

2022 Future Computing Summer Internship Project: (Mapping TCP Global Synchronization to the SST model)

Nicholas Schantz*, anotherfirst[†]

July 15, 2022

Abstract

TCP Global Synchronization is a networking problem in which a burst of traffic in a network causes multiple clients to drop packets and limit their transmission rates. The clients begin to increase their transmission rates consecutively leading to more packet loss and transmission limiting, creating a loop of this activity. SST is used to model this activity and find metrics that can determine if this problem exist in a simulation. A key metric found is to look in a window of activity when packet loss occurs and measure how many clients have reduced their transmission rates.

1 Project Description

The challenge addressed by this work is to map the networking problem 'TCP Global Synchronization' to the SST model. The problem is studied to understand the mathematic conditions that create this problem. This information is used to create a SST model and simulate it to identify metrics to detect that the problem has occurred in simulated systems.

2 Motivation

Identifying the metrics for detecting TCP Global Synchronization will be vital for developing distributed systems that can avoid congestion collapse from occurring during communication. The metrics allow for these problems to be detected when simulating large scale distributed systems, to prevent the problem from occurring when the real distributed system is put into production. 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

The following research does not involve discovering metrics that can detect this problems existence in simulation; however, it was useful in understanding the problem and how different packet dropping policies create or reduce global synchronization. [?]

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-TCPGlobalSynchronization>)

5 SST Model

The model is a simplified version of a reliable network protocol with a tail drop queue management policy. It makes the following assumptions:

- Sender components use the same protocol. They limit their transmission rates to the same value.

- Transmission rate limiting occurs immediately after a sender is notified that its packet was dropped.
- Senders increase their transmission rate's linearly after each tick.

The model involves a receiver component that has n ports which connect to n sender components. The sender components will send packets to the receiver's FIFO queue, and the receiver will process a set number of packets in the queue per tick.

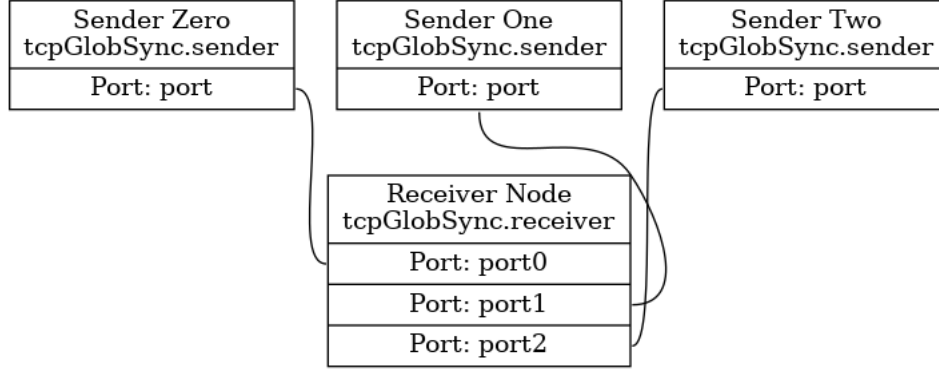


Figure 1: Connection between 3 sender components and 1 receiver component.

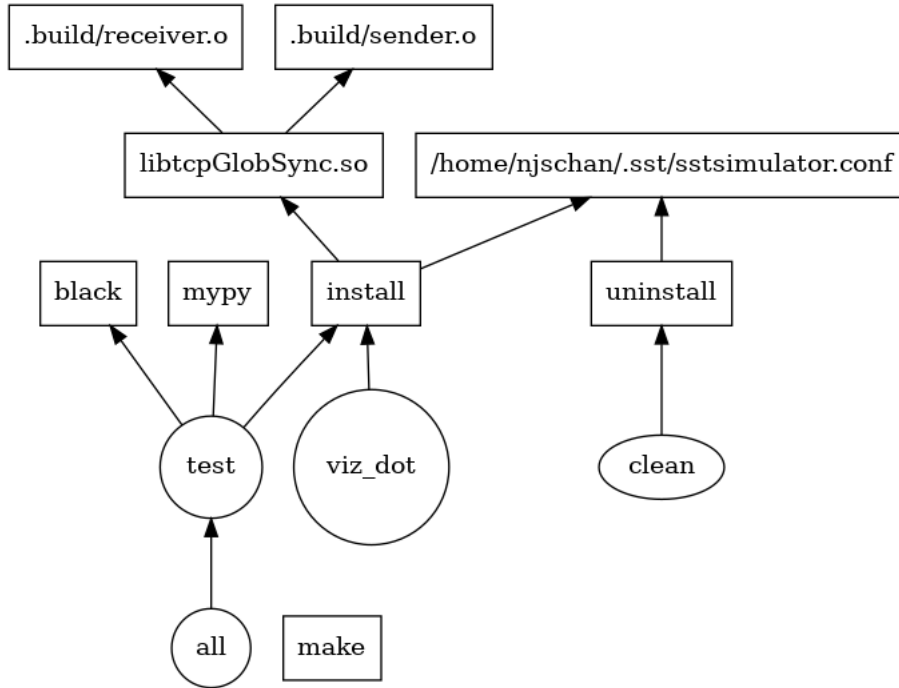


Figure 2: Visualization of Makefile

6 Result

Potential metrics include the following:

- Average aggregate link utilization compared to peak aggregate link utilization
- Queue size fluctuation
- Time in which clients drop

Average aggregate link utilization was chosen as a unique pattern forms when global synchronization occurs. Due to clients syncing and lowering transmission rates all at once, the link can rest for a bit and clear up its queue and will process messages under its capacity, lowering link utilization. As transmission rates increase, link utilization will grow to 100% and this cycle will repeat when packet loss occurs again.

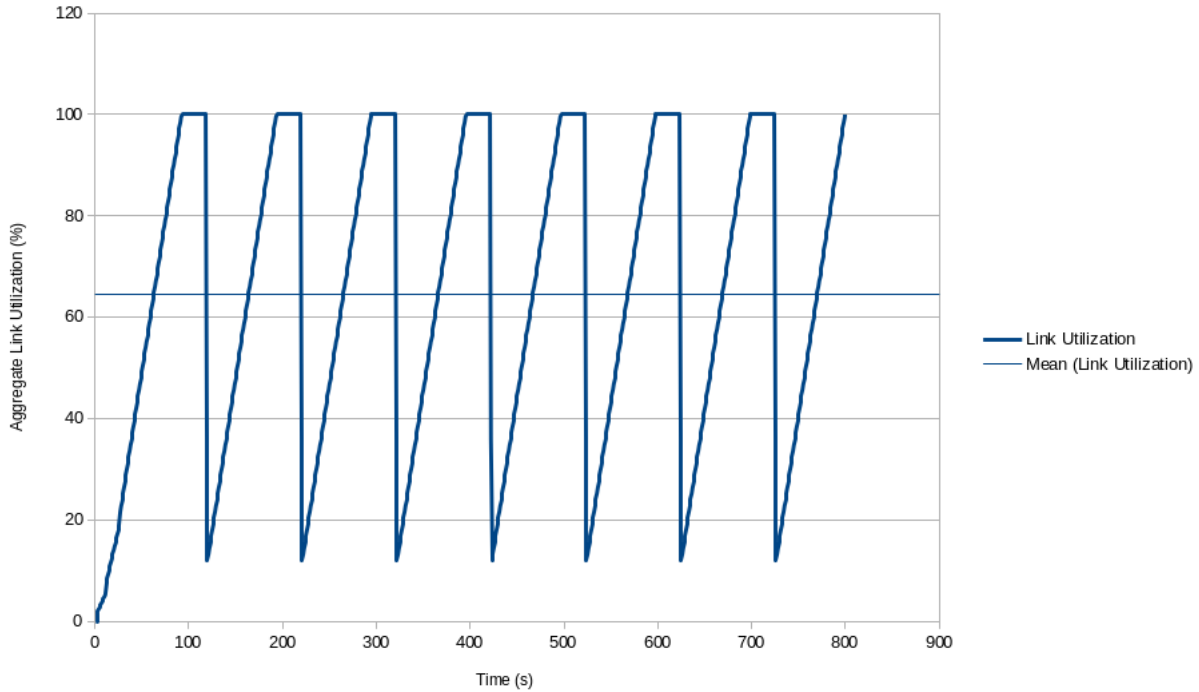


Figure 3: SST Model collected data of aggregate link utilization over 800 seconds.

However, this metric was thrown out when I researched RED (Random Early Detection) which is a packet dropping policy in reliable networks that attempts to avoid global synchronization. Average aggregate link utilization under this dropping mechanism appears to be similar to tail drop except the average is closer to peak utilization. In this case using this metric may alert on false positives since its based on measuring where the mean utilization is compared to the peak utilization.

{ Add image of RED with its average link utilization }

The queue size fluctuates quite often during global synchronization as it will empty out more often during lower link utilization and it will grow during peak link utilization. I thought it would be possible to measure the time span in which the queue size is emptied out and growing to max capacity, to determine if that span of time matches the duration between multiple transmission rate limits of a client.

(Potential picture)

However, during periods of low latency and high link utilization, the queue will be fluctuating drastically. This is because the link is in a stable state where it is clearing its queue as it fill. Therefore this metric could produce false positives and was tossed out.

Measuring the number of clients that limit their transmission rate in a window of time appears to be the most accurate and precise measurement for detecting this problem. Global Synchronization occurs because the clients are synchronized, therefore measuring the times they limit transmissions can determine if they are showing synchronous behavior.

To measure this behavior, a node will monitor the clients connected to the link. When the monitoring node detects that a client is limiting its transmission rate, the monitor will begin sampling input for a window of time. If all nodes limit their transmission rate in the window, the possibility of global synchronization occurring increases at a (arbitrary (uh oh)) percentage. To determine a high chance that this behavior is occurring, the monitor will sample multiple times when it detects a client limiting its transmission rate. This metric is demonstrated on three simulations which simulate three different scenarios of network traffic.

The first simulation is of three clients that send at a maximum rate that is under the links capacity. The link's queue will not fill so packet loss will not occur, and clients will not limit their transmission rates.

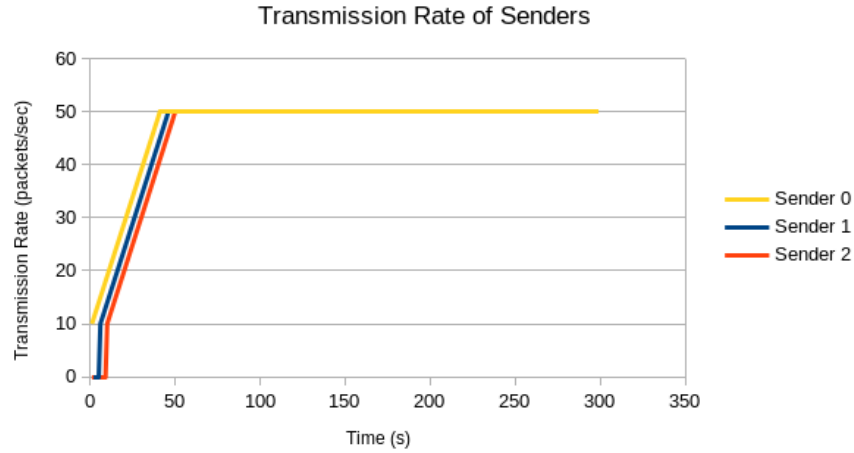


Figure 4: Transmission rates of the three senders. No rate limits occur so global synchronization does not occur.

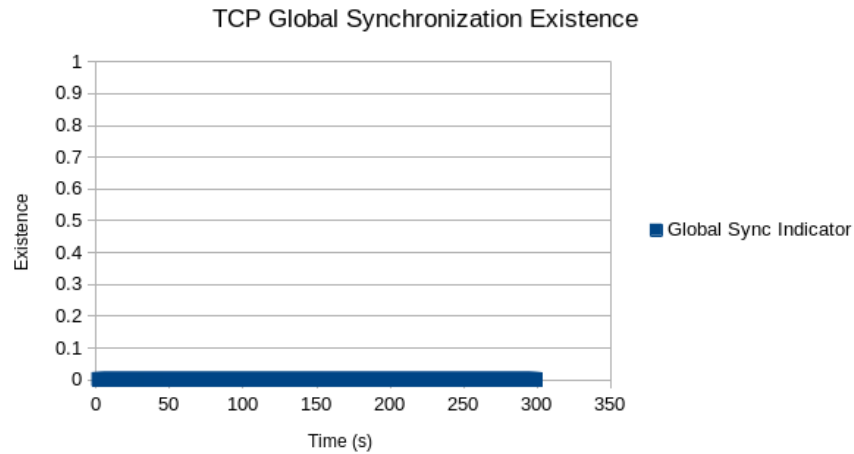


Figure 5: Metric does not detect the existence of global synchronization in the simulation.

{ Add second simulation and figures of network traffic with random send delays (Potentially implement pseudo-red dropping policy) }

The third simulation models two clients sending packets to a link at a rate in which the queue does not fill up so packet loss can not occur. A third client is introduced and disrupts this by sending packets which causes the queue to fill and packets to be dropped. All clients send packages simultaneously so they all encounter packet loss and limit their rates between a small period of time. This creates the problem of

global synchronization as they attempt to increase their transmission rates over time which leads to more packet loss and rate limiting.

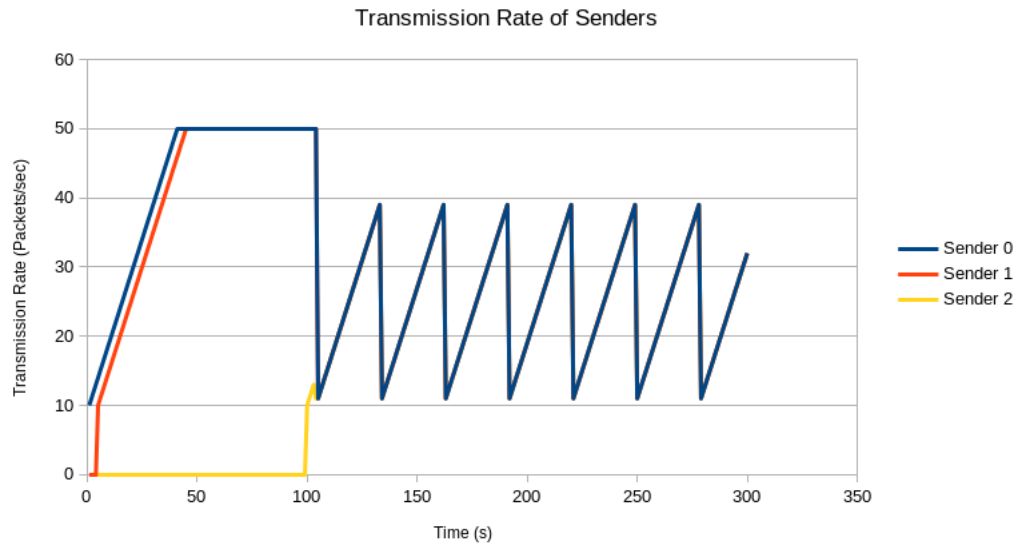


Figure 6: Transmission rates of the three senders over time.

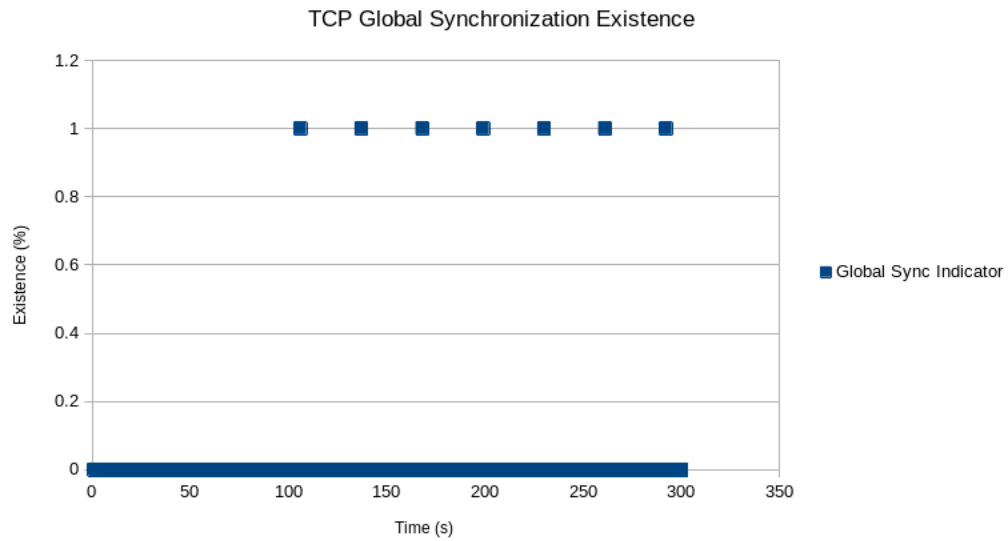


Figure 7: Metric detecting the existence of the problem in the simulation.

7 Future Work

Potentially no future work is needed.

References