"Air-Talk": A TEMPORARY COMMUNICATION SYSTEM BETWEEN AIR VEHICALS

Anuradha Jayakody[1]
Department of Information Systems Engineering, Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka

Rajitha De Silva[2]
Department of Information Systems Engineering, Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka

Pirabanjan Kirupaharan[3]
Department of Information Systems Engineering, Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka

Sanka Tillakarathna[4]
Department of Information Systems Engineering, Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka

Chapa Vidanaarachchi[5]
Department of Information Systems Engineering, Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka

Sharon Shashangan Jayakanthan[6]
Department of Information Systems Engineering, Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka

---***---

Abstract: Data communication between air vehicles is essential for the efficient use of aviation technology for future potential applications. Communication between air vehicles; maintaining flight safety, useful in disaster situations, for agricultural purposes and, above all, for aviation communication, plays an important role in the management of missions and emergency events. Three methods of data communication between aviation vehicles are currently available. They are: the Air to Ground Communication System (ATG), the Ku-Band System and the Ka-Band System [1]. The ATG system needs land base cell towers and satellite connectivity is needed for Ku-band and Ka-band communication systems. In order to maintain contact, those three structures need an intermediary party. The 'Air-Talk' project proposes an airborne communication device for inter-air vehicle communication without the assistance of a third party intermediary. At present, to the best of the authors' knowledge, there is no direct contact mechanism between air vehicles without the assistance of a ground station or satellite. The project is mainly divided into four components. They include the establishment of a secure private communication network for two or more flying air vehicles and a Technological Infrastructure for Communication Network between Two Air Vehicles and Land Control Devices, as well as a secure platform for communication between air vehicles. This secure platform should have higher fault tolerance and availability, be less vulnerable to attacks and be robust and resilient to possible errors and attacks, encode video data for QoS communication, build an effective compression mechanism to transmit low latency and high quality videos, and implement an algorithm center in such a way that if a single algorithm fails the redundancy.

Keywords: Air Vehicles, Blockchain, Decentralized networking, Video compression

I. INTRODUCTION

The Air-Talk research implements a secure, decentralized communication system for the communication between air vehicles without the assistance of an intermediary third party. This system consists of a decentralized file sharing and chat framework where all air vehicles can connect to the network and share the necessary data securely with any connected node. This network allows video feeds of flights, position of aircraft, location of emergency situations, points of interest, geo-referenced drawings and other GIS (Geographic Information Systems) information to be transmitted. This safe data exchange framework leverages the advantages of trending decentralized technologies such as Blockchain and IPFS (Interplanetary File System). A Blockchain is implemented to provide confidence and accountability, removing the need for a third party to build a trust-based model. IPFS, which is a peer-to - peer architecture, has been adopted for this platform to overcome the storage and processing limitations of network nodes. There is also no chance of a single point of failure. On this platform, nodes connected to the private IPFS network are enabled to share information and data within the network. If the data is submitted to the network, it is stored in the IPFS network and a unique hash is returned for each file upload. This hash value is then stored within a private Blockchain providing trust, transparency and immutability. In addition, this framework provides a chat application for real-time communication between the nodes participating in the network. This was also developed using the features offered by IPFS. Decentralized storage, Ethereum Blockchain and encryption are integrated in this project.

This research paper would address one of the key factors in selecting the best and shortest route in less time to boost the quality of services on the network. In this way, this research adds the A-star algorithm to the Software Defined Network and another algorithm that we can easily implement to the Software Defined Network. This research also addresses parallelizing the A-Star algorithm as well as running another algorithm parallel to the A-Star get the shortest path from the best output. Currently, most of the communication systems are based on the standard client-server model. Due to the conventional client-server model, most communication systems frequently suffer from service inaccessibility during server downtime or when servers face single point failure or DDoS attacks [2]. In order to concentrate on contact between air vehicles, unrestricted communication and exchange of knowledge is essential. Communication between air vehicles; maintaining flight safety; useful in the event of disasters; agricultural purposes; and, most importantly, aviation communication, play a key role in the management of missions and emergency situations. Three methods are currently available for data communication between air vehicles. They are ATG (Air to Ground Communication) system, Ku-Band and Ka-Band [3]. The ATG system requires ground-based cell towers, and ku-band and ka-band network systems require satellite communication. These three systems require an intermediary party to maintain the communication. A decentralized communication framework can make these specifications simpler than the traditional centralized server-based model.

The rest of the paper is structured in the following manner – A brief background about the technologies used in this research is discussed in section two. The next section is the methodology where the system overview and the overall system design is explained. Section four discusses the testing and results. The paper concluded with a summary and discussion of the current study.

II. BACKGROUND

This section discusses the technologies and concepts that are used to build "Air-Talk". This contains a brief introduction to build the shortest path in the SDN network which uses A-Star algorithm, Decentralized networking, IPFS, Blockchain video encoding and Decoding.

Traditional network devices like switches and routers do the routing decision in the device itself, SDN separates these components into different planes like data plane and control plane. The research conducted by Jehn-Ruey Jiang,Hsin-Wen Huang, Ji-Hau Liao and Szu-Yuan Chen to find the shortest path uses enhanced Dijkstra algorithm to find the shortest path which uses end to end latency that outperforms the original Dijkstra algorithm. Also, the research conducted by Arnav Shivendu, Dependra Dhakal and Diwas Sharma to find the shortest path in the network uses the bellman-ford algorithm and POX API's [4]. This research paper considers using A-Star algorithm instead of using bellman-ford algorithm, and separates the search algorithm to cloud environment for a much faster and efficient way to calculate the complex network routing path. Also, this research enhances the scope to check different shortest path algorithms and implement it in the application.

A. Decentralized Networking

In simple terms decentralized network is an architecture where the workloads are distributed among several machines instead of relying on a single central server. A decentralized network has a vast range of advantages when compared with a conventional centralized network. Some of those key advantages are increased system reliability, scalability and privacy [5]. Decentralized network has an increased reliability when compared with a centralized network because there is no real single point of failure as the nodes in the network are not reliant on a single central server to handle all the requests. Also, these types of networks are much easier to scale since all what is required is to add more nodes to the network to gain more compute power. A decentralized network architecture provides greater privacy, as the information is not passing through a single point and instead passes through a number of different points. This makes it is difficult to track across a network thus providing more privacy for the information being transferred inside the network.

B. InterPlanetry File System (IPFS)

The InterPlanetary file system (IPFS) is a protocol as well as a peer to peer network for sharing and storing hypermedia in a distributed file system. It is a content addressed file system which uniquely identify each file in a global namespace using content addressing. Main concepts used in IPFS are Distributed Hash Table (DHT); which is used to retrieve data across nodes in the network, Block Exchange; a peer to peer file sharing protocol which coordinates data exchange between untrusted swarms, and Markle DAG; uses a Merkle Tree or a Merkle DAG similar to the one used in the Git Version Control system[6]. It is used to track change to files on the network in a distributed friendly way [7].

There are two types of IPFS networks; Public and Private. All the information inside a public network is accessible to everyone. But most applications and solutions require control over the data transferred inside a network. Therefore, they require more privacy. IPFS private networks help to close the network for certain entities while providing all the features given in a normal public IPFS network [8]. The described project in this paper also uses a private IPFS network to provide more privacy for the data transferred inside the network while reaping most form the features provided by an IPFS network.

C. Blockchain

A blockchain is essentially a fault-tolerant distributed database of records of a public ledger of all transactions that have been executed and shared among participating parties [9]. A consensus of a majority of participants in the system verifies each transaction in the public ledger. With the combination of cryptography and consensus, blockchain provides a tool for increasing data integrity. There is no central server point of control. The information is stored in blocks and each block contains a transaction, a time stamp and a link to the previous block as shown below.

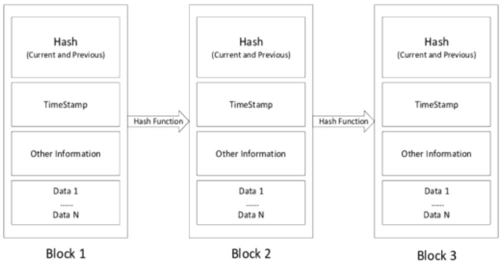

Fig.1. Blockchain Structure.

All blockchain structures fall in to three categories; Public Blockchain architecture, Private Blockchain architecture and consortium Blockchain architecture [9]. From those three categories authors have used private blockchain architecture since those type of blockchains are controlled only by users from a specific organization or authorized users who have an invitation for participating.

D. Video Encoder

The method of compressing and actually modifying the format of video content is video encoding, sometimes even converting an analog source to digital one. In terms of compression, the target is for less volume to be used. This is because it's a lossy process which turns down video-related content. A rough estimate of the original is generated upon retrieval decompression. Therefore, more encoding applied, the more content can be tossed out, and the worse the comparison of the original appears.

E. Video Decoder

A video decoder is an electronic circuit, usually found inside a sole integrated circuit chip, which transforms base-band audio input digital signals video. Video decoders typically provide configurable modulation of video features such as brightness, contrast, and saturation. A video decoder implements a video encoder's inverse function, which transforms uncompressed digital video to analog data. Video decoders are widely found in frame grabbers and video recording apps.

III. METHODOLOGY

This section discusses all the architectural components, design components and methodology of the project. The aim of the project "Air-Talk" is to create a temporary communication system between air vehicles and create a file sharing application and a chat application which can be used without the aid of an intermediary third party using the above described technologies in the Background section.

A. What is a star algorithm?

A-Star algorithms achieve optimality and completeness, which are two valuable properties for a search algorithm. When there is an optimality the best possible solution is guaranteed. When there is a completeness if there is a solution then the algorithm is guaranteed to find it [10]. This algorithm will be used to get the shortest path in the centralized routing table with minimum time compared to the other existing algorithm.

B. What is parallel computing?

It refers to breaking the process into small parts which are independent of each other, often sharing the shared memory for the process to communicate with each other, the results of the combined process will be the final output of the algorithm. The primary goal of parallel computing is to increase the available computation power for faster application processing and problem solving. There had been few researches conducted to fasten the algorithm by introducing it to the parallel computing but introducing the parallel computing in an network is not implemented as the router needs be the with high CPU and RAM. In order to overcome that it has been implemented in the Cloud using an API call. Since in an SDN network all the routing is controlled in the central controller by connecting the controller with the cloud infrastructure to get the outcomes for the complex network will be much easier.

C. Implementation SDN simulation

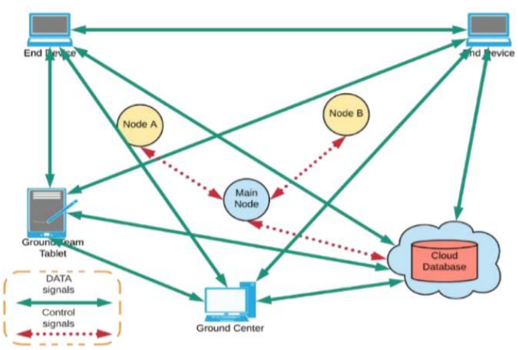

Fig. 2. System overview diagram for SDN implementation

As shown in the Fig. 2 control network updates the routing table of each node to the algorithm hub and requests for the routing path when data network requests for the routing path. In the initial step each node will send the ping packet to calculate the estimated cost for each goal node which will be used by the algorithm in the later part. When there is a change in the routing table algorithm hub is also updated through the central network.

Step 01: Control network collects all the routing table and cost (bandwidth, ping time interval between source and destination) from the data network.

Step 02: Routing table of each node and the cost with ping value to each source and destination is stored and sent to algorithm hub.

Step 03: data network request for the routing path from source and destination from the central network.

Step 04: central network request for the shortest routing path from the algorithm hub.

Step 05: AWS lambda function reads the table and shortest path is sent to the central network.

Step 06: data network receives the shortest path from the central network.

Phase 1: Creating A-Star algorithm

The initial step of the research was creating an A-Star algorithm to find the shortest path instead of using Dijkstra or DUAL algorithm which is more beneficial. For the calculation path central network will provide the cost with a routing table of each node and as an initial network boot it will send the ping to all the router to calculate the cost for each destined node this value will be taken as the heuristic value from the source to the destined node.

$$f(n) = g(n) + h(n)$$

f(n)- total cost heuristic value and the actual cost

g(n)- cost value taken from the routing table of each neighboring node before sending the actual data.

h(n)- cost value taken from the network start by sending a ping command to each node. This heuristic value will be also updated with the old actual value in the routing table when there is a new routing path request.

Phase 2: Creating Algorithms Hub on top of simulation

This part of the research is to enhance the concept of centralized algorithm hub in order to get the shortest path from the routing table faster and efficient as new algorithms are introduced to the industry we will have to create new routing protocol or change the existing protocol to introduce the new path finding algorithm such as A-Star and IDA*. From this algorithm hub we can directly implement the algorithm to the hub and check the results immediately as the central network used the routing path given by the algorithm hub

Phase 3: Creating Algorithm Hub in parallel computing

As shown in the implementation diagram all the algorithms will be parallelized using lambda services so each algorithms can find the best path could be selected with less time, other than that A-Star algorithm will be implemented using thread in parallelized manner so the calculation of the node is simultaneous.

D. Simulation Test Scenario

Basic network was designed on the Mini-net network simulation environment as in the figure below

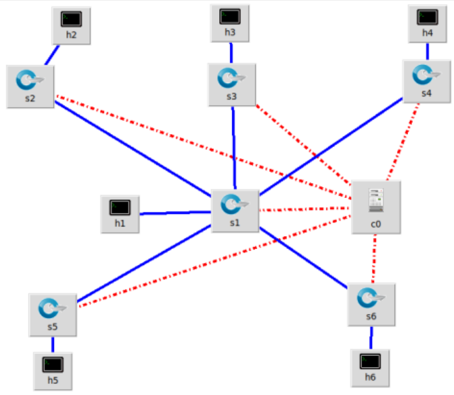

Fig. 3. Mini Net Simulation Network Diagram

This Network includes 6 Dummy SDN routers with 6 host machines. All 6 routers are connected as above figure and SDN network Controller is connected to each router. Pox controller is the controller used as the SDN network controller. That SDN Network Controller and the A-star algorithm configured Path selecting machine learning server are connected together.

Communication between Controller and Path selecting machine learning server is established with API calls that Controller requests for a path or a route from path selecting server and selected path with other routes will be communicated back by the machine learning server.

During the Request, Source router and destination router with their each communication cost value is sent and as the reply that single path that should forward the packet along with intermediary routes that needed to configure in the middle route will be sent.

E. IPFS Technology

IPFS technology has enabled a private distributed P2P network between the nodes participating in the network without an intermediary third party. The file sharing platform is built on top of this private IPFS network.

Therefore, the information transferred inside the network will only be visible for the other peers who have a shared secret key. Inside a private IPFS network, each node specifies which other node it will connect to. Nodes in a private IPFS network do not respond to communications from nodes outside that network [8]. Thus, it provides the greatest level of privacy from the outside world. In addition to this to gain more security for the information being transferred inside this network, the hash received for each information upload inside this network will be saved on a private blockchain. This guarantee that the data cannot be altered.

F. H.264

H.264 is a highly common video compression technique that is widely used with the common and well-known format of .mp4. In this project, we implement a fully working encoder and decoder for the H.264 technique. In order to explore our results, we compare different compression rates by choosing how many reference frames to choose which shall give different video qualities that degenerate at high compression rates. Through the paper, we use a block size of 16 x 16 for the reference frames to carry out the motion estimation and compensation (predicting frames). Although the code is modular and can work with any given block size, we have chosen 16 x 16 to be our test case.

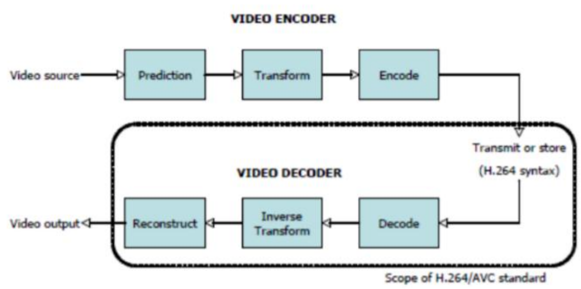

Fig. 4. H.264 video coding and decoding Process [16].

G. System Design

Before creating decentralized applications, a private IPFS network was set up using three virtual machines with Linux Ubuntu 20.04 installed. In this phase command line was used as the main tool for installing necessary packages and configurations. All three virtual machines were given three static IP addresses during the virtual machine configurations. After installing Go and IPFS on all the nodes participating in the private network, the swarm key generation utility was installed. This swarm key allows the user to create a private network and let the peers to communicate only with those peers who share this secret key [8]. "Air-Talk" applications will run on top of this private IPFS network.

The decentralized file sharing application of this project is built mainly using Blockchain and IPFS technologies. The backend of this application is based on a Blockchain protocol called Ethereum [11]. Ethereum uses smart contracts to handle the transactions. Smart contracts are written using Solidity programming language. React is used to write the client-side application of this project. Shown below in Fig. 5 is the System Architecture diagram of the decentralized file sharing application.

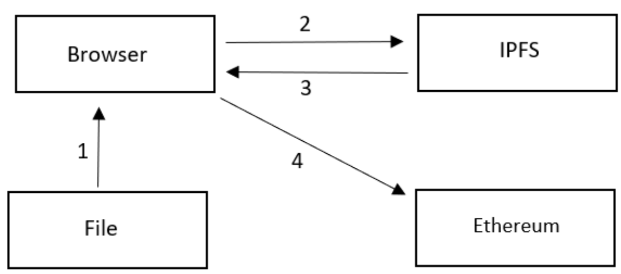

Fig. 5. System Architecture of file sharing application

As shown above in number 1, first the user has to upload a file to the browser. When the user submits the file, the file will be stored in the IPFS (2). Once the file is stored in IPFS, it will provide a unique hash for that file (3). In the step 4, this hash is stored in a blockchain. Then the blockchain will verify the transaction before storing it in the network. To verify the transaction MetaMask [12] will be used. Once the transaction is verified successfully, the hash will be stored in the blockchain. For this the authors have used a private blockchain provided by the Ganache [13] which is a software that allows us to create a personal Ethereum blockchain which can be used to run tests and execute commands. Once the hash is saved in the blockchain no one can edit this. This adds an extra layer of security to the application. Shown below in Fig. 6 is the smart contract written in the blockchain to get the hash from the IPFS and store it in the blockchain. This is written in Solidity programming language. Users with the hash value can view the information uploaded to the IPFS network via a unique URL given to each file inside the network.

```
pragma solidity 0.5.0;
contract Hash {
  //Smart contract is written here
  //take the hash from the IPFS and write it to the blockchain - read function
  //take the hash back from the blockchain and return to the client side - write function
  string filehash;
  //write function
  function set (string memory _filehash) public {
    filehash = _filehash;
  }
  //readfunction
  function get () public view returns (string memory) {
    return filehash;
  }
}
```

Fig. 6. Smart contract used for the blockchain

Decentralized chat application is designed to use for the communication between the nodes participating in the network. It is based on IPFS's publish-subscribe mechanism. It is a standard messaging pattern. In here the publishers do not know whether anyone will subscribe to a given topic. Publishers send messages on a given topic or on a given category and subscribers will only receive messages on a given topic which they have subscribed. In these types of scenarios there is no direct communication between publishers and subscribers. When developing this application, a package called 'ipfs-pubsub-room' which helps to simplify the interaction with the IPFS pubsub facilities was installed. It allows developers to easily create a room based on an IPFS pubsub channel, and then it emits membership events, listens for messages, broadcasts and direct messages to peers [14]. This allows a fast and scalable dynamic communication between peers. The software design pattern used in designing this is the Model-View-viewmodel (MVVM) design pattern. It is a software architectural pattern which facilitates the separation of the development of the GUI (Graphical User Interface) from the development of the business logic or from the back-end logic [15]. Therefore, the view is not dependent on any specific model pattern.

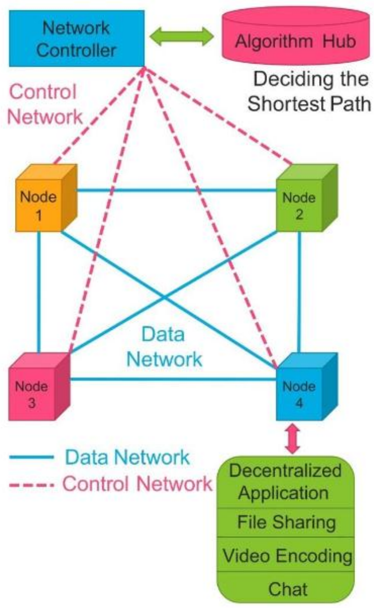

Fig. 7. System Overview

Fig. 7 explain the overall system overview of this research paper.

IV. Results and Discussion

This section contains the testing and results of the project "Air-Talk".

A. *Network Results*

Testing of the network is done with main 3 scenarios and all 3 scenarios were passed with Basic QA testing.

Scenario 1: Where all the Nodes are connected to each other. In this setup, there are 6 end nodes with 6 SDN switches which are connected as a star topology. All the end nodes are connected to each switch and all the switches are connected to a controller.

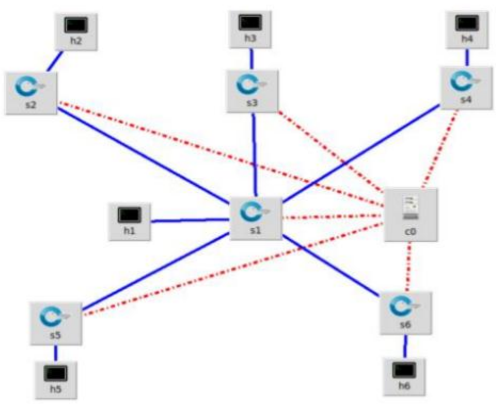

Fig. 8. Scenario 1 Network Diagram

All the network nodes were able to communicate with every other node. Results were obtained from Mininet Simulate software and below figure shows the results that all are connected. Packet transferring rate is 100% and none of the packet is dropped. Figure shows that each end node sends a connectivity checking ping command to the other each end host.

```
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6
h2 -> h1 h3 h4 h5 h6
h3 -> h1 h2 h4 h5 h6
h4 -> h1 h2 h3 h5 h6
h5 -> h1 h2 h3 h4 h6
h6 -> h1 h2 h3 h4 h5
*** Results: 0% dropped (30/30 received)
```

Fig. 9. Scenario 1 Results

Scenario 2: Where partial of the network is with a single point of failure. In this setup, there are 6 end nodes with 6 SDN switches. All the switches are connected as bus topology where S2->S1 connection and S3->S1 connection are dropped. All the end nodes are connected to each switch and all the switches are connected to a controller.

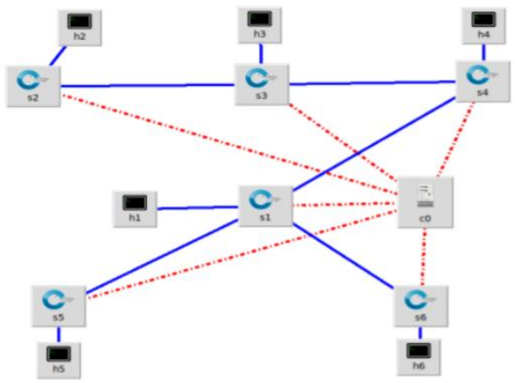

Fig. 10.Scenario 2 Network Diagram

All the nodes were able to communicate with each other every node. Packet transferring rate is 100% and none of the packet is dropped. Figure shows that each end node sends a connectivity checking ping command to the other each end host. S1 switch and the S1 -> S4 connection becomes a single point of a failure in the network.

```
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6
h2 -> h1 h3 h4 h5 h6
h3 -> h1 h2 h4 h5 h6
h4 -> h1 h2 h3 h5 h6
h5 -> h1 h2 h3 h4 h6
h6 -> h1 h2 h3 h4 h5
*** Results: 0% dropped (30/30 received)
```

Fig. 11.Scenario 2 Results

Scenario 3: In this setup, there are 6 end nodes with 6 SDN switches. All the end nodes are connected to each switch and all the switches are connected to a controller. Where previous scenario 2 single point of failure occurred and created 2 networks. Connection between S1->S4 is drooped.

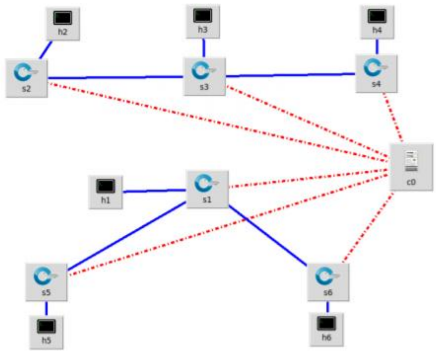

Fig. 12.Scenario 3 Network Diagram

In this scenario half of the network were able to communicate with each other separately. Packet transferring rate is 60% and 40% of the packets are dropped. The upper network which consist of S2, S3, S4 switches can communicate themself but can't communicate with other switches and vice versa.

```
*** Ping: testing ping reachability
h1 -> X X X h5 h6
h2 -> X h3 h4 X X
h3 -> X h2 h4 X X
h4 -> X h2 h3 X X
h5 -> h1 X X X h6
h6 -> h1 X X X h5
*** Results: 60% dropped (12/30 received)
```

Fig. 13.Scenario 3 Results

Below show the micro services that were implemented to check the connectivity of nodes output for Scenario 3. Below figure 10 consist of 4 linux terminal outputs where "Node:h1" and "Node h5" terminals from bottom network and "Node:h2" and "Node:h3" terminals are from the upper network. In every terminal the first line shows its own connectivity and the rest of the lines show the all end nodes with its IP.

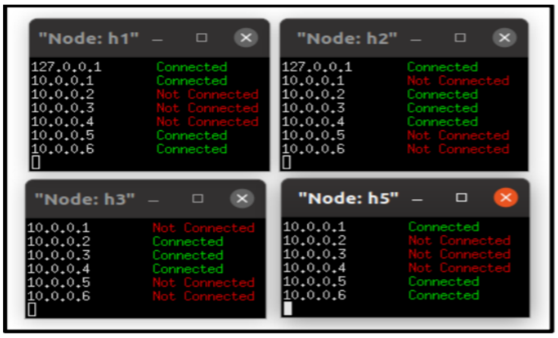

Fig. 14. Microservices Results

Further scope we can develop this research by adding new algorithms to the algorithm hub which we run parallelly to get the shortest path from the routing table. Also measure the efficiency of each routing path at the given point of time and choose the path given by each algorithm to decide the best out of all. In the above A-Star algorithm it was implemented in a single by using thread to calculate the cost of each node but we can run the bi-directional A-Star algorithm in a parallel environment which will intersect at one point and take both outputs as a single routing path from source to the destination.

B. Algorithm hub Results

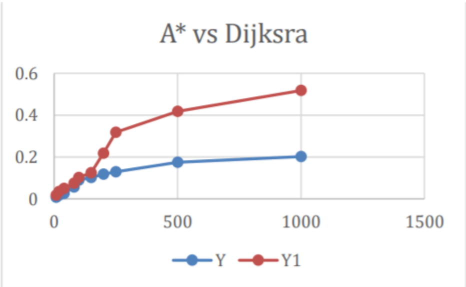

Fig. 15. Time and Node comparison graph for both algorithms.

Algorithm efficiency is measured from the starting of the API request form the controller to the output given by the application. Below graph clearly explains A-star algorithm (Y1) has outperformed the Dijkstra algorithm (Y) when the number of the nodes are increased. In the figure 15, two time were taken to calculate the path from same node to same destination as explained in the methodology. First network will take the heuristic value as the ping time taken from the source to the destination and its represented in the graph as T1 and which take less time to calculate the path but later heuristic value is changed as data has sent real data from all the nodes it can get the real time for each packet transmission and that value is taken as heuristic value which has increase the time represented in the figure 16 as T2. Though the time of the calculation has increased the result given by the algorithm (routing path) is much more efficient with less traffic route utilizing all the nodes resources.

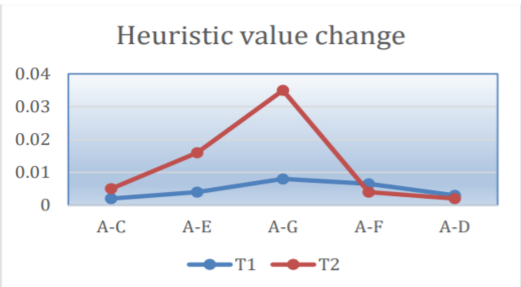

Fig. 16. Path calculation time, after changing the heuristic value.

C. Blockchain Results

In decentralized filesharing application, a private blockchain from Ganache was used to store the hash generated from IPFS. Shown below in Table 1 is the private blockchain used.

TABLE I. Private Blockchain from Ganache

Blockchain Address	Balance	TX count	Index
0x3d94f3c0Fa3FeB4c0D34DA5886bDe6a76470348a	99.92 ETH	32	0
0x39146A5ABad53883a6cBC2B61780A56bEA8B5a17	100.00 ETH	0	1
0xeB4d1a5a78cAda9eB955D09678F2AbeCB49AD69E	100.00 ETH	0	2
0xAD5eE372366b558922a7426e1BeF91EE2B55F2CF	100.00 ETH	0	3
0x8616e8a736E71230363Afac27c9a66cC9Ef4f769	100.00 ETH	0	4
0x5b41D576863fA00340aD111f2a339Acbd9C7A131	100.00 ETH	0	5
0xeBC288871dB4350a50DF5ed8D275Cc92bc0B6437	100.00 ETH	0	6
0x9462FE5e15aB342645c46ef3598893111Fb565e3	100.00 ETH	0	7
0xF98e566407F015A2ebfe205c7CA995627B214Ce1	100.00 ETH	0	8
0xc7bAb69175aD8b9A41520f82b0ad87138F536f4E	100.00 ETH	0	9

Shown below in Fig. 17 is a screenshot of the File sharing application with some files with the URLs which are uploaded to the IPFS network.

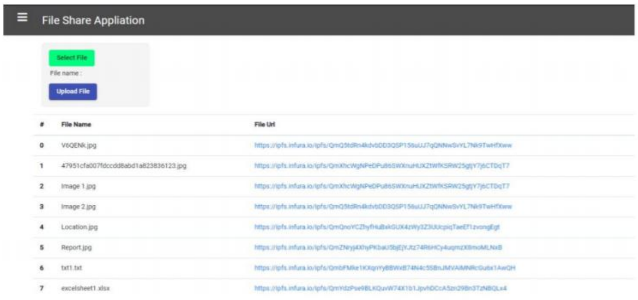

Fig. 17. Interface of the file Sharing application

Shown in fig.18 is a screenshot of the decentralized chat

Application with a sample chat.

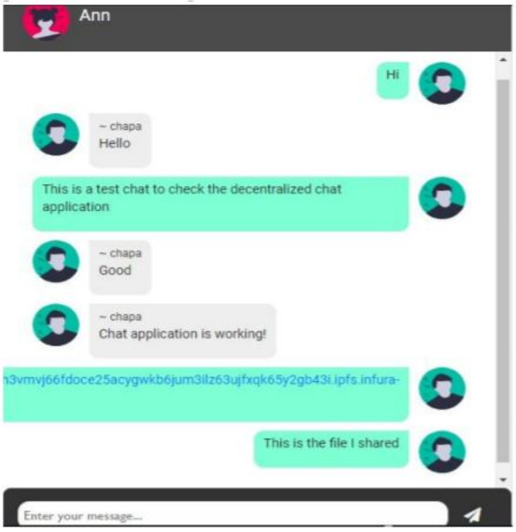

Fig. 18. Interface of the Chat application

D. *Video Encoding and Decoding Results*

The results of video encoding and decoding are obtained from MATLAB simulation work.

Fig. 19. Comparison of original frame and decoded frame

In Fig. 19 shows the comparison of the original tested video frame and the decoded video frame.

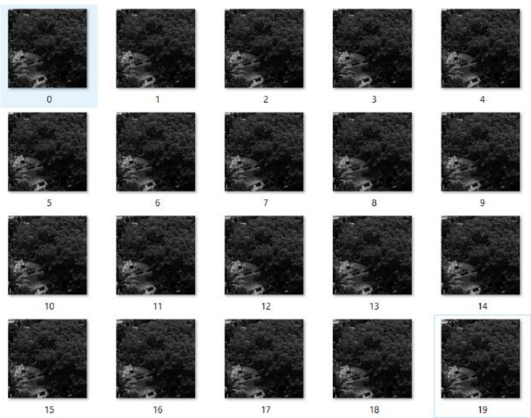

Fig. 20. Encoded Frames

Fig. 20 shows how the encoded video frames were saved in the dedicated folder.

The below Table 2 shows the encoder's configuration parameters.

TABLE 2: Encoder's configuration parameters.

Profile	Main
Number of frames	200
Frame rate	20 fps
Motion estimation	Simplified
RD optimization	High Complexity mde
Symbol mode	CABAC

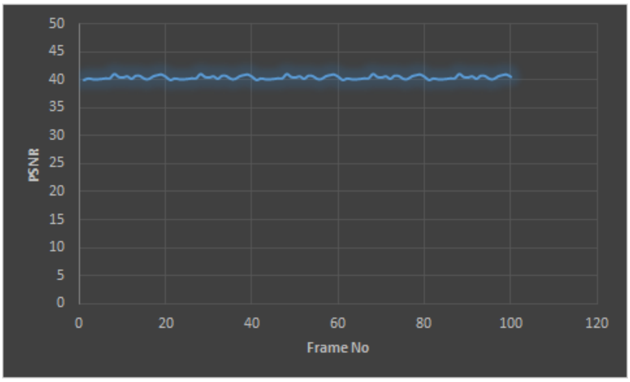

Fig. 21. Peak Signal to noise ratio

Figure 21 shows the PSNR values for the implemented mechanism. If any PSNR value is greater than 35.0, it means that was a good video frame. And if the PSNR value is lesser than 20.0, it means that the video frame was bad.

V. Conclusion

Assessing the progress of the research, we were able to successfully complete the research with the main objective of successfully developing a secure private communication network for two or more air vehicles and a technical communication infrastructure network between two air vehicles and ground control devices. The specific objectives that we intended to accomplish at the end of the research were able to be accomplished as follows. The network that can communicate up to 1.5 km (using 3 hosts) without a single drop of the packet. The distance between 2 nodes can be up to a maximum of 700 m. With no hosts, communication distance can be increased.This paper introduces a secure, decentralized file sharing platform based on IPFS and Blockchain. The main objective of this research is to create a decentralized private network for two or more moving air vehicles to securely share messages and files such as images, videos and documents. At present, most of the communication and information exchange activities between air vehicles are controlled by a centralized authority. This causes data protection and data availability problems. This project is leveraging the best features of the new trending decentralized technologies, such as IPFS and Blockchain, to resolve these issues. Within this platform, data hashes obtained from IPFS for each exchange of information are securely stored in a private Ethereum Blockchain through a smart contract providing data immutability and data source. Also, any malicious changes to the source data can be avoided by using this model.

References

[1] S. Nichols, "L/Ku/Ka-band satellites – what does it all mean?" Get Connected, 2020. [Online]. Available: https://www.getconnected.aero/2017/09/lkuka-bandsatellites-mean/. [Accessed: 22- Feb- 2020].

[2] Sucuri. 2020. What Is A Ddos Attack? Types & Prevention Methods. [online]Available at: <https://sucuri.net/guides/what-is-a-ddosattack/#:~:text=The%20DDoS%20attack%20will%20test, zombie%20devices%20called%20a%20botnet.> [Accessed 25 September 2020].

[3] Tutorialspoint.com. 2020. GSM - Security and Encryption - Tutorialspoint. [online] Available at: <https://www.tutorialspoint.com/gsm/gsm_security.htm> [Accessed 25 September 2020].

[4] "Emulation of Shortest Path Algorithm in Software Defined Networking Environment", Citeseerx.ist.psu.edu, 2020. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.695.4776&rep=rep1&type=pdf#:~:text=The%20main%20goal%20of%20Software,to%20be%20open%20and%20programmable.&text=This%20project%20implements%20the%20Bellman,implement%20the%20Bellman%20%E2%80%93%20Ford%20Algorithm. [Accessed: 01- Sep- 2020].

[5] "Centralized Networks vs Decentralized Networks," Solarwinds MSP, 10-Jun-2019. [Online]. Available: https://www.solarwindsmsp.com/blog/centralized-vs-decentralized-network. [Accessed: 13-Sep-2020].

[6] "Book," Git. [Online]. Available: https://git-scm.com/book/en/v2/Getting-Started-About-Version-Contro. [Accessed: 13-Sep-2020].

[7] "InterPlanetary File System," GeeksforGeeks, 20-Sep-2019. [Online]. Available: https://www.geeksforgeeks.org/interplanetary-file-system/. [Accessed: 13-Sep-2020].

[8] "Building Private IPFS Network with IPFS-Cluster for Data Replication," Eleks Labs, 19-Jun-2020. [Online]. Available: https://labs.eleks.com/2019/03/ipfs-network-data-replication.html. [Accessed: 13-Sep-2020].

[9] "MLSDev", Mlsdev.com, 2020. [Online]. Available: https://mlsdev.com/blog/156-how-to-build-your-own-blockchain-architecture. [Accessed: 28- Feb-2020]

[10] "A-Star (A*) Search Algorithm", Medium, 2020. [Online]. Available: https://towardsdatascience.com/a-star-a-search-algorithm-eb495fb156bb. [Accessed: 02- Jul- 2020].

[11] "Home," ethereum.org. [Online]. Available: https://ethereum.org/en/. [Accessed: 13-Sep-2020].

[12] MetaMask. [Online]. Available: https://metamask.io/. [Accessed: 13-Sep-2020].

[13] Truffle Suite, "Ganache," Truffle Suite. [Online]. Available: https://www.trufflesuite.com/ganache. [Accessed: 13-Sep-2020].

[14] Available: https://github.com/ipfs-shipyard/ipfs-pubsub-room/blob/master/README.md. [Accessed: 13-Sep-2020].

[15] "Model–view–viewmodel," Wikipedia, 31-Aug-2020. [Online]. Available: https://en.wikipedia.org/wiki/Model–view–viewmodel. [Accessed: 13-Sep-2020].

[16] H.264 video coding and decoding Process [online] Available:https://www.vcodex.com/an-overview-of-h264-advanced-v ideo-coding/ [Accessed: 24- Jan-2020].