Evaluating TCP performance with packet duplication

pc1074
Benny Pingng Chong
University of New Hampshire
April 2019

***Abstract*—Packet loss is one of the main reasons of worsening the TCP throughput performance. More precisely, the repercussion of packet loss costs the time that slows down the transferring time of data. That said, reducing the re-transmission by packet loss is a significant point to improve the TCP throughput performance. To avoid packet loss, duplicating packets is a possible method. This concept is similar to the forward error correction (FEC). The intentional packet duplication improves TCP throughput performance when packet loss plays a key role.**

I. Introduction

The focus of this paper is on the impacts of packets duplication. The paper examines at what point the packets duplication makes a significant improvement. Moreover, the paper also aims at figuring out the network performance with different duplication rates under different network conditions, precisely speaking, latency and packet loss.

II. Methodology

A network emulator is placed between enp4s0 interfaces of rb1 and rb2. Thanks to the emulator, we can set different values of loss, latency, and duplication rate. The loss rate will be taken as an independent variable, while the throughput rate would be the dependent variable. By sending a sufficiently large block of data, we can collect and compare the data between different loss/latency/duplication configuration, which will give us an idea of TCP throughput performance under varying packet loss, network latency, and duplication. Unlimited bandwidth mode is set for tests that can possibly utilize the bandwidth. The maximum bandwidth is around 1Gb/sec. Iperf 2.0.10 is used in the test.

III. Packets duplication under low latency condition

This test shows the result of TCP throughput rate after implementing packets duplication under low latency condition. Using a net-storm emulator, 0.1ms is set between rb1 and rb2. Since the latency is trivial, this test can show the how packets duplication improves the network performance that is mainly affected by packet loss.

Figure 1 indicates that duplicating packets make a difference. Each curve represents a specific percentage of duplication. In general, the higher percentage of packets duplication, the flatter the curve is. In other words, increasing the duplication will slow down the decrease of throughput. On the other hand, TCP with duplication does not outperform TCP without duplication when there are no packets loss. When the percentage of duplication is over 50%, the throughput rate is under 80% with 0% packet loss, whereas the throughput rate is over 90% when the percentage of duplication is less than 50% in the 0% packet loss condition. The 100% duplication prevails in the long term, although it does not start as good as other curves do. It's throughput does not drop as rapid as other curves do, in spite of the sudden drop at around 5% packet loss. But still, the throughput of the pink curve(100%) has not dropped below 20% when the packet loss is less than 10%.

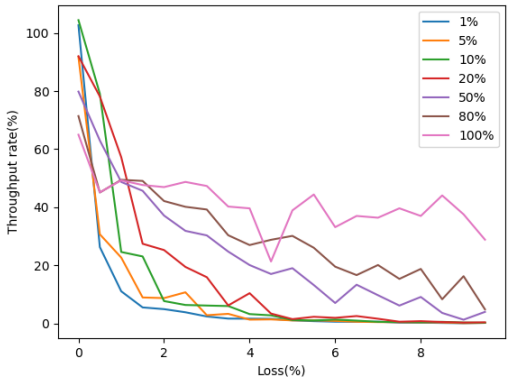

Fig. 1. Throughput rate vs loss with 7 different duplication rates under 0.1ms latency

IV. Packets duplication under high latency condition

As we expected, once we set a high latency, the overall throughput performance is worse. Figure 2 shows that the curve 0%, 25%, and 50% do not make a huge difference, even the 75% one is closed to those three. It suggests that intentional packet duplication offsets the impacts from packet loss, however, as soon as latency is high, packet duplication will not be as useful as it does in the low latency scenario. It is noteworthy to point out that the purple curve(100%) outperforms other curves the most in the range of 1% to 4% packet loss.

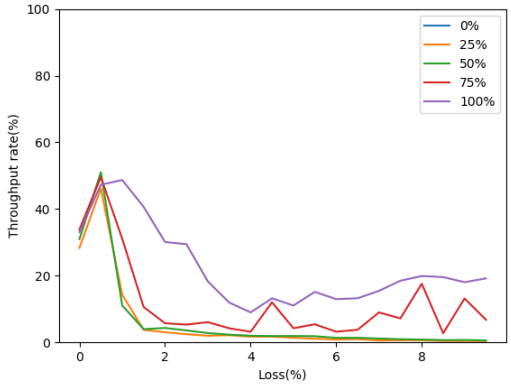

Fig. 2. Throughput rate vs loss with 5 different duplication rates under 2ms latency

V. Limitation

There are several more tests that could be done. What if we test with more packet loss? What if we test with higher latency? Unfortunately, due to the computation limitation, those are not covered in this paper.