My partner and I designed the packet format to be as follows

```
struct {
        int type;
        int bufferLength;
        void * buffer;
}
```

The ‘int type’ will hold either 0 or 1. This is to signify to the receiver if this message is a heartbeat signal or a message to be passed on.

The ‘int bufferLength’ will hold the size of the buffer, so the receiver knows how many bytes to read from the buffer.

the ‘void * buffer’ is the data being sent.

The header is the 2 ints, and the body sent is the buffer.

For the heartbeat messages we designed a way for the client and server to distinguish between this. If the header received has a type of 1, it means the packet is a heartbeat message, and the buffer just contains the ID session and does not need to be passed to the telnet sessions. If the type is a 0, the proxy knows this is actual data that needs to be passed onto the telnet sessions.

If the serverProxy does not receive 3 consecutive heartbeat messages, it will terminate the connection with the clientProxy and begin accepting new connections from clients. When the client reconnects the server will check the ID from the heartbeat, if this is the same as before it will continue the session with telnet like nothing happened. However if the client connects with a new ID, the server will terminate its session with telnet and start a new session.

If the clientProxy does not receive 3 consecutive heartbeat messages, it will close the current server socket and open a new socket and try connecting infinitely.

The main actions between clientProxy and serverProxy have 2 main functions. First, if the packet is being received from a telnet application the proxy application needs to add a header in front of the body. We will set the header so that type represents actual data and not a heartbeat message, and the bufferLength will contain the size of the body being sent.

If the packet is being received from a proxy, the receiving proxy needs to extract the body message from the packet and send only the body, which is just a buffer of bytes, to the telnet session.