

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

Nicholas Schantz\*, anotherfirst<sup>†</sup>

July 6, 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

Research for determining metrics for the detection of congestion collapse was not found. However, research on the conditions that trigger this problem are actively researched. See [?]

## 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 Progress

Three SST models were made in the attempt to collect useful data to determine metrics which identify the existence of the problem. The first model was to map a cargo shipping company to the SST model, this model quickly became confusing because of failed attempts at combining TCP protocol to a real world example. For instance, correctly implementing sliding window when a node sends multiple messages per cycle became a problem due to data racing when multiple senders were added into the composition.

The second model attempted to simplify the simulation by limiting the simulation to a sender node and a

receiver node. The receiver has an infinite queue and sender will send a variable amount of packets per cycle. The issue with this model was the usage of an infinite queue which limited packet loss in the simulation. However, packet loss needs to be accounted for as retransmissions can be duplicate packets but also first packets which were dropped previously and are still goodput. Furthermore, we will need to parameterize queue depth for metrics.

In this case, a third iteration of the simulation is being written which attempted to be a simple implementation of a reliable network protocol. Changes to this simulation was to remove the infinite queue which allows queue depth to be parameterized for data gathering. In this case the idea of packet loss was also added which would need to be accounted for since a resent packet loss could still be goodput.

The simulator is used to purposefully cause congestion collapse by having  $n$  clients send a set of frames at a rate that the link will be unable to process before the client's timeout and retransmit their data.

## 6 Result

To determine if congestion collapse exist in a network. Three 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.

The following graphs will be changed when the third model mentioned above is complete. This new model is being developed to also work on the problem TCP Global Synchronization as well.

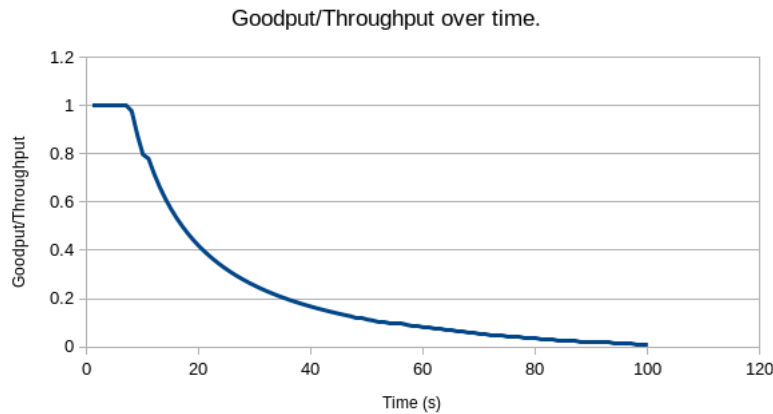


Figure 1: Ratio of goodput to throughput over a span of 100 seconds

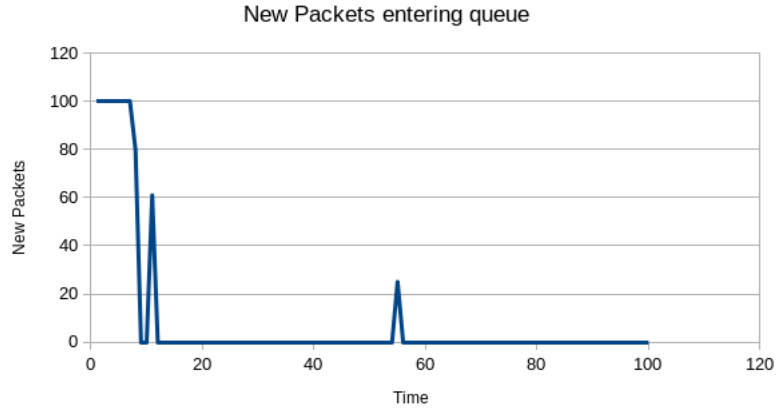


Figure 2: Amount of new packets entering the queue each second for 100 second

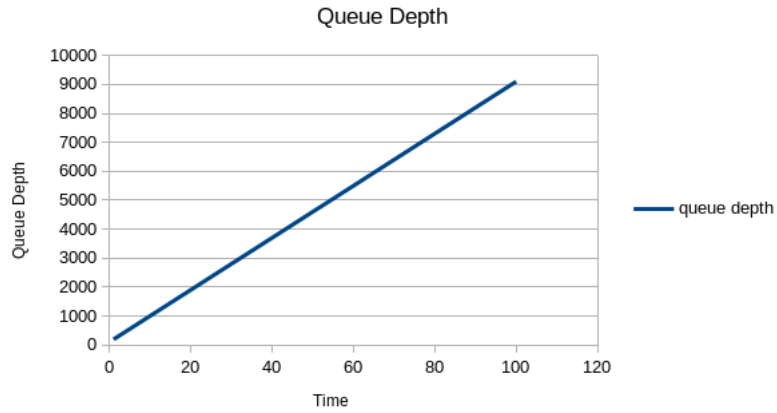


Figure 3: Infinite queue which increases over time. This queue will be finite in the next model

## References

- [1] ADD BIB FILE
- [2] ADD BIB FILE