. Conrad E, Feldman J. Three-way Handshake. Sciencedirect. 2022. Available from: <https://www.sciencedirect.com/topics/computer-science/three-way-handshake>
6. Ldapwiki: Two Generals Problem [Internet]. Ldapwiki.com. 2016 [cited 28 April 2022]. Available from: <https://ldapwiki.com/wiki/Two%20Generals%20Problem#:~:text=Overview%5B1%5D%5B2%5D%5B3%5D%23&text=Two%20Generals%20Problem%20shows%20that,failures%20of%20communication%20are%20possible.>
7. Finematics. Two Generals' Problem [Internet]. 2018 [cited 28 April 2022]. Available from: <https://www.youtube.com/watch?v=s8Wbt0b8bwY>
8. Configure MTU to maximize network performance [Internet]. Projectcalico.docs.tigera.io. 2022 [cited 1 May 2022]. Available from: <https://projectcalico.docs.tigera.io/networking/mtu>
9. Kaelin M. How to optimize the MTU for better performance [Internet]. TechRepublic. 2021 [cited 1 May 2022]. Available from: <https://www.techrepublic.com/article/how-to-optimize-the-mtu-setting-on-your-router-for-better-performance/>
10. Iveson S. IP Fragmentation in Detail - Packet Pushers [Internet]. Packetpushers.net. 2019 [cited 12 May 2022]. Available from: <https://packetpushers.net/ip-fragmentation-in-detail/>
11. Hozan E. How Data Moves Across a Network [Internet]. Secplicity. 2019 [cited 16 May 2022]. Available from: <https://www.secplicity.org/2019/03/01/how-data-moves-across-a-network/>
12. Vasseur J, Dunkels A. Message passing interface communication protocol [Internet]. Science Direct. 2015 [cited 12 May 2022]. Available from: <https://www.sciencedirect.com/topics/computer-science/synchronous-protocol>
13. Synchronous vs. Asynchronous Communication: A Guide For Managers [Internet]. Fellow.app. 2020 [cited 12 May 2022]. Available from: <https://fellow.app/blog/remote/synchronous-vs-asynchronous-communication-guide-for-managers/#:~:text=Basically%2C%20synchronous%20communications%20happen%20in,one%20or%20the%20other%2C%20respectively.>
14. Difference Between Synchronous and Asynchronous Transmission [Internet]. BYJUS. 2022 [cited 12 May 2022]. Available from: <https://byjus.com/gate/difference-between-synchronous-and-asynchronous-transmission/#:~:text=Synchronous%20transmission%20is%20faster%2C%20as,own%20start%20and%20stop%20bit.>

15. Synchronous and Asynchronous Transmission [Internet]. Raysync. 2020 [cited 14 May 2022]. Available from: <https://www.raysync.io/news/advantages-and-disadvantages-of-synchronous-and-asynchronous-transmission>
16. Difference between Synchronous and Asynchronous Transmission [Internet]. GeeksforGeeks. 2020 [cited 14 May 2022]. Available from: <https://www.geeksforgeeks.org/difference-between-synchronous-and-asynchronous-transmission/>
17. Williams L. Difference between Synchronous and Asynchronous Transmission [Internet]. Guru99. 2022 [cited 15 May 2022]. Available from: <https://www.guru99.com/difference-between-synchronous-and-asynchronous-transmission.html>
18. Tong G, Achilles T. The Rising Importance of Organizational Interoperability [Internet]. Healthcareimc.com. 2017 [cited 17 May 2022]. Available from: <http://www.healthcareimc.com/main/the-rising-importance-of-organizational-interoperability/#:~:text=The%20term%20organizational%20interoperability%20refers,impact%20system%20design%20and%20use>.
19. What is Interoperability? [Internet]. Heavy.ai. 2019 [cited 18 May 2022]. Available from: [https://www.heavy.ai/technical-glossary/interoperability#:~:text=Interoperability%20is%20the%20property%20that,wide%20area%20networks%20\(WANs\)](https://www.heavy.ai/technical-glossary/interoperability#:~:text=Interoperability%20is%20the%20property%20that,wide%20area%20networks%20(WANs)).
20. Interoperability | RingCentral UK Blog [Internet]. RingCentral UK Blog. 2020 [cited 18 May 2022]. Available from: <https://www.ringcentral.co.uk/gb/en/blog/definitions/interoperability/>
21. Yocabet C. The Key to Connectivity: Interoperability & Open Networks | TIE Kinetix [Internet]. Tiekinetix.com. 2021 [cited 18 May 2022]. Available from: <https://tiekinetix.com/en/blog/key-connectivity-interoperability-open-networks>



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Socket Programming & Concurrency Assignment 1 - Report & Implementation

By

Samuel Groves

Supervisor: Andrew Green

Second Marker: John Williams

May 2022

Contents

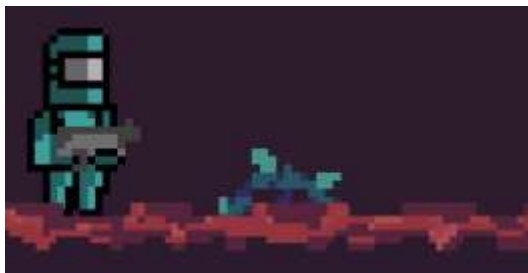
Design & Development of Application.....	15
1. Design Decisions	15
2. Features	17
3. Best Practice.....	17

Design & Development of Application

This report contains the design decisions, features implemented, and the best practices adhered through the multiplayer 2D game created.

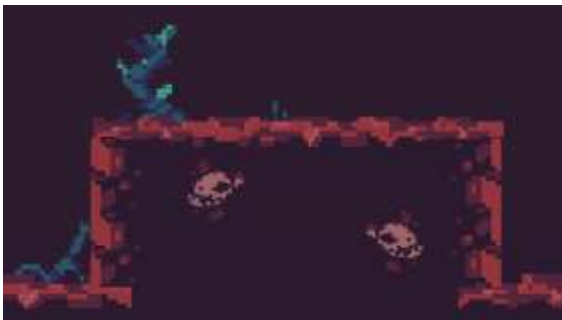
1. Design Decisions

The initial stages required understanding the resources needed to design a multiplayer game. Specifically, the game engine, software, and the resources required to develop the game (such as assets). A plan of the game, its designs, features, and risks were created.



Unity was chosen as the game engine due to range of different free assets available, and the compatibility with different programming IDE's. Unity also contains a large range of features which can be utilised to design a game. The programming IDE chosen was Visual Studio. This was chosen due to having familiarity with this IDE and the compatibility with the Unity game engine.

The second stage of design decisions were related to the type of multiplayer game to design, and the resources required. As briefly introduced, a multiplayer 2D game was decided. Two players are able to play the game. If more than two players attempt to join the game, they are unable to connect, and exceptions are given.



The third stage of design decisions required the understanding of the technologies and protocols to implement to create the multiplayer game. Through research and teachings, TCP and UDP were found to be the most popular network protocol in regard to multiplayer games. Therefore, these protocols were implemented into a server to allow communication between clients within the network. One such communication is the player movement.

TCP was used to manage important data such as supplying effective and efficient connections to users. UDP was implemented to manage less important tasks and data such as the player movement.

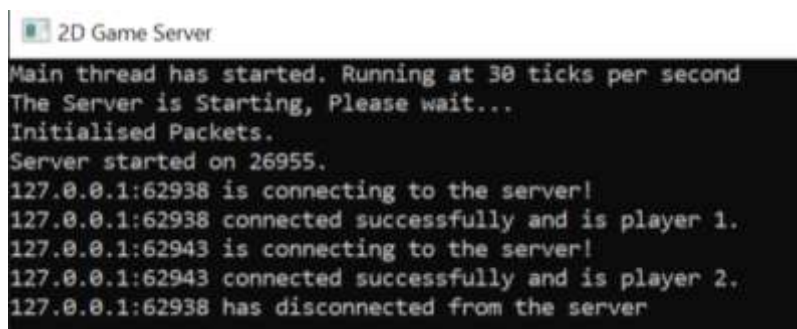
Along with the server script, a range of other scripts had to be implemented on the server side. These scripts were:

- **Packet Script** – This script was used to structure the data sent to clients within the network. This script ensured that sent data is in a specific format and that it can be read by client-side technologies.
- **Player Script** – This was used to store the positions of users, their ID (allocated by the server) and their username (entered upon connecting to the server within the game).
- **Client Script** – This script controlled the data buffer size (the amount of memory used to store temporary data) allocated to the server. Implemented within this script were the connection sockets (TCP and UDP) and the data that was required to be sent to the client. Disconnections were also handled within this script.
- **Program Script** – This script was implemented to start and manage threads and to start running on the ports selected within the client-side “client” script alongside the number of ports available to users (two as there are only two players).
- **Server Handler Script** – To check user ID and usernames and add them to within the game.
- **Server Send Script** – To send required data to clients.
- **Thread Manager Script** – Manage threads and execute them
- **Game Logic Script** – Manage player movement and constantly update by sending to thread manager.
- **Constants Script** – Contains the speed that the server and thread is running. Initialised and shown within the start of the server.

Along with the server script, a range of client-side scripts had to be implemented into the game. Mostly containing game logic and player movement inputs. Some scripts are very similar to the server side such as the packet, client, client handler and the thread manager.

A game manager script was implemented to spawn players within the game. A player manager script to assign them their ID and selected username. A score script, to provide users with a score, and finally a quit script to end the application.

Code is run on one thread due to the complexity of implementing multi-threading technologies within a multiplayer game.

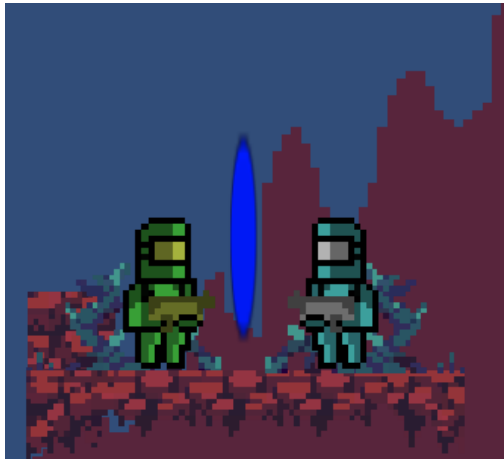


```
2D Game Server
Main thread has started. Running at 30 ticks per second
The Server is Starting, Please wait...
Initialised Packets.
Server started on 26955.
127.0.0.1:62938 is connecting to the server!
127.0.0.1:62938 connected successfully and is player 1.
127.0.0.1:62943 is connecting to the server!
127.0.0.1:62943 connected successfully and is player 2.
127.0.0.1:62938 has disconnected from the server
```

Shown above is the server managing connections and disconnections, alongside the starting of the server and main threads.

2. Features

The goal of this 2D multiplayer game is to reach the end “portal” (goal) to complete the level. There is a score that depreciates every second and is shown to the player once they have reached the objective. The players can then converse with each other, attempting to reach the best score possible.



Two players spawn in the starting location and must run through an obstacle course to reach the end goal. The design is unique, and each player has their own movement ('A' to move left, 'D' to move right, and 'Space' to jump). Each player will appear a different colour to the other as to not create confusion.

3. Best Practice

The port number for the server (the port users/players connect to) is 26955. This is an unused port number. If this multiplayer game were to be released publicly, this will allow users to connect to the server (if a reserved or used port is added, users would be unable to connect).

As briefly described, a data buffer size was implemented within the server so that temporary data could be contained. If this data buffer size was too large, it may place strain on the server and the latency of the network could suffer, hence the implementation. The buffer size implemented was 4096 bytes.

Code throughout the server side and client-side scripts used exceptions and debugging features during creation and within the final design. This allows not only clients, but the server to provide information on why a specific event or error has occurred.

```
Console.WriteLine($"Error sending data to player {ID} via TCP: {ex}!");
```

Code was thoroughly commented. There was a large range of code required to design the proposed application. Implementing comments within the code provided thorough instructions on what occurs when specific functions are called and used. This allows for easier debugging, alongside a better understanding of how the server and client communicate.