Mobile Computing Wireless Network **27 August, 2024**

Lecture : Reliable User Datagram Protocol

Lecturer: Jigar Shekhat *Scribe/s: Romil Modi, Vishwajeet Muthe, Krishal Modi*

1 Introduction

The User Datagram Protocol (UDP) is a fundamental protocol in networking, widely used for sending messages, called **datagram**s, across the internet. It's known for being **fast and efficient** because it doesn't require setting up a connection before sending data. This makes it ideal for applications where speed is crucial, such as live video streaming, online gaming, and voice calls.

However, UDP is also considered unreliable because it doesn't guarantee that messages will arrive in order, without duplication, or even at all. This lack of reliability is acceptable in some scenarios but can cause significant issues in applications where data accuracy and order are essential.

1.1 Problem in UDP

The main problem with traditional UDP **is its unreliability**. Unlike the **Transmission Control Protocol** (TCP), which ensures that data is **delivered correctly and in sequence by using acknowledgments, retransmissions**, and other mechanisms, **UDP does not provide these guarantees**

1.2 Objective of RUDP

The Reliable User Datagram Protocol (RUDP) was created to address the reliability issues associated with UDP. RUDP seeks to maintain UDP's speed and simplicity while adding features like acknowledgments, retransmissions, and packet sequencing, which make it more reliable. This protocol aims to offer a balanced solution that provides the efficiency of UDP with the added reliability needed for more demanding applications.

2 The Reliable User Datagram Protocol (RUDP)

The Reliable User Datagram Protocol (RUDP) is an extension of the standard User Datagram Protocol (UDP) that adds reliability features typically associated with the Transmission Control Protocol (TCP).

While UDP is known for its speed and simplicity, it lacks mechanisms to ensure that data is delivered correctly and in order. RUDP was developed to fill this gap, providing a more reliable alternative that retains the low-latency, connectionless nature of UDP while incorporating additional features to improve data integrity and transmission reliability.

2.1 Comparison with standard UDP and TCP.

Feature	UDP	TCP	RUDP
Reliability	Unreliable; no guarantee of packet delivery.	Reliable; guarantees delivery, order, and integrity.	Reliable; adds reliability features to UDP.
Connection Type	Connectionless; no setup required.	Connection-oriented; requires a handshake before data transmission.	Connectionless, like UDP, but with added reliability mechanisms.
Overhead	Low overhead; minimal control information.	High overhead; includes error checking, acknowledgments, and flow control.	Moderate overhead; adds some control information for reliability.
Acknowledgments	No acknowledgments; packets are sent without confirmation.	Acknowledges every packet received.	Uses acknowledgments to confirm the receipt of packets.
Retransmission	No retransmission; lost packets are not resent.	Retransmits lost packets until they are successfully received.	Retransmits lost packets based on acknowledgment and timeout mechanisms.
Sequence Numbers	No sequence numbers; packets may arrive out of order.	Uses sequence numbers ensure correct order of packets.	Implements sequence numbers to reorder out-of-sequence packets.

Figure 1: comparison Table with UDP and TCP

2.2 Design Goals:

- **Reliability**: Ensure that all data packets are delivered accurately and in the correct order, similar to TCP, but while maintaining the simplicity of UDP.
- **Minimal Overhead**: Achieve reliability without introducing significant overhead, ensuring that the protocol remains lightweight and efficient for use in scenarios where low latency is critical

- **Compatibility with Existing Systems**: Design RUDP to be easily integrated with existing UDP-based applications and networks, requiring minimal changes to current infrastructure.
- **Scalability**: Ensure that RUDP can scale effectively, handling a large number of connections without significant performance degradation.
- **Flexibility**: Allow for configurable parameters such as timeout intervals and re-transmission limits to adapt to various network conditions.

2.3 Features of RUDP:

- **Acknowledgment Messages**: RUDP introduces acknowledgment (ACK) messages, which are sent by the receiver to confirm the successful receipt of data packets. This ensures that the sender knows which packets have been received correctly and can re-transmit any that were lost.
- **Sequence Numbers**: Each data packet in RUDP is assigned a unique sequence number. This allows the receiver to reorder packets that may have arrived out of sequence, ensuring that data is reassembled in the correct order.
- **Re-transmission Mechanisms**: If a packet is lost or an acknowledgment is not received within a certain timeout period, RUDP will re transmit the packet. This mechanism increases the reliability of the protocol, ensuring that all data is eventually delivered.
- **Flow control** : RUDP can implement flow control mechanisms to prevent the sender from overwhelming the receiver with too much data at once. This helps maintain efficient data transfer rates without causing congestion.
- **Error Detection**: RUDP includes error detection features, such as checksum, to identify corrupted packets. When an error is detected, the protocol can request re-transmission of the affected packets.
- **Timeout Management**:RUDP uses a timeout mechanism to detect lost packets. If an acknowledgment is not received within a specified time, the packet is considered lost and is re-transmitted.
- **Compatibility with UDP**: RUDP is designed to be compatible with existing UDP applications and networks. This means that it can be used in place of UDP without requiring significant changes to the underlying infrastructure.

3 RUDP Packet Structure

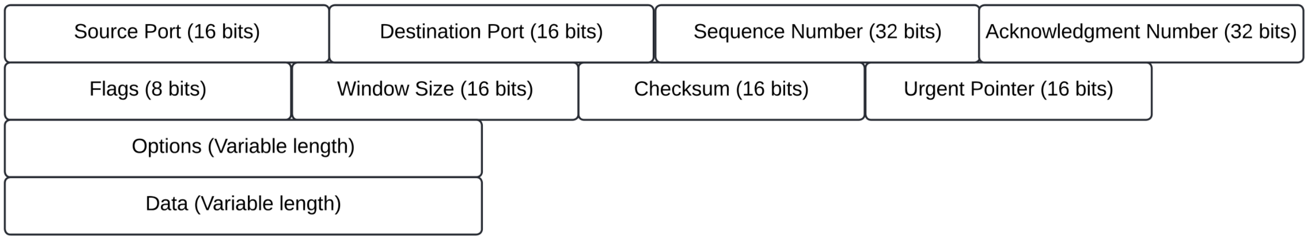

Figure 2: Header structure for RUDP

3.1 Comparison with UDP

Field	UDP	RUDP
Source Port	16 bits	16 bits
Destination Port	16 bits	16 bits
Length	16 bits	Not present
Checksum	16 bits	16 bits
Sequence Number	Not present	32 bits
Acknowledgment Number	Not present	32 bits
Flags	Not present	8 bits
Window Size	Not present	16 bits
Urgent Pointer	Not present	16 bits
Options	Not present	Variable length (optional)
Data	Variable length (data)	Variable length (data)

Figure 3: Comparison Between UDP and RUDP

4 Implementation of RUDP

4.1 Protocol Operations:

- **Handshaking**: Establish a reliable connection between the sender and receiver.
- **Data Transmission**: Ensure reliable, ordered, and complete delivery of data.
- **Closing Connections**: Gracefully terminate the connection while ensuring all data is delivered.

4.2 Algorithms:

1. **Retransmission Algorithms**: Retransmission algorithms handle the re-sending of packets when acknowledgments are not received in time.

   - **Timeout-Based Retransmission**: The sender maintains a timeout value for each packet sent.If an acknowledgment for a packet is not received within the timeout period, the sender retransmits the packet.
   - **Fast Retransmi**t: The receiver sends duplicate acknowledgments (DUPACKs) when it receives out-of-order packets.When the sender detects three or more duplicate acknowledgments for the same packet, it performs a fast retransmission of the missing packet without waiting for the timeout.
   - **Selective Acknowledgment (SACK)**t:The receiver acknowledges packets that are received successfully and provides information on the blocks of data that are missing.The sender can retransmit only the missing packets, not all packets after the lost one.

2. **Flow Control Mechanisms**: Flow control mechanisms ensure that the sender does not overwhelm the receiver with too much data..

   - **Window-Based Flow Control**: The receiver specifies a window size, indicating the amount of data it can buffer and process.The sender adjusts its transmission rate based on the receiver's window size.
   - **Congestion Control**: Manages the sender's rate of data transmission to avoid network congestion.Adjusts the transmission rate based on network feedback, such as packet loss or delay.

5 Application of RUDP:

1. **File Transfers**
2. **Real-Time Communication**
3. **Streaming Media**
4. **IoT (Internet of Things)**
5. **Database Replication**
6. **Messaging Systems**
7. **Firmware Updates**

6 Conclusion

In the realm of modern networking, the Reliable User Datagram Protocol (RUDP) represents a significant advancement over traditional UDP by bridging the gap between simplicity and reliability. RUDP retains the low overhead and efficiency of UDP while incorporating crucial features to ensure data integrity, order, and reliable delivery