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?
   - It helps split the application to different sub parts and lower the complexity of working on it.
   - It also allows for each layer to serve its own responsibility.
   - Networks are very complicated as they work with different media types, devices and applications with different demands, hence it is split into separate network layer stacks

2. What is an IPAddress and what is the valid structure of an IPAddress?
   An IPAddress is the digital address of a computer that is used to identify the location where data is to be sent to. The structure of an v4 IP address is a 32-bit number, such as "0xf0000000" which is simplified from its original format to a 4 dotted decimal value such as "192.168.0.1".

3. What is a loopback address and what do you use it for?
   A loopback address is an IP address that goes back to its sender. It can be used for testing or running a local server.

4. What is a URL and what is its use?
   A URL is an alias for an IP Address. It is used to make it easier to remember an IP Address and if an IP address changes you can still connect to it through a URL

5. What is the difference between the IP protocol vs the TCP/UDP protocols?
   TCP/UDP both add an app address on the system whereas IP is just the address of the computer

6. List 3 differences between the TCP & UDP protocol.
   UDP:
   - Connectionless.
   - Creates Duplicates.
   - Data can be out of order.

   TCP:
   - Connection oriented.
   - Guaranteed delivery.
   - Ordered delivery.

7. What is a port and what port range should you use as an application programmer?
   Ports are IDs so that UDP/TCP knows where the data is to be sent to (for an application specifically). As a programmer the

dynamic ports are what should be used (49152-65535)

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

A dedicated server is a specific application that is installed on a pc but the game/app is not being used on it, only for running the server.
A non-dedicated server runs through the client program and creates a server for other clients to connect to.

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?

Bottom to top

Physical/Link layer:

- It is a combination of two layers into one
  - Actual physical aspects (cable, wifi)
  - The technology that turns bytes into signals and vice versa
- It transmits data across a specific medium in frames
- Defines things such as:
  - The ways to identify a host so you can adress the host
  - The byte/frame format
  - How to convert bits into signals and vice versa

IP Layer:

- Is the agreement on how packets are routed between systems.
- It takes the complicated process of moving packets through a network and makes it look like a single end to end channel
- Makes use of Ip addresses to send the data

Transport Layer:

- Is the agreement on how packets are routed between apps
- Uses a port to identify hat app to send the data to

Socket:

- It is one endpoint of a two-way communication link between two programs running on a network.

Application:

- Is the agreement on what sort of messages the app requires or is able to process

2. 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.

Client authoritative	**Server authoritative**
Client: Client shoots at enemy and kills them. Server: Understood.	Client: Client inputs the fire button Server: The client fired their gun, they were aiming at another player and hit them dealing x damage, target dies.

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

These messages would be application protocol messages since they are data that is meant to be processed by the application.

3. 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 2 Peer:

- Advantages
  - It is simple and easy to setup for users, requiring only an active internet connection from the players/a hub to connect locally to one another
  - No requirement for a dedicated server (saves costs)
- Disadvantages
  - The performance is much more limited as the player hosting needs to have a good enough computer to run both the game and host at the same time
  - It is difficult to prevent cheating unless there is a designated authoritative peer.
  - A poor internet connection on the host if online would cause a poor connection for the rest of the players.
  - If the host disconnects, everyone does.

Client/Server:

- Advantages
  - Certain types of cheating can be prevented much more easily
  - The internet connection of one player does not directly affect another's games
  - The server's performance is not tied to any player so it will run smoother in most cases.
- Disadvantages
  - Will cost extra money to run the server in comparison to P2P
  - If the server has problems, everyone connected to it will experience it as well.

4. List which protocol type is more appropriate for each of the message types below:
   a. Login message — TCP
   b. Fast paced position update messages — UDP
   c. Video stream data — UDP
   d. Player hit messages — TCP

5. For each situation below indicate which protocol has been used (more than 1 correct answer possible):
   a. The client sends 1 message and the server receives it: — UDP / TCP
   b. The client sends 1 message and the server receives it twice: — UDP
   c. The client sends 2 messages and the server receives them in order: — UDP / TCP
   d. The client sends 2 messages and the server receives them out of order: — UDP

Very good

Examine the ***001_basic_tcp_echo_server_commented*** example and answer the questions below:

1. What do we mean with blocking operations?

   Blocking operations means that the code will not continue until the call conditions are met

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

   TcpListener.AcceptTcpClient() and NetworkStream.Read() are both blocking calls

3. 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)

   System.Net.Scokets.SocketsException (multiple errors, can use …SocketException.ErrorCode to find exact error occuring)
   System.InvalidOperationException (The listener has not been started with a call to System.Net.Sockets.TcpListener.Start/The client is not connect to a remote host)
   System.ObjectDisposedException (The client was closed)

4. 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 because all clients depend on a server, and I would rather have separate clients encounter issues than the entire server being problematic.

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

   The server waits for data to be sent which an empty string sends 0 bytes of data. If none is sent, it assumes it's still on the way/has not been sent yet. As well since .Read is a blocking call the client cannot perform anything until the server's "echo" happens.

Excellent

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

   The structure of the server makes it so that it only accepts the first client due to the while loops. Meaning no clients after will be able to connect whilst the first client stays connected.

2. 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?

   Disconnect the first client.

3. 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)

   Use the Start call with a maximum number of pending connections.

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

   They output that they have connected to the server but inputs are not being received from the client to server because the server is only reading data from the first client's stream

5. 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.
   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

   A) It is not possible to bind more than one TcpClient to the same port. One one usage of a socket address is permitted.