Reflection on Coursework

Reliability

Reliability is ensured by applying sequence numbers and checksums in the reliability layer. By specifying sequence number for each message and requiring the receiver to remember the segments to detect repetitions and to prevent unnecessary retransmission. The checksum ensures that the packet will never be modified. The sequence number is used to give each packet's order. This layer of the overtime mechanism can be effectively avoiding the loss of data packets. In other words, the sender did not receive the corresponding packet acknowledgment in time and the resend operation will be triggered.

Message ID is a child node of the neighbor loss detection sequence number. Message ID must be routed to the sequence number of the routing information to distinguish between routing information and normal information. For normal messages, the sequence number starts with value 1 and increments. The routing table and neighbor loss detection are updated regularly. This ensures that the left state is kept up to date.

For the sender, when sending a new message, the sequence number is incremented. The expected serial number is set by the receiver.

Data Structure

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Checksum | Length | Source | Destination | Flag | Message | Sequence | Data |

Before packaging the header, the sender uses the string format to save the following message to the list. In order to transmit data packets with other layers, the data packets must be packaged before being sent. Then, the receiver unpacks the packet arriving at the time packet. The format of an unpacked packet is a tuple. Due to the unmodified characters in the tuple format, the format of the tuples must be transferred to the list.

Protocol Improvement

In this program, this internet protocol is currently feasible in full-mesh networks. Since the mapping table of each node in the network is completely shared, this program is also feasible in non-full mesh networks.

Testing

Sending Message

When sending a message to a specific user, entering “/ s USERNAME MESSAGE.” For instance, Node A wants to send the message “hello” to Node B, as shown in Figure………..

Another test situation, Node B sends a message to Node A. Once the message has been sent, on the Node B side, it represents the sender, displaying the header and content of this message. At the same time, in the side of the receiving terminal, node A, only the contents of the message are displayed. The result is shown in Figure……..

Through some testing, it can make sure send multiple messages. The result is shown in Figure…….

File transmission

When transferring files to a specific user, entering” /f USERNAME FILENAME”. For example, Node A wants to send a file which name is “hi.txt” to Node B, as shown in Figure……

‘/r YES|NO used to choose whether to accept or reject files that sent to you from the receiver. If the receiver’s file was accept, the file is informed that it has been successfully received. It also created a .txt file named ‘received\_file’. The validation result is shown in figure……

By comparing the received file with the original file, the result shows that the content of these two is the same, which means that the file transfer is successful. As shown in Figure…….

When the receiver refused the sent files, the terminal indicates to the user that the file has been rejected. It is shown in Figure…..

List Current users

When entering ‘/l’ is used for listing current users. The result is shown in Figure……..

Conclusion

This program is based on the design of the Python language network protocol to achieve. Firstly, we set up a neighbor detection mechanism to detect changes in the network. Secondly, we need to send reliable messages according to our protocol design, and we complete reliable communication with another node. At the same time, we can correctly identify the message type.

After we finished our code, we tried to connect to the other groups, and then we tested whether we implemented all the basic functionality required by the protocol. This protocol can provide two or more users with a reliable way to transfer information and files. It implements a reliable flow control message protocol. The packet can be forwarded correctly. It can implement the datagram forwarding routing protocol. Provide routing table to deal with. It can be choose to provide multicast forwarding and routing services.