(reliable UDP) RUDP – 6

Introduction :

The UDP protocol is widely used for transmitting data over networks because it is simple and efficient. However, it does not provide a transmission reliability mechanism, which can lead to data loss or corruption. To overcome this problem, the RUDP protocol (Reliable User Datagram Protocol) was developed. RUDP is a transport layer protocol that ensures the reliability of data transmission over networks using techniques such as flow control, retransmission of lost packets and acknowledgment of received packets. Although less commonly used than UDP, RUDP is an attractive choice for applications that require reliable data transmission over networks. In this text, we will examine how RUDP works, the differences between RUDP and UDP, the advantages and disadvantages of RUDP, and we will also describe a practical case of using RUDP for communication between a client and a server. Proxy.

How the protocol RUDP work :

RUDP is a transport protocol that provides guarantees of data transmission reliability by using techniques of flow control, retransmission of lost packets and acknowledgment of received packets. The protocol was developed to overcome the limitations of UDP, which does not provide a transmission reliability mechanism. Unlike TCP, which uses a transmission control protocol to ensure reliable data transfers, RUDP is datagram-based and does not involve a prior connection between hosts.

The differences between RUDP and UDP are mainly related to the reliability of data transmission. While UDP is an unreliable transport protocol, RUDP ensures reliable data transmission by using techniques of retransmitting lost packets and acknowledging received packets. Unlike TCP, which uses a flow control system to avoid network congestion, RUDP uses a congestion control mechanism based on a maximum allowed throughput for each connection. This ensures fast and reliable data transmission without overloading the network.

The process of sending and receiving RUDP packets is similar to that of UDP, with some differences related to transmission reliability mechanisms. When a host sends a RUDP packet, it expects an acknowledgment (ACK) from the

destination host. If the ACK is not received within a specified time, the packet is retransmitted. The destination host verifies the integrity of the received packet and sends an ACK back to the originating host to indicate correct receipt of the packet. If a packet is lost during transmission, the destination host can request its retransmission by sending a retransmission request message. The advantages of RUDP lie primarily in its ability to provide reliable data transmission without the time and resource costs of establishing a connection first, as is the case with TCP.

However, RUDP is not suitable for all situations. It can cause excessive use of network resources in high-congestion environments, and its use may require application adaptations that were not designed to support transmission reliability. Furthermore, RUDP does not guarantee delivery

In our project :

To guarantee the reliability of data transmission, we have chosen to implement a reliable transport protocol. We opted for a RUDP protocol (Reliable User Datagram Protocol), a simplified variant of the UDP protocol (User Datagram Protocol), which ensures the delivery of data without guaranteeing their order. In our implementation of RUDP, the server manages a flow control system that limits the number of packets sent by the client at the same time, in order to avoid network congestion. The server also keeps track of the packets sent, to ensure that each packet is received correctly and to avoid data loss. The client, on the other hand, sends numbered packets with a unique identifier for each packet, so that the server can verify that all packets are received correctly and in the correct order. If a packet is not received, the client retransmits the missing packet.

We have also put several flags or SIGNAL in place to allow interaction between the client and the server:

- SIGNEW: This signal is used by a new client to register with the server. The client sends this signal containing his username to the server, which records the client's information and sends back an acknowledgment (ACK).
- SIGET: This signal is used by a client to request a list of available files from the server. The server sends a response containing the list of requested files.
- SIGNECHO: This signal is used by a client to send an echo request to the server, which returns a response containing the response time between itself and the server.

- SIGNSTAT: This signal is used by a client to request statistics on server performance. The server returns a response containing the requested statistics.
- SIGNEND: This signal is used by a client to unsubscribe from the server and close the connection. The server returns an acknowledgment (ACK) and closes the connection with the client.
- SIGNACK: this signal is used by the server to confirm receipt of a packet sent by a client. The server returns this acknowledgment (ACK) to indicate to the client that its packet has been received.
- SIGNLOST: This signal is used by the server to inform a client that a packet sent by the client has been lost. The customer can then resend the missing package.

Conclusion :

In conclusion, we have set up a communication protocol based on the use of the UDP protocol with the management of the reliability of the data exchanged. We have implemented a flow and congestion control mechanism to optimize data transmission between the server and the clients. We have also defined different signals to enable smooth communication between server and clients, such as SIGNNEW for creating a new client, SIGNGET for data retrieval, SIGNLOST for signaling packet loss, SIGNFULL for informing a client that its congestion window is full, and SIGNEND for the end of the communication. All of these functionalities make it possible to guarantee the security and reliability of data exchanges between the server and the clients, while optimizing the transmission of data in an unstable and/or congested network environment.