3D3 – Computer Networks: Project 2

Group Number: 32

Group Members: Jamie Columb (19335427)

Shruti Kathuria (21355061)

A: USE CASE

For this project we as a group have decided to focus on applying a peer-to-peer architecture to a network of autonomous underwater vehicles (AUV's). The purpose of an AUV is to gather data from the ocean using on-board sensory equipment without the need for a human operator. The data gathered by the AUV is then sent to the surface using a client-server architecture. For more complex operations, it would be beneficial to have a group of AUV's communicating and sharing collected data with each other to improve coordination and efficiency. We as a group propose that these AUV's should use a peer-to-peer network to intermittently communicate where each AUV is a peer capable of sharing and distributing files to the other AUV's in the network. There are a number of constraints when dealing with underwater communications. The primary constraint is the limited range of communication. AUV's use acoustic modems to send and receive signals and due to the limited battery life, they have to limit the range in which they cand send these signals. To mitigate this, each AUV could use data from received signal from other AUV's to adjust their course to ensure they are always within range of the groups signals.

We are using TCP to make the client and the server connection and further for clients to interact the UDP is used in the code to make a peer-to-peer connection.

B: Picture / Diagram / Logical Diagram

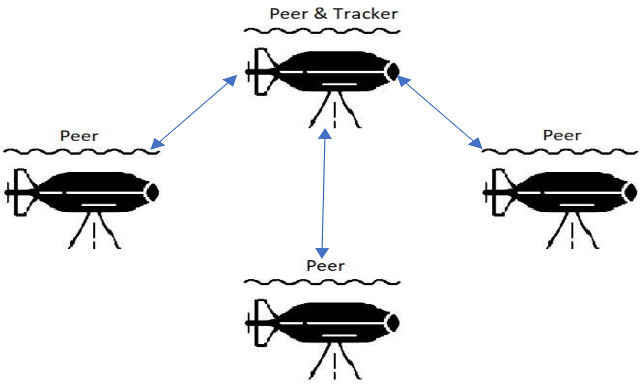


Fig.1 AUV's Registering to the Tracker

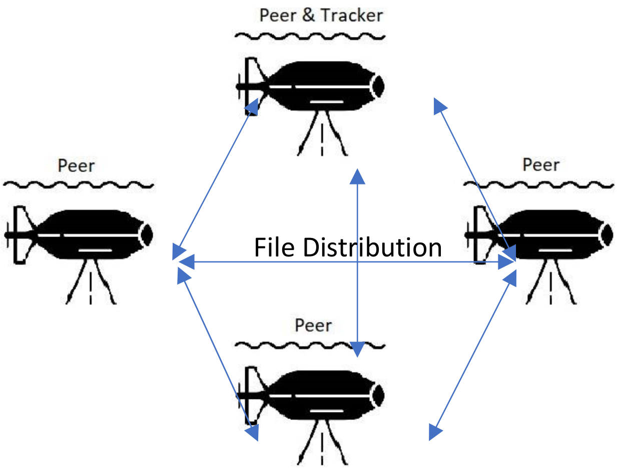


Fig.2 File Distribution Between AUV's Using P2P Architecture

C: High level Messaging/ Functionality/Functional Description/Purpose

Initially, the only interaction is between the device wishing to connect to the peer-to-peer network and the tracker. The device registers itself with the tracker and provides the correct key. In return, the tracker gives the device a list of all the peers in the network. Once this device is connected to thenetwork it becomes a peer, and since any device that connects is previously authorized, they all haveequal trust boundaries.

Once the AUV begins to gather data, it can start to share this data in chunks with its neighboring peers using UDP. The AUV will also share location data so that if one AUV begins to veer off course, it knows the location of the rest of the group and can adjust its movements. This is why its beneficial to have equal trust boundaries so that important data can be shared as fast as possible.

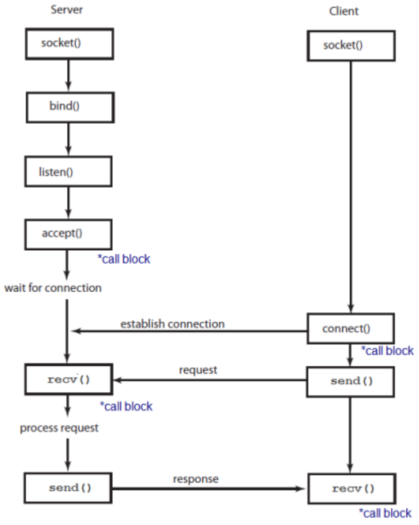
D: Algorithm/Pseudo code/Structure/Logical Thinking

Pseudo code for TCP Server:

```
•Import the socket library
•Create a socket through socket programming
•Start thread:
        Listen for connections on the socket.
        When a connection has been made, use accept() function and store the connection object and
        address information in two separate arrays.
        If two clients connect to the server, jump to handle_client thread.
•Handle_client thread:
        Separate the two elements in the clients array.
        Extract the clients IP and port number from the tuple.
        Send the IP and port number to the other client.
```

Pseudo code for TCP Client:

```
•Create a Server Socket
•Connect to the tracker server and share IP and port number with the server.
•Bind Socket to a specific port where the client can contact you.
•When two clients are registered through the server, receive the other client's information.
•Use UDP hole punching to establish connectivity with the given IP and port number.
•Listening thread:
        Create new UDP socket.
        Listen on the source port.
        While the port is listening, wait to receive information and print.
•Send data by creating new socket and binding to the destination port.
•Wait for user to press ENTER before sending message to the peer.
```



E: Networking/Communication Model: Example: Client to Server or Peer to Peer (Any Lower layer (Application/Network)

Peer-to-peer networks involve two or more computers pooling individual resources such as disk drives, DVD players and printers. These shared resources are available to every computer in the network. Each computer acts as both the client and the server, communicating directly with the other computers. On a peer-to-peer network, for example, a printer on one computer can be used by any other computer on the network.

A client-server network involves multiple clients, or workstations, connecting to at least one central server. Most data and applications are installed on the server. When clients need access to these resources, they access them from the server. Servers often have private user directories as well as multiple public directories. Client-server networks tend to have faster access speeds because of the large number of clients they are designed to support. The clients are allowed to function as workstations without sharing any resources.

In our project we are using the Client and server networking model to connect through TCP. Further to make an interaction between the clients from peer to peer UDP is being used.

F: Other Considerations (Security, SW Engineering etc)

Using the peer-to-peer communication model, any device could potentially register with the trackerand obtain file chunks from the AUV's already participating in the network. For this particular use case, if a group of AUV's were to be used for military applications, it would be a security risk if an unauthorized device were to connect and obtain confidential data.

To add a layer of security to this network, each AUV will have a recognizable key. When an AUV tries to register with the tracker and gain access to the network it will be asked to provide this key. Any unauthorized device that cannot provide a key or provides the wrong key will not be given the addresses on each peer in the network and will therefore not be able to participate in the filedistribution.

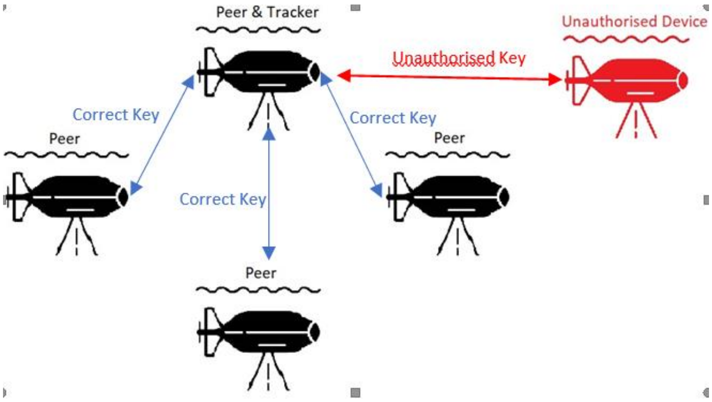


Fig.3 Unauthorised Device Attempting to Connect to Network

G: Implementation Details

Two clients are sending data to each other directly through the UDP protocol.

At the initial stage the client is using TCP to connect to the server and the clients are communicating through the UDP protocol.

Server Code:

At first a socket is build and it waits for connections and when the new connection is found it is added to the clients list. When the list has 2 clients on the list then server send s information to both of them.

Client Code:

First step is connecting to the server through the IP address and wait for it to send data .

The Server establishes a connection by sending information to both the clients for them to have a peer-to-peer connection.

H: Highlights/Unique selling points of the system/Implementation

I am using a Client Server Networking model to make the client and the server interact.It has a lot of advantages to be used.

1) Centralization: In Peer to Peer, there is no central administration, but in client server network architecture there is a centralized control. Servers help in administering the whole set-up and also accessing rights and allocating resource is done by Servers.

2) Proper Management: Since all the files are stored at the same place management of files becomes easy making it easier to find files.

3) Back-up and Recovery possible: Making a back-up of all the data is easy as the data is stored on server. Suppose there's some break-down and data is lost, it can be recovered easily and efficiently. While in peer computing we have to take back-up at every workstation.

4) Upgradation and Scalability in Client-server set-up : If you want to make changes you will need to simply upgrade the server. Additionally, you can add new resources and systems by making necessary changes in server.

5) Accessibility: From various platforms in the network, server can be accessed remotely.

6) As new information is uploaded in database, each workstation need not have its own storage capacities increased (as may be the case in peer-to-peer systems). All the changes are made only in central computer on which server database exists.

7) **Security:** Rules defining security and access rights can be defined at the time of set-up of server.

Further I am using a peer-to-peer based architecture to make the clients interact.

1) The main advantage of peer-to-peer network is that it is easier to set up

2) The peer-to-peer network is less expensive.

3) In peer-to-peer networks all nodes are act as server as well as client therefore no need of dedicated server.

4) It is easier to set up and use this means that you can spend less time in the configuration and implementation of peer-to-peer network.

5) It is not required for the peer-to-peer network to use the dedicated server computer. Any computer on the network can function as both a network server and a user workstation.

I: Changes Advances/or Improvements from project 1 and why?

In Project 1 only the description of the project was given about how AUV's work and the algorithm that could be used to make them interact.

Now in this Project 2 we have implemented the Algorithm of Client server network and the Peer-to-peer network. Project 2 is implemented on vs code and the terminal and outputs can really be seen. Project 2 is the full implementation as described in project 1.

References:

1) https://www.google.com/search?q=socket+proegramming+&tbm=isch&ved=2ahUKEwjr9taim8P2AhUR57sIHSesDxgQ2-cCegQIABAA&oq=socket+proegramming+&gs_lcp=CgNpbWcQAzIHCCMQ7wMQJ1CFDViFDWDwDmgAcAB4AIABPIgBbpIBATKYAQCgAQGqAQtnd3Mtd2l6LWltZ8ABAQ&sclient=img&ei=QvQtYuvNBpHO7_UPp9i-wAE&bih=722&biw=1536&rlz=1C1CHWL_enIE972IE973#imgrc=8FeBlFMvHxg89M

2) computernotes.com/computernetworkingnotes/network-technologies/what-are-benefits-of-the-clientserver-model

3) https://realpython.com/python-sockets