Reliable Transfer Protocol 3.0

Amit Magar

Rochester Institute of Technology
Email: ajm6745@g.rit.edu

Table of Contents

Introduction

Reliable Data Transfer protocol 3.0 works on Transport Layer, it reliably transfers packet from a source to a destination over an unreliable transfer media. Protocol ensure every packet gets delivered to the destination in case of following scenarios

- Packet loss in network
- Packet corruption
- Acknowledgement lost
- Acknowledgement corruption
- Delayed acknowledgment

How Protocol works:

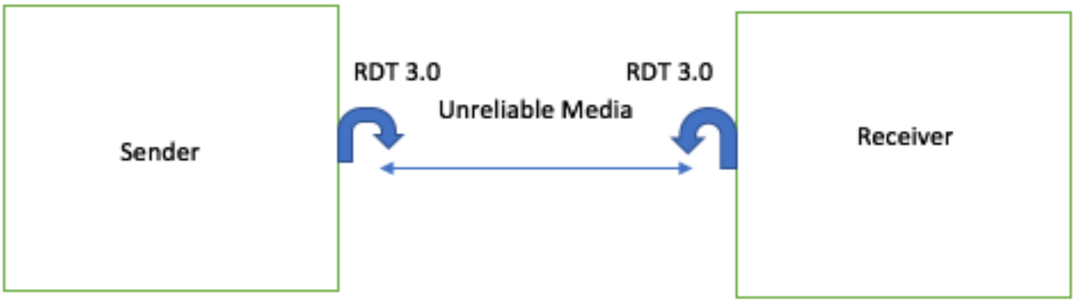

Figure 1Reliable Transfer Protocol 3.0

Basic building blocks of the Reliable Transfer Protocol

- ❖ Error Detection
  - ➢ A sender and a receiver detects if the packet they received has lost some data during transmission on an unreliable transfer media or not. This error detection is handled with the help of the Checksum field in packet.
  - ➢ A sender computes the checksum of a packet based on payload of the packet, attaches this checksum to the packet
  - ➢ A receiver computes the checksum of a received packet and compare it with the checksum field of packet. If the computed checksum and checksum field are same, then the receiver assumes that the received packet is not corrupted else receiver assumes the received packet is corrupted during transmission.
- ❖ Sequence Number
  - ➢ This protocol uses 1 bit sequence number 0 and 1.
  - ➢ A sender and a receiver maintains information about the expected sequence number.
  - ➢ A sender expects the sequence number of acknowledgement to be same as the packet it has sent.
  - ➢ At receiver side if the sequence number of last successfully received packet was 0 then the receiver will expect the next packet with sequence number 1.

- Feedback
  - For the each packet a sender sends it expects an acknowledgement feedback from the receiver. If a sender doesn’t receive the acknowledgement packet correctly, it assumes the packet delivery has failed and sends the packet again.
- Timer
  - Timer is used to detect packet loss on the network during transmission.
  - When a sender sends a packet to the receiver, the sender starts a timer with fixed expiration time. If it receives an acknowledgement packet before the timer expires then timer is stopped, else the sender assumes that the packet has been lost in the network and resend the packet again.

Procedure

Algorithm at sender side.

1. When RDT a sender receives the data from the upper layer it creates a packet for the received data, compute the Checksum based on the received data and adds sequence number to the packet either 0 or 1 alternatively.
2. Sender sends the packet to the receiver and starts a timer and wait to receive an acknowledgement from the receiver for the sent packet.
3. If timer expires, it again resends the packet.
4. If it receives acknowledgment before timer expiration and if the acknowledgement packet is corrupted it discards the acknowledgment, resends the packet again.
5. If the sender receives the acknowledgment before timer expiration and if the acknowledgement packet is not corrupt. It fetches out the sequence number from the acknowledgement, if this sequence number matches with the sequence number of the sent packet then the sender assumes the packet has been successfully received by the receiver and moves on to create a packet. Else it will resend the packet again.

Algorithm at Receiver Side

1. Receiver receives the packet and computes the checksum of payload data if it matches with checksum of received packet then it will consider this packet for further processing else it will send acknowledgement of previously successful packet receiver received
2. Check sequence number of the received packet it matches with packet sequence number the receiver is expecting then the receiver considers it for further processing sends acknowledgement of previously successful packet receiver received. (if the Sequence number of previously successfully packet at receiver is 0 then the receiver will expect next packet to have sequence number as 1 and vise a versa)
3. Receiver will create the acknowledgement packet with the same sequence number of the correctly received packet and send it back to the sender.

Result and Discussion

When RDT 3.0 is used for packet transfer I observed that sender and receiver could send and receive all the packet successfully on unreliable transfer media.

Different scenario handled by RDT 3.0 are as follows

Scenario 1

Following is scenario when sender sends packet to receiver (a non-corrupt packet), sender receives the acknowledgement packet corrupted or dropped or normal, and the receiver receives packet with incorrect sequence number.

Sender: - (First attempt)

```
Sending new packet to receiver
Received corrupt acknowledgement
```

At receiver side: -

```
Received Packet from sender
Sequence Number of Packet :1
Contents of Packet are: This is Message from Client0
Check Sum is :2524
Sending positive acknowledgement
Simulating Corrupted Acknowledgement
```

Sender: - (Second attempt)

```
Resending earlier packet
Acknowledgment lost in network packet
```

At receiver side: -

```
Received Packet from sender
Sequence Number of Packet :1
Contents of Packet are: This is Message from Client0
Check Sum is :2524
Expecting Packet with sequence 0 sending acknowledgement of previous packet
Simulating Acknowledgement Drop
```

Sender: - (Third attempt)

```
Resending earlier packet
Received acknowledgement successfully
```

At receiver side: -

```
Received Packet from sender
Sequence Number of Packet :1
Contents of Packet are: This is Message from Client0
Check Sum is :2524
Expecting Packet with sequence 0 sending acknowledgement of previous packet
```

Scenario 2

When sender and receives packets with **proper sequence number** with **no network failures.**
Sender:

```
Sending new packet to receiver
Received acknowledgement successfully
```

At receiver side:

```
Received Packet from sender
Sequence Number of Packet :1
Contents of Packet are: This is Message from Client2
Check Sum is :2526
Sending positive acknowledgement
```

Scenario 3

When receiver receives corrupted packet from receiver.

Sender:

```
Sending new packet to receiver
Simulating corrupted packet
Received previous acknowledgement
```

Receiver:

```
Received Packet from sender
Sequence Number of Packet :0
Contents of Packet are: This is Message from Client3
Check Sum is :2528
Received corrupted packet sending acknowledgement of previous packet
```

Scenario 4

When timer expires at receiver side.

Sender:

```
Resending earlier packet
Acknowledgment lost in network packet
```

Receiver:

```
Received Packet from sender
Sequence Number of Packet :0
Contents of Packet are: This is Message from Client3
Check Sum is :2527
Sending positive acknowledgement
Simulating Acknowledgement Drop
```

Scenario 5

Sender:
When packet is lost before it reaches to receiver

Resending earlier packet
Simulating packet drop
Acknowledgment lost in network packet

Receiver:

Packet time out

Conclusion and future work

I learned what failure could happen during packet transmission in network and how RDT 3.0 handles those failures. Using RDP 3.0 we can reliably transfer data on the unreliable transfer media. Different failure scenarios make successful transmission of the packet from a source to a destination complicated, Because of these failures network utilization using RDT 3.0 becomes less to improve network utilization we can add approaches like cumulative acknowledgement, selective repeat or packet pipelining to RDT 3.0.