ISSN:2413-6999
Journal of Information, Communication, and Intelligence Systems (JICIS)
Volume 1, Issue 10, December 2015

Enhancement of Congestion Control Mechanism in Transmission Control Protocol over High-Congested Networks

Maaeda Mohsin Rashid

Abstract— **Transmission Control Protocol (TCP) has become the key factor in manipulating the behavior and performance of the networks. The TCP congestion control plays a vital role in controlling the applications that request the services over various networks. Furthermore, the congestion control provides the amount of traffics that can be inserted into the network, where it overrides the behavior and the performance of the communications processes. The TCP congestion control includes two main phases, slow-start and congestion avoidance and these two phases even work separately, but the combination of these two phases will control the congestion window (cwnd) size and control the packet injection to network pipe. This article introduces a novel technique to initiate the congestion window in slow-start phase by duplicating the window size after receiving each acknowledgment (ACK) and then approximates the window size using a quadratic interpolation formula to reach ssthresh.**

Index Terms— TCP, slow-start, congestion control, NS-2.

I. Introduction

The congestion control repeatedly regulates the transmission of packets so that the links of network are completely exploited although they have the limitation of the loss of congestion [1]. These two algorithms are added to most of the TCP variants to solve risks during congestion, the two phases of slow-start and congestion avoidance is merged to work with each other [2].

The major features of the congestion control are as follows: (i) reduce the flow of traffic inflowing the network, (ii)relieve congestion at a certain point in the network and (iii) convey the flows to other paths that are away from the nodes, which suffer from congestion [3]. Generally the congestion happens when a request for resources of the network is greater than the supply; consequently it leads to the drop in several packets, which will end up in retransmission for these packets [4].

The employment of the congestion control scheme of TCP is not ingenuous. If the TCP senders are ready for packet's transmission, the mechanism initiates an upsurge in the size of the congestion window while attempting to increase the rate of transmitted packets. The size of the window is exponentially improved until an assured threshold level slow-start threshold (ssthresh) is achieved, and this step is named as the 'slow-start' phase. In slow-start phase, the TCP sender saves the double size congestion window to increase the transmitted rate for each Round-Trip Time (RTT). When the window value overdoes the level of ssthresh, the TCP moves to the next phase called as "congestion avoidance", and in this phase, the window size increases moderately in a slower rate.

The TCP senders immediately detect the loss in packets by observing the total number of duplicate accumulative acknowledgements, which are received from the destination nodes. If the TCP sender obtains a duplicate acknowledgement for a previously sent packet, it is considered that the packet is lost at the end node. Hence the size of the congestion window is automatically decreased, which eventually decreases the packet's rate, and sets a new level to the ssthresh on the new congestion window value. These stages are called as fast retransmit and fast recovery phases, where the TCP sender arrives at these phases by passing the slow-start phase and congestion avoidance phase.

If the congestion situations are severe, the TCP sender misses the ability to detect the packet loss. In such conditions, the TCP depends on other techniques such as 'retransmission timeout', which produces retransmission for the packets which are lost. It is worth mentioning that all the experiments, modeling, simulation, and evaluations of this paper are executed by using Network Simulator 2 (NS-2) [5].

II. New Slow-Start Mechanism

The exponential increasing of TCP slow-start mechanism consumes a lot of time to start up the window.

That causes low exploitation of the available bandwidth in the network path. In addition, in initial slow-start phase, the

Manuscript received July 21, 2015.
Maaeda Mohsin Rashid, College of Medicine, Kirkuk University, (e-mail: maeda_hiderm@yahoo.com). Kirkuk, Iraq.

ISSN: 2413-6999
Journal of Information, Communication, and Intelligence Systems (JICIS)
Volume 1, Issue 10, December 2015

sender nodes are not able to recognize the available network bandwidth because it uses the default values of ssthresh [6].

As discussed earlier in the previous slow-start technique the multiple packets are sent through one sending window, to push the window to grow faster and to control the packet drops by using an interpolation increments to avoid packet loss. In spite of this, the technique compounds more than one packet in the same transmitting window but cannot always perform well due to the radical performance drop of the TCP.

Interpolation, a fundamental topic in the numerical analysis, is the problem of constructing a function which goes through a given set of data points. In some applications, these data points are obtained by sampling a function or process; subsequently, the values of the function can be used to construct an "*Interpolant*", which must agree with the interpolated function at the data points [7]. The "*univariate*" polynomial is the simplest kind of interpolation, in which most of the developments have been made. The multiple formulae for polynomial interpolation have been proposed by a lot of researchers, especially by Newton [8] and Lagrange [9].

Polynomial interpolation involves in finding a polynomial of order n that passes through the $n+1$ data points. One of the methods used to find this polynomial is called as the Lagrange Interpolating Polynomial [10]. Other methods include Newton's divided difference polynomial method and the direct method. The idea of interpolation is to find a function of a specified from, which passes through a given list of points. As shown in Figure 4, the Lagrange's interpolation uses polynomials, and when three points, (x_1,y_1), (x_2,y_2), and (x_3,y_3), with different x-coordinates, either the three points lie in a line, or else there is exactly one parabola (second degree polynomial $y = ax^2 + bx + c$, second order polynomial interpolation also called Quadratic Interpolation) through the three points.

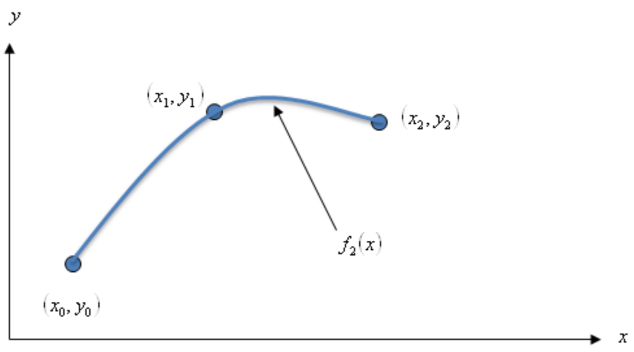

Fig. 1. Quadratic interpolation

Equation (1) represents a polynomial of degree at most three, which passes through the points $(x_0, y_0), (x_1, y_1)$, and (x_2, y_2). When constructing interpolating polynomials, there is a tradeoff between having a better fit and having a smooth well-behaved fitting function. The more the data points used in the interpolation, the higher the degree of the resulting polynomial, and therefore, the greater oscillation will be exhibited among the data points.

$$f(x) = y_1 * f_1(x) + y_2 * f_2(x) + y_3 * f_3(x) \quad (1)$$

Applying this formula in slow-start algorithm needs to assume the three points to get the next value of the congestion window by interpolating the behavior of the window. To simplify the required relationship between cwnd and ssthresh, equation (1) can be written in another form:

$$y = a_1 + a_2 x + a_3 x^2 \quad (2)$$

where *y* is the value of cwnd after one RTT and *x* represents the current cwnd. Then:

$$cwnd(t+\tau) \rightarrow y$$
$$cwnd(t) \rightarrow x$$

The coefficient a_1 neglected to avoid the shift in cwnd level while the other coefficient, a_2 and a_3 can be calculated using the relation between the value of ssthresh and the time differences for RTT period and then apply this parameters to estimate the coefficient, a_2 and a_3 using the equations (3) as shown below:

$$cwnd(t+\tau) = a_1 + a_2 * cwnd(t) + a_3 * (cwnd(t))^2 \quad (3)$$

Therefore, the final formula will be:

$$cwnd(t+\tau) = 2cwnd(t) - \frac{1}{ssthresh}(cwnd(t))^2 \quad (4)$$

Equation (4) shows the required formula to increase the congestion window size when the double size of cwnd becomes larger than the half of the ssthresh value.

The purpose of using of Lagrange's interpolation polynomial is to accurately increase the congestion window, when the duplicating widow approach is used. This algorithm

ISSN:2413-6999
Journal of Information, Communication, and Intelligence Systems (JICIS)
Volume 1, Issue 10, December 2015

can enhance the growth of the congestion window for TCP sender, because it supports to duplicate the congestion window for each RTT until the value of the congestion window is larger than ssthresh/2, where at this point the proposed algorithm enters a new mode by increasing the congestion window in small and precise steps calculated, using quadratic interpolation based on Lagrange formula.

III. EVALUATION AND RESULT'S ANALYSIS

The new slow-start mechanism is applied over TCP Tahoe by removing the previous standard algorithm with required modification to the main file (tcp.cc) using NS-2, where this file is responsible to control the slow-start and some other features of congestion control. The pseudo code of the proposed algorithm arranged as shown below:

```
        /* Slow-Start phase*/

if (cwnd_ < ssthresh_/2){

       /* Duplicating cwnd */

     cwnd_ = 2*cwnd_ ;
          }

if (cwnd_ > (ssthresh_/2) && cwnd_ <
ssthresh_){

        /* Quadratic interpolation */

 cwnd_=
((2*ssthresh_*cwnd_)-(cwnd_*cwnd_))/ssth
resh_ ;
           }
```

Figure 2 illustrates the standard and improved slow-start algorithm.

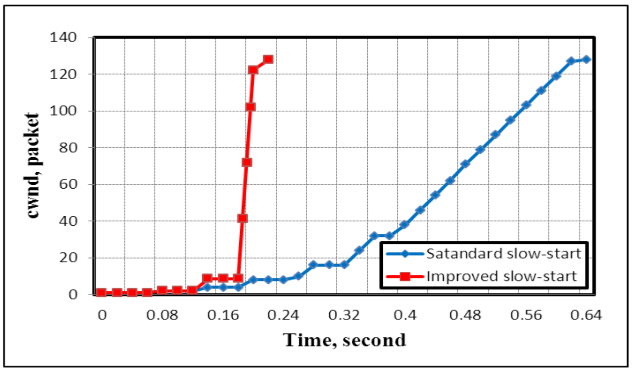

Fig. 2. Initial slow-start phases for standard and improved TCP

The standard growth is needed to 0.621 sec to complete the initial slow-start phase and touch the limit of ssthresh (128 packets), while the improved TCP reaches the ssthresh level within 0.203 sec, which demonstrates that the new slow-start scheme is faster than the typical scheme for a reasonable time period.

The effect of the proposed mechanism will appear with large bandwidth network or when the size of the window is large, so the proposed technique permits the congestion window to increase in a short period and start the next phase, while the original algorithm with exponential increasing consumes a lot of time to reach the same level, because it is based on single increment.

The congestion window is tested with maximum ssthresh 128 packets as shown in Figure 3, where the two TCP's (standard and improved) are tested, so it is easy to detect the wide gap in time between the standard and improved TCPs. The first congestion avoidance phase of the standard TCP is completed after 8.2 sec, while the improved TCP requires only 7.3 sec to complete first congestion interval. The fast growth in the window is performed by new slow-start mechanism, which allows the TCP sender to send a larger amount of packets exceeding the ability of the standard TCP. The alterations in performance between two approaches (standard and improved) should be increased when the network path is still full and when the TCP sender tries to inject new packets through the network pipe.

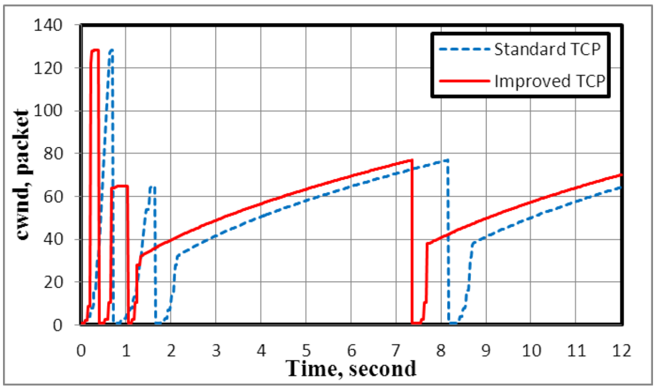

Fig. 3. Behavior comparisons of cwnd for standard and improved TCP

Other experiments are used to test the new and standard slow-start mechanism with early loss events in the network, with 20 packets as maximum window and 1460 bytes as a packet size. As shown in Figure 4, the congestion window of new congestion control still performs well with loss and congested network path, where the fast increase in the window as the fast slow-start algorithm forces the congestion window to reach a reasonable level as clear as in 2 seconds.

The new window reaches 10 packets, while the standard TCP stops with six packets only despite the two slow-starts are begun in same time.

ISSN: 2413-6999
Journal of Information, Communication, and Intelligence Systems (JICIS)
Volume 1, Issue 10, December 2015

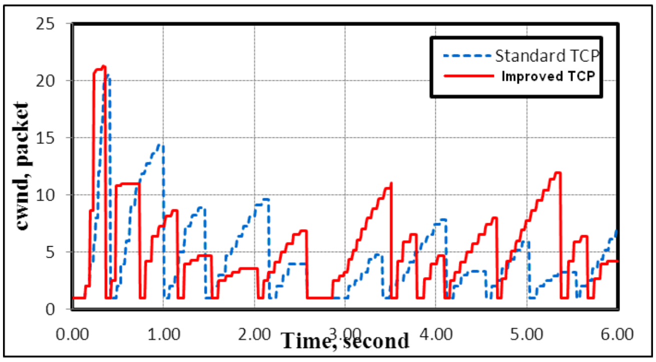

Fig. 4. Congestion window comparisons with congested bottleneck

The fast window growth helps the TCP sender to send more packets because the new slow-start scheme needs a shorter period to send packet and to receive the acknowledgment and even at the time of loss there is enough time (compared with a standard slow-start algorithms) to retransmit the lost packet again.

IV. Conclusion

In this paper, a new approach had been proposed to improve the congestion window initialization in slow-start phase. The new technique is based on duplicating the congestion window per ACK and then approximates the size according to Lagrange's interpolation formula to avoid the window size from exceeding the ssthresh. The proposed approach minimizes the required time to reach the ssthresh and also delays the congestion point of the network connection; moreover, the number of transmitted packets will increase due to the fast clocking of the congestion window. The fast increment in TCP slow-start provides high throughput over the network path when the RTT becomes small, it is effectual to inject a large amount of packets in network connection from the TCP sender to TCP receiver, instead of wasting a lot of time to start up the window with the exponential scheme. It can be concluded that the new mechanism achieves the requirements of high congested connections and the challenging network by enabling the congestion window to grow faster.

References

[1] S. Chen, B. Mulgrew, and P. M. Grant, “A clustering technique for digital communications channel equalization using radial basis function networks,” *IEEE Trans. on Neural Networks*, vol. 4, pp. 570-578, July 1993.

[2] J. U. Duncombe, “Infrared navigation—Part I: An assessment of feasibility,” *IEEE Trans. Electron Devices*, vol. ED-11, pp. 34-39, Jan. 1959.

[3] C. Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, and M. Miller, “Rotation, scale, and translation resilient public watermarking for images,” *IEEE Trans. Image Process.*, vol. 10, no. 5, pp. 767-782, May 2001.

[4] H. M. Shirazi, "Smart Congestion Control in TCP/IP Networks," *Journal of Information and Communication Technology*, vol. 2(2), pp. 73-78, 2009.

[5] J. Mo, R. J. La, V. Anantharam, and J. Walrand,"Analysis and comparison of TCP Reno and Vegas," *in INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 1999, pp. 1556-1563 vol.3.

[6] P. Sarolahti, "Congestion Control in Linux TCP," *in USENIX 2002 Annual Technical Conference*, 2002, pp. 49-62.

[7] L. Yuan-Cheng and Y. Chang-Li, "TCP congestion control algorithms and a performance comparison," in Computer Communications and Networks, 2001. *Proceedings. Tenth International Conference on*, 2001, pp. 523-526.

[8] S. Fahmy and T. P. Karwa, "TCP CONGESTION CONTROL: OVERVIEW AND SURVEY OF ONGOING RESEARCH," *CSD-TR-01-016*, 2001.

[9] J. F. Kurose and K. W. Ross, Computer Networking-Complete Package, THIRD EDITION ed., 2006.

[10] M. Kodama, G. Hasegawa, and M. Murata, “Implementation experiments of TCP Symbiosis: bio-inspired mechanisms for Internet congestion control,” *in Proceedings of CQR 2008*, Apr. 2008.

[11] G. A. Abed, M. Ismail, And K. Jumari, "Architecture And Functional Structure Of Transmission Control Protocol Over Various Networks Applications," Journal Of Theoretical And Applied Information Technology, Vol. 34, 2011.

[12] G. A. Abed, M. Ismail, And K. Jumari, "The Evolution To 4g Cellular Systems: Architecture And Key Features Of LTEAdvanced Networks," Spectrum, Vol. 2, 2012.