RDTP - Reliable Data Transfer Protocol

Protocol Performance Analysis

Overview

This document provides an analysis of the RDTP protocol by varying different elements of a network connection like packet loss, packet delay, packet corruption, packet duplication, packet reorder and network jitter.

Network simulation is performed using the **Netem** tool.

System Architecture

A Computer with the following specifications have been used to perform the analysis:

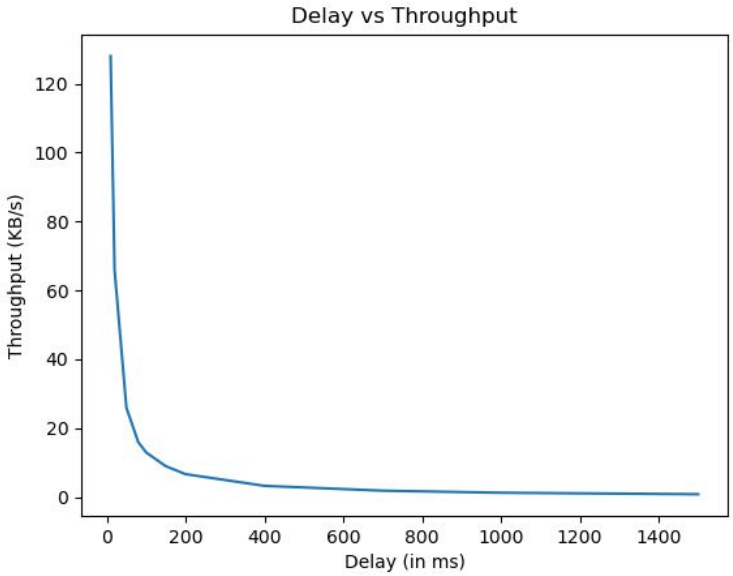
- CPU: Intel i7 7th gen 7700HQ
- RAM: 8GB
- OS: Arch Linux 5.6.6-arch1 64bit
- Kernel: Linux 5.6.6

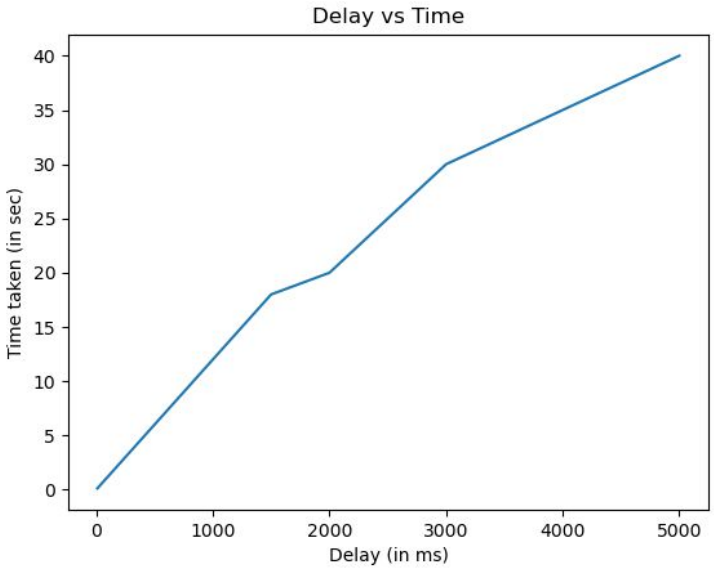
Analysis

Analysis is performed for a file transfer application which uses the RDTP library to send/receive the file data. A single binary file having a size of 16.2 KB is used for file transfer analysis.

The file is a binary which is generated by a compiled C program which is a basic input output program. This file is selected for the purpose of ease of running and checking if the binary works properly.

Packet Delay

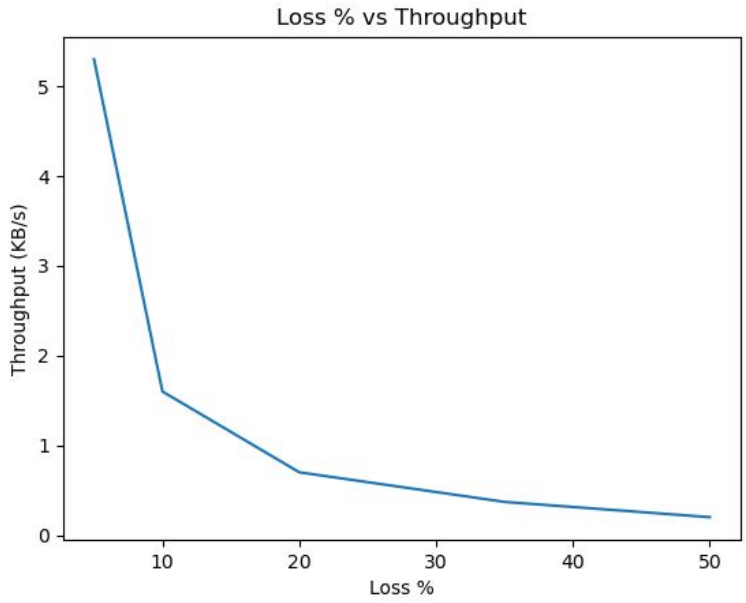


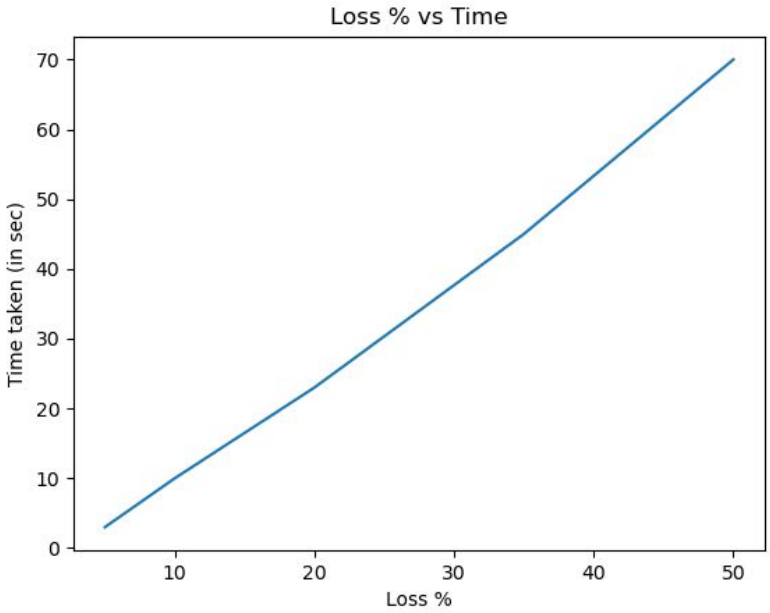


Observation: As expected, the file transfer time increases with increase in packet delay. But here, we can observe that the throughput exponentially decreases for small delays and remains constant after 800ms delay at around 1 KB/s or 8Kbits/s.

Packet Loss

We have simulated packet loss from 0 to 50% using the netem tool. It was not possible to perform analysis for loss > 50% as the throughput becomes very small and hence takes a lot of time.

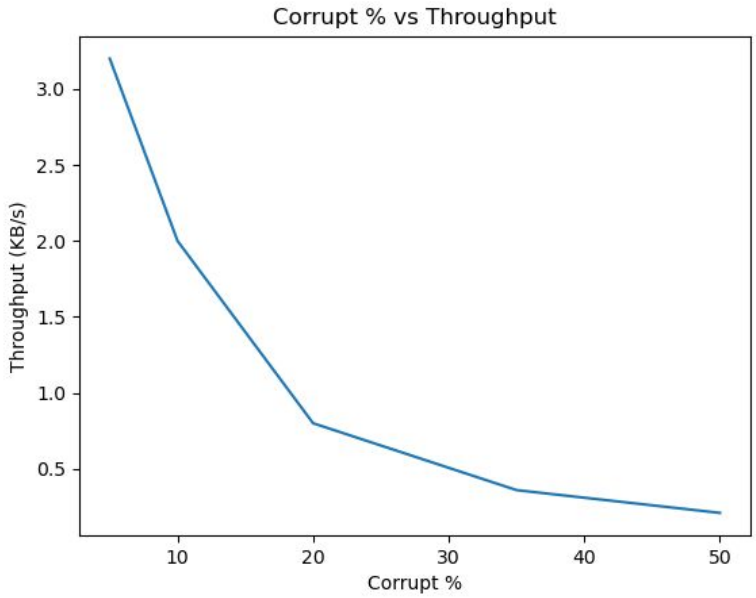


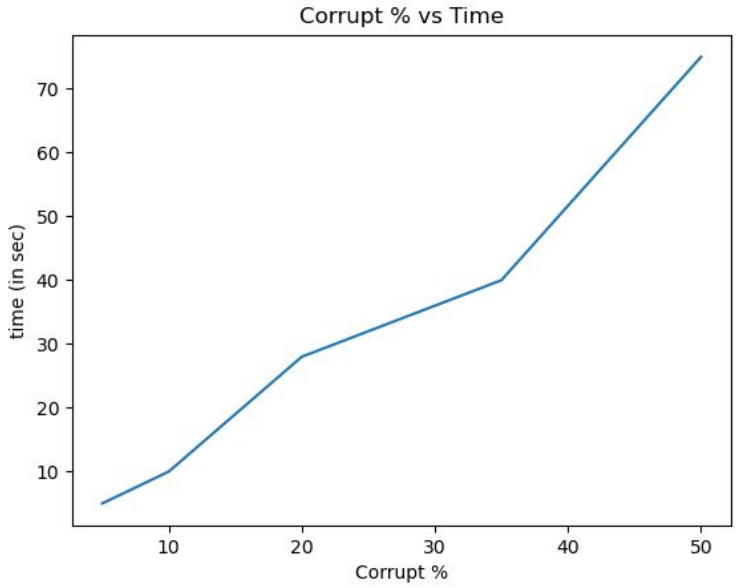


Observation: It is observed that as loss increases, time taken also increases in a linear fashion. But the effect of Loss % on throughput is very high. For loss > 50%, the throughput falls below 0.5KB/s which is not sufficient to transfer large files.

Packet Corruption

Packet corruption is caused when there is a change in the data of a packet. This can be caused due to the hardware properties or due to a malicious intent. In RDTP, each packet has a unique checksum corresponding to its headers and payload. If there is corruption, the checksum would not match and hence the packet would be dropped.

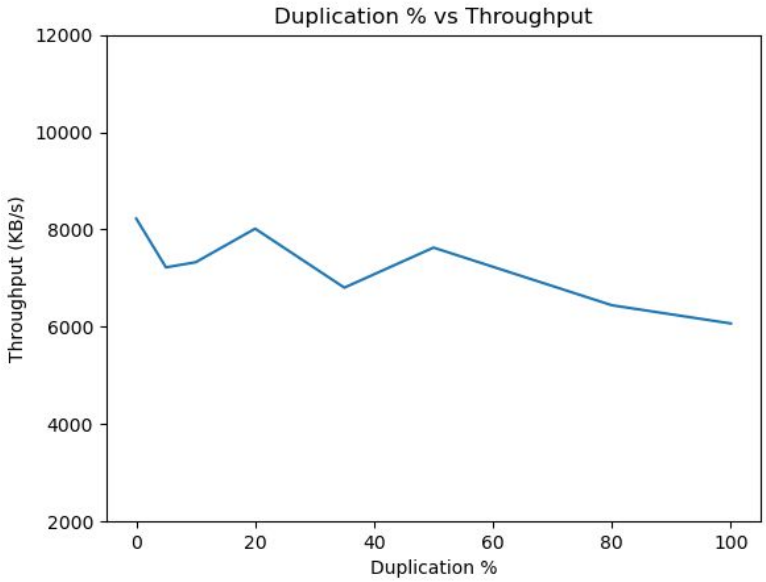


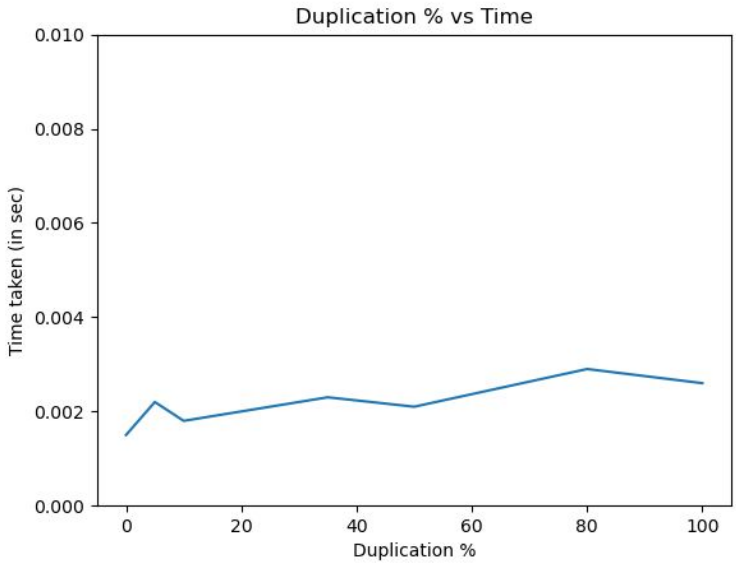


Observation: We can observe that the packet corruption case is also similar to packet loss. This is due to the fact that dropped packets directly imply that the packet is lost. Hence, we see that the same trend is followed for packet corruption also.

Packet Duplication

In a real world scenario, packets are routed through different ways and there can be instances of packet duplication.

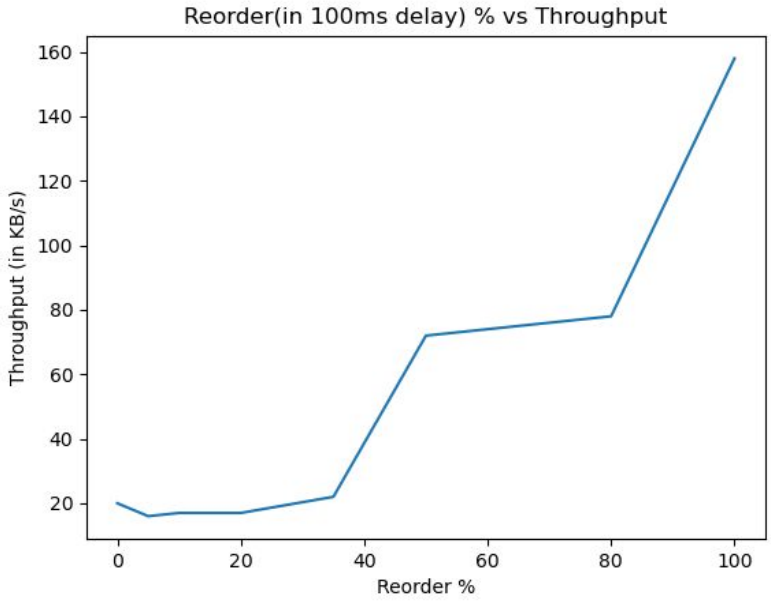


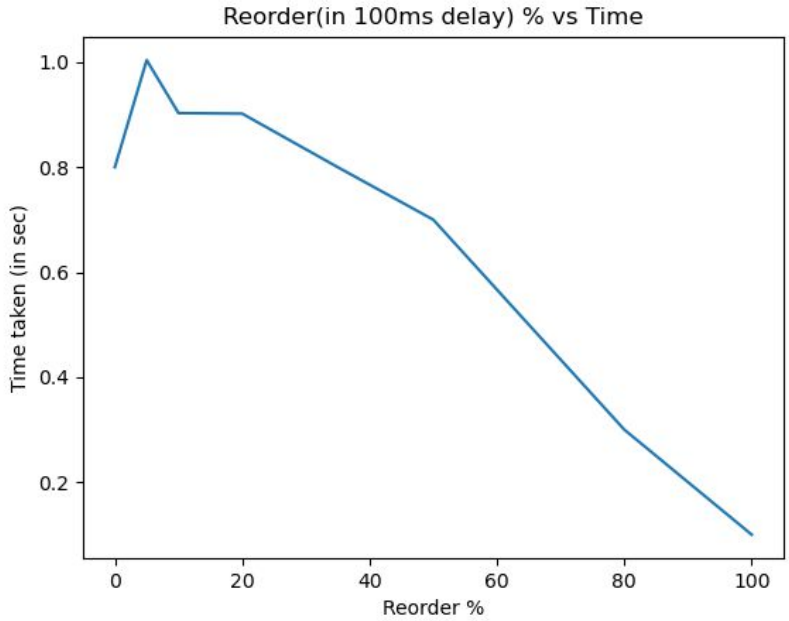


Observation: It is observed that packet duplication has no trend and oscillates between 6MB/s and 8MB/s. Keep note that these are obtained assuming that there is no packet delay which leads to high throughput. It can be assumed that duplication has almost no effect on the protocol as RDTP silently drops any duplicates it receives. Therefore, the changes observed in the above graphs might be due to the filling up of socket buffers with duplicate packets

Packet Reorder

Packet reorder, as used by the netem tool, required a delay to set reorder percentage. So, we have selected 100ms delay for testing purposes.

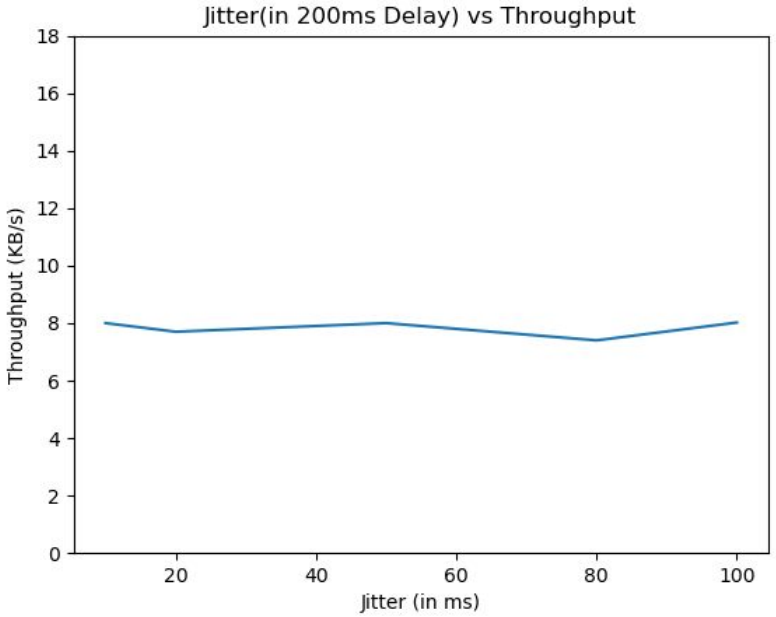


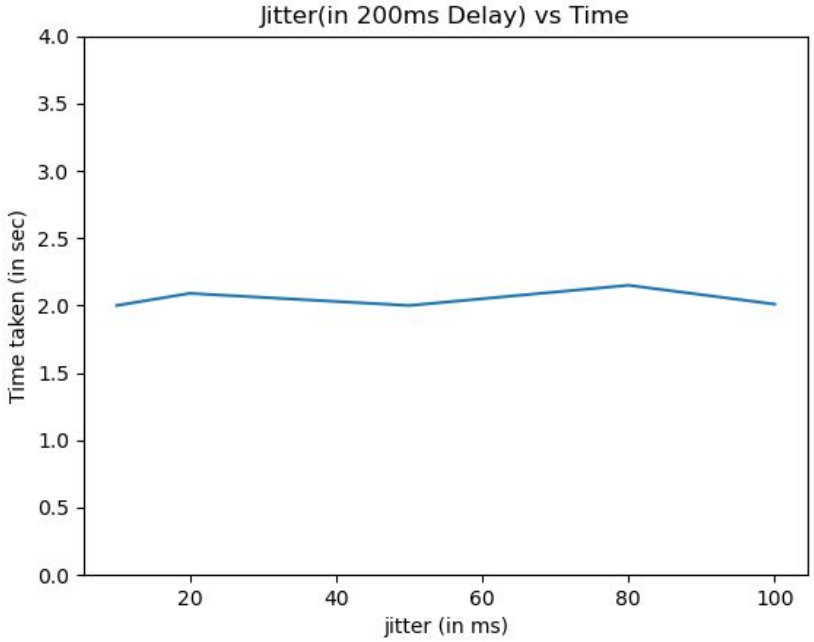


Observation: We have observed that increasing reorder percent, our protocol somehow performs better which should not be the case. RDTP uses a selective repeat, so theoretically, there should be no difference in time or throughput by simulating packet reorder. A possible explanation is that simulating lower reorder % is causing a high overhead on the socket.

Network Jitter

According to Wikipedia, **jitter** is the deviation from true periodicity of a presumably periodic signal. In terms of network jitter, it refers to the varying delay of in between 2 consecutive packets. Similar to packet reorder, netem requires a specified delay to set jitter.





Observation: It is observed that at 200ms delay, even a jitter of 100ms has almost minimal effect on the application transfer throughput or time.