Journal Homepage: -www.journalijar.com
INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)
Article DOI:10.21474/IJAR01/ 9566
DOI URL: http://dx.doi.org/10.21474/IJAR01/9566
ISSN NO. 2320-5407

RESEARCH ARTICLE

MODIFIED SLOW START FOR TCP CONGESTION CONTROL.

Advait Thakkar And Aditya Panchal.

Manuscript Info

Manuscript History
Received: 12 June 2019
Final Accepted: 14 July 2019
Published: August 2019

Abstract

Introduction:-

Computer networks went through massive and rapid growth over the last few decades and with that growth have come severe congestion problems; basically, when the packet sending rate is increased than the receiving rate then congestion arises example given, it is often observed that internet gateways drop about 10% of the incoming packets due to overflow of local buffers and much of the time these problems are caused due to transport protocol implementations (not in the protocols themselves): The 'obvious' ways to implement a window-based transport protocol can result in exactly the wrong behaviour in response to network congestion.

The base shape of the TCP algorithm was presented by Van Jacobson [1] Exponential increase of the sending rate was instigated; the initial window size was set at three packets. This protocol offered fast data transfer, congestion control, reliable connection and flow control. The mechanism was efficient back in time when 56K modem communication was common. In today's time, the broadband links are described as connections providing speed faster than 1Mbps. Gradually TCP was in the process of amelioration, a variety of protocols were announced, each designed with the goal to supply better network efficiency. Slow Start, congestion avoidance and flow control are the main phases in which they differ. TCP was developed to provide best-wired network utilisation. Limited Slow Start [2] Larger Initial Window, TCP Reno, Dual, Fast, Bic, Sack, Hamilton, Hybla, Scalable,Cubic, Abs, Fack, Linux, Highspeed, New Reno, Vegas, Westwood, Veno, Low priority, Illinois, Compound, Snoop, Full are some of the algorithms and TCP versions that were developed to improve the network utilisation. Most of the developed protocols were targeting wired network utilisation; work is done in wireless traffic utilisation too. Initial network conditions are changed; today we are dealing with broadband networks, the need for faster slow start phase is obvious if we want to provide better network utilisation. This is the main driver of our research and motivates us to modify the slow start phase. Even though users have greatly benefited by the discovery of TCP slow start algorithm experiencing faster downloads since slow start finds and uses the maximum connection speed. it still consumes the duration of time in deciding the optimised capacity to transfer over the channel, which can be reduced and the maximum acceptable size can be chosen in a faster way. The paper is organised as follows: Section II gives a brief overview of the slow start phase, discusses some related work and motivates the need for our approach. Section III describes our proposed algorithm and section IV presents the mathematical calculation and comparison. Section V concludes the paper.

Corresponding Author:-Advait Thakkar, Aditya Panchal.

Section II: Brief about Slow Start

TCP slow start performs the key role in balancing the throughput in a network. It increases the amount of data transmission in a gradual way until it finds the maximum acceptable data packet size. In a network, transmission is dependent on things such as Receiving window size (rwnd), i.e. the maximum data size packet that can be received by end system, Congestion Window (cwnd) is a TCP state variable that constraints the data size the TCP can send into the network before receiving an ACK, channel size - maximum data rate in the medium used for transmission of data & round trip time which is the length of time it takes for a signal to be sent and the length of time it takes for an acknowledgment of that signal to be received. This time delay includes the propagation times for the paths between the two communication endpoints. [3]

The common way to optimise throughput in a connection is to increase the bandwidth. However, any medium can be overloaded if a device tries to send out data in abundance. When a medium reaches its saturation point and crosses the limit it is called congestion, resulting in buffered connection or data loss in some cases.

Slow start intercepts network congestion by regulating the amount of data that is sent over the network. It settles the data size that is acceptable between a sender and receiver by defining the amount of data that can be transmitted with each packet, and gradually increases the amount of data until the network's capacity is reached so that maximum amount can be transmitted over the medium. Slow start ensures that as much data is transmitted as possible without choking the network.

The very first step for the congestion control process is TCP slow start. The amount of data a sender can transmit (known as the congestion window) with the amount of data the receiver can accept (known as the receiver window) are kept in balance. The maximum amount of data that the sender is allowed to transmit before receiving an acknowledgment from the receiver is the minimum of two values (Congestion window and Receiver Window).

Slow start phase, congestion avoidance phase and congestion detection phase are the congestion policies in TCP. In slow start phase the number packets that are sent from congestion window increment exponentially to the threshold. In this phase after every RTT the congestion window size increments exponentially. The size of the congestion window will be as 2 ^ n where initially n = 0. [4]

Initially cwnd = 1

After 1st RTT, cwnd = 2 ^ (1) = 2
2nd RTT, cwnd = 2 ^ (2) = 4
3rd RTT, cwnd = 2 ^ (3) = 8

In the congestion avoidance phase after reaching the threshold the congestion window size increments by one. This phase starts after the threshold value denoted as ssthresh (When a loss occurs, fast retransmit is sent, it is half of the current CWND) is reached. The size of the congestion window increases additively. After every RTT cwnd will increase by one (cwnd = cwnd + 1).

Initially cwnd = p

After 1st RTT, cwnd = p + 1
2nd RTT, cwnd = p + 2
3rd RTT, cwnd = p + 3

In congestion detection phase the sender goes back to the slow start phase or congestion avoidance phase.

The congestion window size decreases if congestion occurs. The only way a sender can recognise that congestion has occurred is when a segment is to be retransmitted. In order to recover a missing packet which is assumed to have been dropped by a router due to congestion retransmission of the packet is carried out. Retransmission due to timeout and retransmission due to three acknowledgment duplicates are two causes when a retransmission occurs.

Section III: Proposed Algorithm

The algorithm instituted here suggests to first calculate and send the data packets of the size of half of the available channel capacity. Check if the receiver accepts the packets and iterate over next few transmissions but greater or smaller sizes decided as per the algorithm explained in the latter part of the section

Here, the channel is a medium of communication, the path that data takes from source to destination and the "capacity" of a channel is the theoretical upper limit to the bit rate over a given channel. It can be derived by Shannon's channel capacity [5] which is an equation that determines the information capacity of a channel from a few physical characteristics of the channel. A communication system can try to cross the Shannon's capacity limit for a given channel, but numerous errors will be encountered in transmission, and the expense to fix those errors is generally not worth. Shannon's capacity, therefore, is a maximum bit rate below which information can be transmitted with the least number of errors.

The Shannon channel capacity *C* (unit: bits/sec) is given by the equation:

$C=W*\log_2(1+SNR)$

Where C is the maximum capacity of the channel, W is the available bandwidth in the channel, and SNR is the signal to noise ratio, not in DB.

considering receiving window as R. The maximum size of the data that can be accepted by the receiver at a time over the channel.

And let's consider the packet size which is to be sent over the network as P;
and the initial value of P as (0+C)/2
Total Data size to be sent, D
and L is the lower value of the last calculated average for data size.

```
While (D>0)
{

        If (P < R && Ack == 1)
                D = D - P;
                P= (P+C)/2;

        else if (P == R && Ack == 1)
                P=R
                D = D - P

        else
                P= (L+P)/2;
}
```

Here considering the acknowledge variable ack as 1 in case of successful transaction and 0 as unsuccessful.

Briefing the above snippet, initially the data with size of average between minimum acceptable data size (considering 0 here) and maximum channel capacity is sent, now if the receiver responds with positive acknowledgment then average of the size of previously accepted data packet and maximum capacity is taken the data with that size is forwarded and this iterations keep on going; now if the receiver responds with rejection than for initial case then average of minimum size (i.e. 0) and the rejected value is taken. Similar steps are taken in case of other rejections, the average of lower value of the previous step and the rejected value is calculated and that amount of data is transferred in the next try.

Section IV: Comparison

The proposed algorithm of modified slow start is compared with the basic original version of slow start considering an example as given below.

A data of 1Mb is to be sent from System A to System B (receiver), the Channel capacity is considered as 128 bits, System B's receiving window size is 100 bits, considering the Round trip time (RTT) as 2 microseconds following are calculations for both versions of the algorithm. For 1MB data, the time difference is 18 units in the positive side of the modified TCP slow start algorithm.

Basic Version:

Data sent	Data remaining	status	RTT
1	1023	accepted	2
2	1021	accepted	2
4	1017	accepted	2
8	1009	accepted	2
16	993	accepted	2
32	961	accepted	2
64	897	accepted	2
128	897	rejected	2
65	832	accepted	2
66	766	accepted	2
67	699	accepted	2
68	631	accepted	2
69	562	accepted	2
70	492	accepted	2
71	421	accepted	2
72	349	accepted	2
73	276	accepted	2
74	202	accepted	2
75	127	accepted	2
76	51	accepted	2
51	0	accepted	2
Total time taken			42

Proposed Version:

Data sent	Data remaining	status	RTT
64	960	accepted	2
96	864	accepted	2
112	864	rejected	2
104	864	rejected	2
100	764	accepted	2
100	664	accepted	2
100	564	accepted	2
100	464	accepted	2
100	364	accepted	2
100	264	accepted	2
100	164	accepted	2
64	64	accepted	2
Total time taken			24

The line plotted with orange colour represents a conventional TCP slow start algorithm and line plotted with pink represents a modified version of TCP slow start to avoid congestion control. Juxtaposing the time taken to send packets across is much less in the modified algorithm as compared to the standard algorithm.

Conclusion:-

In this paper we have proposed a modified version of slow start algorithm which can efficiently send the data in lesser time as compared to the conventional method of slow start algorithm as it directly sends the data of 50% the size of the channel capacity and then iterates on 50% of the upper or lower half of that 50 based on if its accepted or rejected. This iteration keeps repeating and the maximum acceptable data size is identified quite early as compared to the conventional method and hence the data is transmitted quickly. The claims have been supported by the mathematical proofs and calculations given in section IV where the conventional and proposed algorithm are compared by a case of similar data constraints.

References:-

1. V. Jacobson, "Congestion Avoidance and Control", SIGCOMM Symposium on Communications Architectures and Protocols, pages 314–329, 1988. ,V. Jacobson. "Modified TCP Congestion Avoidance Algorithm", Technical report, 30 Apr. 1990
2. Sally Floyd. RfC 3742: Limited Slow-Start for TCP with Large Congestion Windows, 2004.
3. https://en.wikipedia.org/wiki/Round-trip_delay_time
4. http://eee.guc.edu.eg/Courses/Networks/NETW503%20Internet/Tutorials/Tutorial%203 (Problems).pdf
5. https://en.wikibooks.org/wiki/Communication_Networks/Channels
6. Exploration and evaluation of traditional TCP congestion control techniques by Ghassan A., Abed Mahamod Ismail and KasmiranJumari.