# Assignment 1

## Introduction

Most assignments for this course are practical in nature. This first assignment however is an exception to that rule, since we want to make sure you have a good grasp on the basics and have plenty of time to get used to the quirks of TCP/IP communication using the example provided during the lecture, before we dive into the more complicated stuff.

## Sufficient

Review the lecture material and answer the questions below.

1. Why are networks modelled using a layer stack?

|  |
| --- |
| To simplify the design and handle the complexity of networks better. |

1. What is an IPAddress and what is the valid structure of an IPAddress?

|  |
| --- |
| Is a numerical label assigned to a device that uses Internet Protocol, are identifiers that allow information to be sent between devices, contain location information.  Valid Structure:  A 32-bit number  We know it as 4 dotted decimals  *172.16.254.1* |

1. What is a loopback address and what do you use it for?

|  |
| --- |
| Is an address that sends signals to the same computer for testing. It allows a user to treat the local machine like a remote machine. |

1. What is a URL and what is its use?

|  |
| --- |
| It is a reference that specifies the address of a web resource. It is used to alias IPAddreses. |

1. What is the difference between the IP protocol vs the TCP/UDP protocols?

|  |
| --- |
| IP protocol defines hosts, TCP/UDP define ports/applications. |

1. List 3 differences between the TCP & UDP protocol.

|  |
| --- |
| UDP – Fast, with errors  TCP – Slower, with error check  UDP – out of order packets  TCP – packets are ordered  UDP - connectionless  TCP – connection oriented |

1. What is a port and what port range should you use as an application programmer?

|  |
| --- |
| Ports are numbers that represents an ID. The port range for applications is: 49152-65535 |

1. What is the difference between a dedicated and a non dedicated server?

|  |
| --- |
| A dedicated server doesn’t rely on clients to host the session and can exist even if there are no clients – it’s a separate application. A non-dedicated server needs a host so that a session can exits and the session will end if the host ends it. |

## Good

Review the lecture material and answer the questions below.

1. What are the 5 layers of the discussed networking stack and what is the purpose of each?

|  |
| --- |
| The 5 layers are:   1. Physical/link    1. Transmits data across medium 2. Internet protocol    1. Defines addresses for hosts and how data is transmitted between them 3. Transport    1. Defines addresses for applications and how data is transmitted between them 4. Sockets    1. Defines the endpoint for sending/receiving data, it binds a port to the local machine and listens for connections 5. Applications    1. Defines an agreement on what data the game/application requires/is able to process. |

1. You are playing a network game where one player can shoot another player.  
   a) List the network messages (in regular English/JSON/XML) between client & server for both a client authoritative setup and a server authoritative setup for such an event from the moment a player presses the fire button.

For example if I wanted to describe a login message I could write something like  
<Login name="..." pass="..." /> or Login = { user:"...", pass:"..." } to make it clear what kind of message I want to send and what data it contains.

PressFire={weaponType:”Rifle”,currentAmmo:”1”, requiredAmmo:”2”}

|  |  |
| --- | --- |
| **Client authoritative** | **Server authoritative** |
| C: Fire the rifle, no ammo! BTW other player is hit  S: Ok. You have fired the rifle. And other player is hit | C: Fire the rifle!  S: Not enough ammo. Sorry.  C: But other player is hit right?  S: No. |

b) Explain whether the messages from the previous question are IP, Transport or Application protocol messages.

|  |
| --- |
| Application protocol. The data is relevant only for application protocol. |

1. Lecture 1 discussed two common setups for building a network game (Peer 2 peer & Client/Server). List/research some advantages/disadvantages of both setups.

|  |
| --- |
| Peer to peer   * Advantages   + Easier to setup   + The network can extend easily to include new clients   + If one computer fails it doesn’t disrupt other parts of the network * Disadvantages   + Performance issues   + There is little security   Client/Server   * Advantages   + All files are stored in a central location   + Backups and security are controlled centrally   + Users can access shared data * Disadvantages   + If any part of the network fails a lot of disruption can occur   + Professionals are required |

1. List which protocol type is more appropriate for each of the message types below:
   1. Login message UDP / TCP
   2. Fast paced position update messages UDP / TCP
   3. Video stream data UDP / TCP
   4. Player hit messages UDP / TCP
2. For each situation below indicate which protocol has been used (more than 1 correct answer possible):
   1. The client sends 1 message and the server receives it: UDP / TCP
   2. The client sends 1 message and the server receives it twice: UDP / TCP
   3. The client sends 2 messages and the server receives them in order: UDP / TCP
   4. The client sends 2 messages and the server receives them out of order: UDP / TCP

## Very good

Examine the ***001\_basic\_tcp\_echo\_server\_commented*** example and answer the questions below:

1. What do we mean with blocking operations?

|  |
| --- |
| An operation which prevents the code after it from being executed, until a required action has been made. |

1. Name 2 different blocking network operations used in the given example.

|  |
| --- |
| listener.AcceptTcpClient()  Console.ReadLine() |

1. List some exceptions that might occur while trying to communicate over the network.  
   (Hint: check the code hinting or look up some network calls on MSDN)

|  |
| --- |
| OutOfMemoryException, ArgumentNullException, SocketException |

1. Imagine you've been given a client and server without any error handling code and you only have   
   time to fix one of them. Which one would you fix and why?

|  |
| --- |
| The server should be fixed. The errors should be caught by the code that calls the client. If an error occurs the server decides what happens to the client-if it is still usable or it needs to be aborted. |

1. Why does the client stop working in the given example if you send an empty string?

|  |
| --- |
| An empty string is 0 bytes. The server is trying to read 1k bytes to a byte array, which will keep it waiting. |

## Excellent

1. Start the server and **two** clients. Why is the server only responding to the first client?

|  |
| --- |
| The second client isn’t accepted by the server while the first client is. |

1. Start the server and **two** clients. What is the simplest way without making any code changes to have the server respond to the last client?

|  |
| --- |
| End the connection with the first client. |

1. How can you prevent a client from connecting at all if there is already another client waiting to be served? (Hint: research the TcpListener.Start call)

|  |
| --- |
| TcpListener.Stop. This way the queue of connections will be lost. |

1. What happens to clients that are trying to connect, but have not yet been accepted by the TcpListener.AcceptTcpClient call?

|  |
| --- |
| They can send messages, but nothing will happen with the server until the client has been accepted. |

1. Replace line 17 of the client with:   
   TcpClient client = new TcpClient(new IPEndPoint(IPAddress.Any, 55556));  
   Start the server and two clients again and note what happens.

**System.Net.Sockets.SocketException:** 'Only one usage of each socket address (protocol/network address/port) is normally

Undo line 17 and repeat, note the port the clients are connecting two.

Select the correct statement below and motivate your answer:

a) It is not possible to bind more than one TcpClient to the same port.

b) Multiple TcpClients can be bound to the same port

c) Multiple TcpClients can be bound to the same port on the server but not on the client

d) Multiple TcpClients can be bound to the same port on the client but not on the server

|  |
| --- |
| c) The server listens on a single socket. If the clients are bound to the same socket, a socket exception will occur. |