

2022 Future Computing Summer Internship Project: (Mapping Congestion Collapse to the SST model)

Nicholas Schantz*, anotherfirst[†]

July 16, 2022

Abstract

Congestion Collapse is a issue involving reliable network protocols that send large amounts of retransmission clogging a link with duplicate data. This leads the network in a state where useful throughput has vastly decreased or does not exist over time. SST will be used to model the problem and parameterized so that the problem can be simulated and metrics can be collected which identify if congestion collapse exist. We determine that the ratio of goodput and throughput in the link can assist in identifying the existing of the problem. Furthermore, we determine that a link's send queue depth and ratio of first packets to duplicate packets directly measure the existence of congestion collapse.

1 Project Description

The challenge addressed by this work is to map the networking problem 'Congestion Collapse' to the SST model. The problem is simulated to identify mathematical conditions that cause the problem. This information is used to develop metrics to identify that the problem has occurred in generalized systems.

2 Motivation

Identifying the metrics for detecting congestion collapse will be vital for developing distributed systems that can avoid congestion collapse from occurring during communication. Furthermore, the metrics answer the question as to why the problem has occurred in the system. To add, the SST models written are resources that other users can use to learn and utilize SST's discrete event simulator.

3 Prior work

4 How to do the thing

The software developed to respond to this challenge was run on one laptop. The software is available on (<https://github.com/lpsmodsim/2022HPCSummer-CongestiveCollapse>)

5 The Model

There are two models that simulate the behavior of congestion collapse. Both models contain a n sender node that send data to a single receiver node.

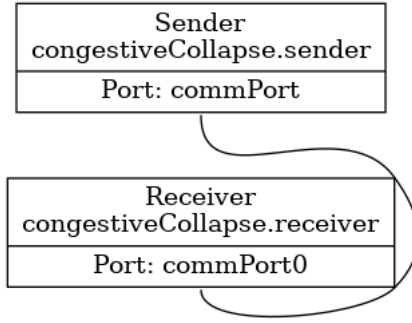


Figure 1: Example of model with one sender and receiver node

The models differ by the properties of the sender and receiver nodes. In the first model, the sender will send a set number of packets per tick while the receiver will process a set number of packets in its queue. In the second model, the sender and receiver both send and process packets one time a cycle. This was done to better implement the sliding window sending process for the TCP reliable networking protocol. Both model's receiver nodes will have an infinite queue to avoid implementing packet loss.

6 Result

To determine if congestion collapse exist in a network. Three potential metrics were determined:

Calculating a ratio between the useful throughput and the total throughput of a link gives insight to how many duplicate or corrupted packets have passed through a link. However, if this ratio is low due to corrupted packets, that does not indicate congestion collapse. In this case queue depth is measured and if it is largely being filled with duplicate packets rather than new packets, then we can declare that the network is being flooded with retransmissions. In this case these metrics can identify that congestion collapse exist in the network.

An example of the problem is simulated and gathered data is shown below. In this SST simulation, a node is sending packets at 10 times the rate that a receiver node can process them at. In this case the receiver node's queue fills quickly and the receiver is unable to send acknowledgments under the sender's timeout time. Retransmissions are sent which cause more packets to fill the queue and the useful throughput exponentially decays to zero in a short span of time.

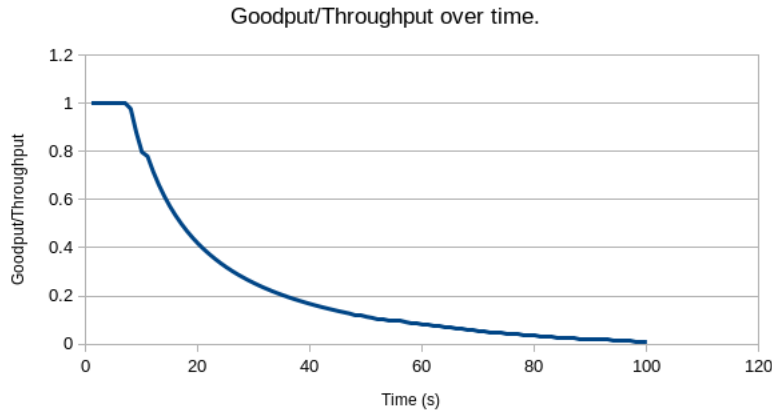


Figure 2: Ratio of goodput to throughput over a span of 100 seconds

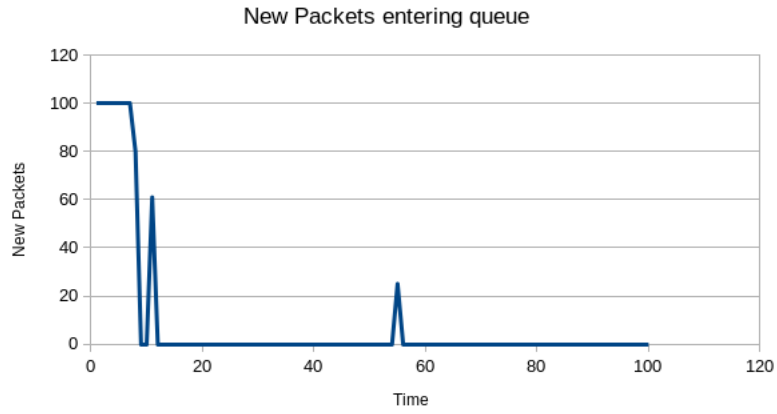


Figure 3: Amount of new packets entering the queue each second for 100 second

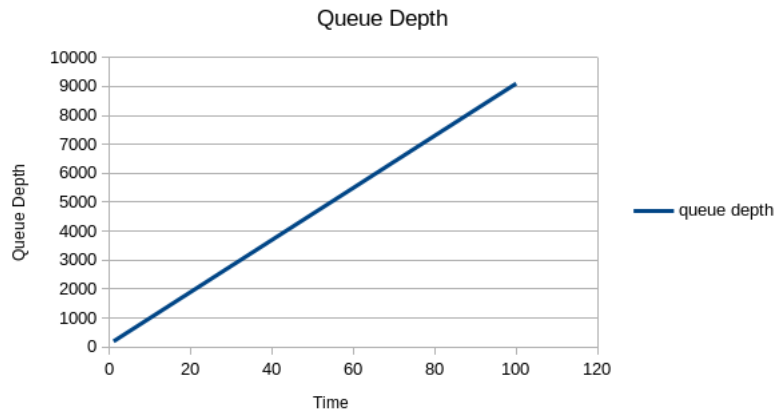


Figure 4: Infinite queue which increases over time.

7 Future Work