3D3 – Computer Networks: Project 3

Student Name: Jamie Columb **Student ID:** 19335427

Student Name: Shruti Kathuria **Student ID:** 21355061

Student Name: Aarya Sharma **Student ID:**21352973

Aspects Used from Project 2:

Jamie Columb and Shruti Kathuria’s project 2 focused on a group of Automated Underwater Vehicles (AUV’s) that would initially register with a tracker server using a client-server interaction. The tracker server would then exchange the client’s IP and port numbers with each other so they could communicate directly in a peer-to-peer network. UDP socket programming and hole-punching was used in the peer-to-peer interactions. This was proven to be effective and easiest to understand so we have decided to use this feature of project 2 in this project.

Aarya Sharma’s project 2 focused on a swarm of aerial drones connected in a peer-to-peer network to share environmental data and coordinate flight paths. This project made use of separate functions to handle every aspect of data sharing. The project had functions dedicated to handling inbound and outbound connections, disconnections, client messages and requests to stop sending data. The project also made use of unit tests to validate the functionality of the code. We will make use of both of these features while implementing out solution for our chosen use case .

Specific Use Case:

The specific global concern we have chosen to focus on is climate change, more specifically climate related wildfires and air quality surveillance. Wildfires are an unfortunate symptom of climate change or manmade activity, and one which has devastating consequences when not monitored. Our proposed solution to this concern is a network of drones monitoring and in constant communication with one another, surveying the quality of the air/pollution and wildfire in suspected areas. Since the scale of such surveillance is considerably large, one swarm of drones will connect to a server using a client-server architecture, this network of drones will then be able to send data to any other networks of drones created, thus allowing the network of communication to be scaled up. Another aspect of this solution that will be dealt with in this project is the sharing of meaningful data that will allow these drones to coordinate. If the drones send constant updates to each other informing them on their current coordinates and air quality measurements, any nearby drones will know if another drone is detecting a wildfire and will be able to fly to its location to help. Our goal is to create a decentralised system containing network of routers which enable continuous and real-time analysis on growing data assets to manage drone safety

Networking Aspects of our Solution:

- This solution uses a combination of client-server interactions and peer-to -peer interactions.
- Initially, five clients will connect and register with a server, creating a single network.
- This network will continuously send out data from the clients until another network is created which will receive the data.
- The client and server code uses UDP to send and receive data.
- We have also implemented multicast for group communication between specific router and drones to create a way for effective and real-life approach towards data transmission and Real-Time Data like coordinates ,battery level, status, data transfer, port connection are transferred using broadcast
- Similarly, unicast is used for a one-to-one data transmission between drones to transfer information like nearest route(dijkstra’s algorithm),fire alert,collision prediction weather and object alert(for example a tree is in the path).

Requisite Knowledge Necessary for our Solution:

We needed to create a secure network which we achieved by using utf-8,SHA256 hash function and PBKDF2HMAC encrypting data transmission..Python subsets like socket(peer to peer) and threading where also necessary to implement different nodes representing routers parallelly and concurrently. One important aspect of our project is drone to drone data transfer and creating a data lake containing thermographic imaging of the domain, weather information, coordinates, gyroscope , api ,accelerometer ,object alert, path to analyse and inspect the data to create collision prediction and pollution chart to alert other drone in future and take a real world approach. Other than that we needed to implement a fail-safe in case a drone disconnects due to collision or issue with the network. In that case the router/drone will be killed(disconnected) after 60 seconds. After 3 mins of delay the new root node will be initialised. In our project the network shuts down after 10 minutes of activity

Planned Solution

Drone network for fire detection

Hardware required:USB camera,HDMI Monitor/Laptop,Raspberry pi, drone or quadcopter made using motor/controller/power adapter and raspberry pi,temperature sensor(DHT11,DS1B20),Barometer(BMP180),moisture sensor,MQ-2 Gas sensor,PIR motion sensor,Ultrasonic sensor(HC-SR04),Infrared distance meter,maganetic switch,GPS neo module,gyroscope,compass,RTC,bluetooth sensor,SIM

Software:Python 2.7 or above, Py dronekit,OpenCV and Tensor flow,Raspberry Mi module,Py GPS module

We initially deploy two drones(or DIY quadcopter)(or more) as soon as the network is created, we can share route, node and flight data to drone A.The drone after it’s takeoff share weather,air pressure, barometer,distance etc to the router and to the other drone.After calculating it’s nearest/safest route using algorithm like travelling salesman or Dijkstra it proceeds towards the domain area.Using infrared distance meter and gps, drone shares it’s coordinates to it’s nearby nodes(drone B in this case).The drone uses OpenCV and tensor flow to detect objects and people in it’s way to analyse the data to predict future collision as well as transfer the data to drone B to alert it if there’s an object in it’s path like a tree or a building.Once it reaches the domain area it detects the api and transfer the data to nearby node.In case the sensor detects fire it broadcast alert in the system with the coordinates and send thermal imaging which can be used to safeguard wildlife and people.The drones can be used to calculate extent of fire,direction and even to transport goods to places where humans can’t.In case a drone is disconnected an alert with it’s coordinate is broadcasted when this happens the host(us) can takeover manually and guide it back.

Code Layout:

```
|_drone.sh

|_router.py

|_server.py

|_file_transfer

    |_client.py

    |_heat.mp4

    |_security.py

    |_server.py

    |_cn

       |_<empty>
```

Drone.sh

- Run a p2p network of 5 drone
- in case the drone is disconnected kill connection
- After 10 minutes stop the p2p network

Router.py

- connect to node port 3308
- connect to host

```
class RouterClass():
```

```
def __init__(self, host, port):

def getData(self):

def main():
```

Server.py

- Initalize diffrent drones and routers to transmit information

```
class drone:

   def __init__(self, host, port, hostname, p2p):

        def error(self):

        def recieve (self):

        def recieve_data(self):

         def sender(self):

         def process_data(self):

         def check_if_fire(self, data)

         def drone_network(self):

         def start(self, droneName):

          Def main():
```

TEST CASES:

We have tried implement the system based on three aspects of functionality,volatility and scalability.

To test the scalability we implemented two networks with 6 nodes each parallelly sending information to each other, creating a network of 12 nodes however our current function is limited to that as adding more nodes will create an error. This number still can be customized by adding a few components .

To test the volatility we tried sharing larger files and creating more nodes. Even though file transfer with a larger size was still processed, there was incomplete data transfer while using two different remote servers (McNeil server,rasp13 and 33). We failed to conclude if the issues occurred to a lack of admin rights or due to an error in data transfer as the code is not able to identify the cause of the error.

To test the functionality we ran the program on different local computers and remote server with different files including thermal imaging mp4,CSV, and text and it worked successfully and was stable .However, the code is very limited in its functionality which is it’s biggest drawback

Distributed responsibilities of the group:

SERIAL NO.	NAME	CONTRIBUTIONS
1.	JAMIE COLUMB	From the very initial state gave an idea about how to improve the project 2 to project 3 relating it to a global concern.Worked with the presentation.Worked with the code and the report too.
2.	AARYA SHARMA	Worked on the code part and debugged the code well.Also connected to the macneil server.Worked on the report and explained everything in detail.
3.	SHRUTI KATHURIA	Figured how to connect to the macneil server and worked with the debugging of the code. Also worked with the presentation. Worked on the report too.

The potential importance of the solution:

Rapidly spreading fires enormously affect our current climate. They are answerable for up to 20% of yearly worldwide CO_2 outflows and imperil the existence of people, yet additionally of creatures, and consequently, devastatingly affect biodiversity.

With regards to out-of-control fires, there isn't a moment to spare. An answer is required that identifies fires ahead of schedule by identifying smoke, and hydrogen.

By Integrating computer networks with climate concerns, we have devised a solution to surveying and mitigating climate change-related wildfires.

We used drones to identify and detect fire, which indeed will help us to decrease the pollution and help to prevent massive CO_2 emissions caused by wildfires, and protect the environment, wildlife, and communities from devastation, while dramatically reducing the financial impact.
By having swarms of five drones in constant communication with each other using p2p network, air quality can be monitored in a coordinated fashion, mitigating the damage caused by these fires.

Individual peer learning and experience:

SHRUTI KATHURIA:

From the very first individual project to the third group project, I learned how to use the client-server and the peer-to-peer architecture in the projects. I discussed and collaborated well throughout the project. As a team, we expressed our ideas and implemented the project well. These projects helped me grow my knowledge in the computer networking aspects. I learned a lot of things like the python language, about drones, AUVs, underwater communication, raspberry pi etc. With the coming generations and the increasing technology this module made it easier for me to understand concepts.

JAMIE COLUMB:

Working on this project has given me more experience with developing code as a team. From using Gitlab and regularly meeting with the project group, it has helped me improve my teamwork skills and task management skills. There were many coding and networking aspects of this project that were completely new to me which I now have a solid grasp of. The Computer Networks projects were my first experience with socket programming, and from project 3 alone I have learned networking aspects such as multicast and unicast.

Aarya Sharma:

While working on this project I came across a lot of new information like unicast, multicast, and broadcast. Using SHA256 and encryption for the security part was definitely new to me and understanding each concept was well worth it in the end. Our goal was to create a decentralised system with real-world implementation which led us to do a lot of research and understand computer networking in more depth. Working on the project with a diverse implementation range was quite challenging, however, as a team, we tackled all the issues together and came through.