ISSN:2413-6999
Journal of Information, Communication, and Intelligence Systems (JICIS)
Volume 2, Issue 1, February 2016

Effective Approach to Approximating TCP Window Size in High Latency Traffics

Marwah K. Farhan

***Abstract*— Slow-start mechanism in Transmission Control Protocol (TCP) represents the prime technique for usage the operative network resource. The objective of the slow-start mechanism is to create a TCP connection in an optimal state - run for network route rapidly without giving a disproportionate load to the network. Due to the rate of packet transferred is limited by the bandwidth of the narrowest bottleneck alongside the connection pipe, then, the transferring rate cannot exceed this bandwidth and the maximum rate competitions the permissible bandwidth of the link that own narrowest bandwidth over end-to-end connections. In large bandwidth networks, the mechanism of slow-start does not perform suitably and may cause considerable degradation in network performance. Therefore, it's necessary to develop a new slow-start technique that can increase the packet transferring over large bandwidth links environments. This article investigates the behavior of TCP protocol after applying a technique based on using available bandwidth estimation to detect the current level of the network slow-start threshold (ssthresh) in high latency links.**

Index Terms— TCP, ssthresh, congestion window, packet-pair.

I. Introduction

Large-bandwidth and high-latency links such as satellite systems and some networks that used in long range links, the rate of packet transferred is limited by the bandwidth of the narrowest bottleneck alongside the connection pipe, then, the transferring rate cannot exceed this bandwidth and the maximum rate competitions the permissible bandwidth of the link that own narrowest bandwidth over end-to-end connections [1]. Some approaches that are used to estimate the available bandwidth based on deriving an algorithm to obtain the (joint) distribution of the separation between the probes at the destination(s) for a given distribution of the spacing at the input [2]. While some other approaches went to propose lightweight countermeasures to detect attacks on bandwidth measurements.

Marwah K. Farhan, Ministry of Higher Education and Scientific Research, *(e-mail: marwah_farhan@yahoo.com). Baghdad, Iraq.*

This technique could detect whether delays were inserted within the transmission of a packet-pair [3]. The other problem with TCP over a large bandwidth link that when packets are transmitted, the link enters to slow link and the packet period is extended. That means, if TCP source sends a packet of 1024 Byte, this will need to 0.08 seconds over 1 Mbps bandwidth, while over 100 Kbps it needs to 0.8 second. In other hands, the segment acknowledgment is used as a trigger for transferring and the arriving period of the acknowledgment equal to the period of the source's segments that was extended with the bottleneck. So when acknowledgement segments are used to initiate the transmitting segment, the transmitting period will equal to the bottleneck bandwidth and will come to be optimized for connection path.

II. Packet-Pair Algorithm

In the evaluation of TCP slow-start behavior over large bandwidth networks and experimenting the performance, we used the network topology shown in Fig.1 and using network simulation 2 (NS-2) [4]. This figure demonstrates the proposed topology to explain the performance variation of slow-start mechanism with two different bandwidths. It consists of node S as a TCP source which is connected to the corresponding node D as a TCP destination with 10 Mbps as a bandwidth and 5 msec as a propagation delay. The two nodes, S and D are connected through bottleneck of 1Mbps bandwidth and 10msec latency.

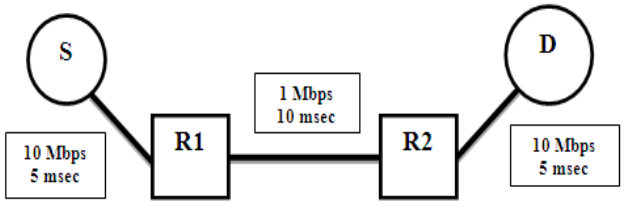

Fig.1 Network Topology

The TCP used over this simulation is Reno, where this TCP can perform better than other TCP variants with stable congestion window. If the bottleneck bandwidth is set to 1

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

Mbps, the cwnd exponentially startup within the slow - start mechanism with an interval equal to 0.25 msec and when the loss in the packet is happening after 0.7 msec, the sound is entered the congestion avoidance mode which increased linearly as shown in Fig. 2.

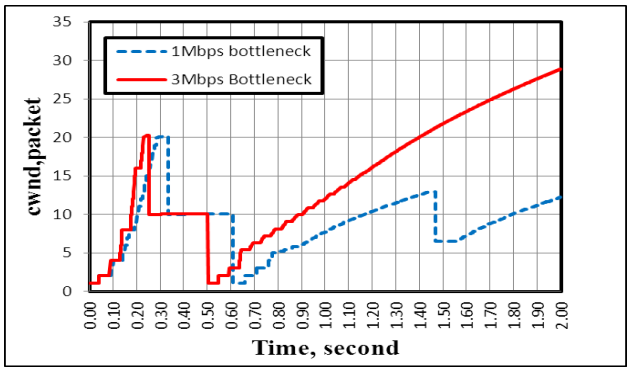

Fig. 2. Slow-Start comparison with two different bandwidths

Estimation of TCP slow-start performance over 1 Mbps bottleneck based on the window and Round-Trip Time (RTT) values is shown in Eq. 1:

$$Throughput(Byte/\sec) = \left[\frac{Windowsize(Byte)}{RTT(\sec)}\right] \quad (1)$$

As a result, the optimal window size for definite bandwidth is estimated using Eq. 2:

$$Window\_size(Byte) = Bandwidth(Byte/\sec) * RTT(\sec) \quad (2)$$

When RTT is 0.14 Sec, then the window size can be calculated from:

$$Window\_size = 1Mbps * 0.14\sec$$

$$= \frac{1024000(bit/\sec)}{8(bit/byte)} * 0.14(\sec)$$

$$= 17920(Byte)$$

$$optimal\_window\_size = 17.5(Packet)$$

Then, in slow-start phase, the optimal window size is 17920 Bytes (17.5 packets) and the maximum window size reaches to 20480 Bytes (20 packets) after 0.3 sec. When the bandwidth of the bottleneck is changed to be equal 3 Mbps instead of the previous bandwidth (1 Mbps), the RTT value of the network path is increased and the optimal window size becomes very large while the maximum window size of initial slow-start sets to be equal 20 packet only. That means the difference between the maximum and optimal window is very far and that leads to make the congestion avoidance phase to take a long period and that causes degradation in network performance.

Generally, the limited performance of TCP slow-start is because the constant parameters used to set the window size with network action, and for this reason, the proposed slow-start algorithm support an adaptive behavior to rapidly adjust the window size.

Packet-Pair proposed by Melander et al. [5] provide accurate measurement to the available bandwidth of a network path by sending many pocket pairs at regularly increasing rates from the sender to the destination. Packet-Pair approach is built on the supposition that when two packets are transmitted with narrowly spaced back-to-back timing, their inter-arrival time at the destination straightly reflects the bandwidth of the bottleneck bandwidth over the network path. Also, if the path is un-congested, the equivalent ACKs are received at the TCP source with the same time spacing.

Therefore, the TCP source can estimate the bottleneck bandwidth by dividing the length of the transmitted packets by the inter-arrival time between the equivalent ACKs [6]. The proposed technique aims to estimate the available network bandwidth as well as the ssthresh value using the essential data amount per RTT period as shown in Eq. 3:

$$ssthresh = \frac{EBW * RTT_{\min}}{MSS} \quad (3)$$

Here, ssthresh is valued via estimate the available bandwidth (EBW), minimum RTT (RTTmin), and the Maximum Segment Size (MSS). The new streets should be used to achieve an adaptive slow-start algorithm with updated bandwidth where this technique pushes the congestion control to send packets over real network capacity. In Eq. 3, MSS and RTTmin are representing fixed parameters while EBW values need to estimate rapidly to find the optimum ssthresh values.

Fig. 3 specifies two packets with same size sending from TCP source to its destination where the wide portion of the

ISSN:2413-6999
Journal of Information, Communication, and Intelligence Systems (JICIS)
Volume 2, Issue 1, February 2016

link pipe denotes a large bandwidth link while the thin portion denotes a small bandwidth link. The space between these two packets is due to the queuing at the bottleneck keeps fixed downstream since there is no extra downstream queuing.

In addition, the two assumed packets should have the same size because dissimilar size packets have different velocities. So, when the first packet is greater than the second, then the second packet transmission delays and continuously less than the first packet. Correspondingly, if the second packet is greater, then it will be later than the first packet [7].

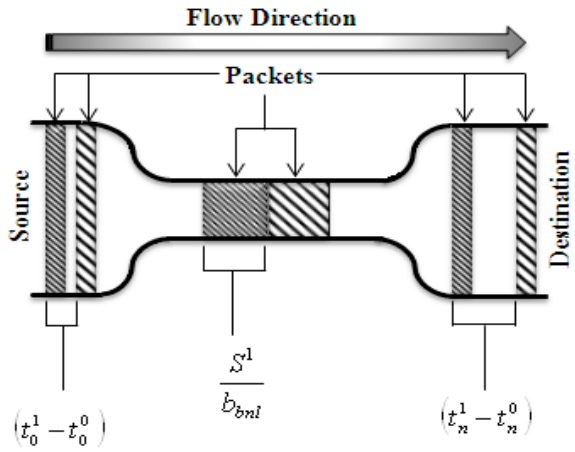

Fig. 3 Packet-Pair queuing

The property of Packet-Pair algorithm that it expects the change in the arrival period of the two packets which have the same size that sending from the same source to the same destination:

$$t_n^1 - t_n^0 = \max\left(\frac{S_1}{b_{bnl}}, t_0^1 - t_0^0\right) \quad (4)$$

Where tn1 and t_n^0 are the inter-arrival times period of the first and second packets one by one at the receiver, t_0^0 and t_0^1 are the sending times of these packets respectively, S_1 is the size of the second packet, and bb_{nl} is the bandwidth of the bottleneck link [8]. The instinctive validation for Eq. 4 is when two packets are transmitted nearby enough together in sending time to causing the both packets to queue at the bottleneck pipe:

$$\left(t_0^1 - t_0^0 < S_1 / b_{bnl}\right) \quad (5)$$

Then, the packets will reach at the receiver with the same time spacing t_{n1} - t_{n0} as when they departed the bottleneck pipe S_1/bb_{nl}. The time spacing will keep the same since the packets are the same size and no downstream of link of the bottleneck pipe has a lesser bandwidth than the bottleneck as shown in Fig. 3.

To solve Eq. 4, since Packet-Pair algorithm depend on the point that if two packets are queued following to each other at the bottleneck pipe, so they will departure the link t seconds away from each other as follows [9]:

$$t = \frac{S_2}{b_{bnl}} \quad (6)$$

Supposing the time separation for the bottleneck is constant, and then the two packets will reach at the destination with t seconds away from each other. So the estimation of bottleneck bandwidth where S_2 is known:

$$b_{bnl} = \frac{S_2}{t} \quad (7)$$

The packet - Pair algorithm creates some suppositions that could not hold in practice. For example, it is incredible to assure that the two packets are queue each other at the bottleneck. In other hands, when new packets queue between the two measured packets, Eq. 7 becomes:

$$b_{bnl} = \frac{S_2 + S_0}{t} \quad (8)$$

Also, by using the property of Packet-Pair, Eq. 4 can solve for bb_{nl} to obtain the received bandwidth:

$$b_{bnl} = \frac{S_1}{t_n^1 - t_n^0} \quad (9)$$

The value of bb_{nl} represents the received bandwidth since it is measured at the destination. Furthermore, when another technique called Filtering used with Packet-Pair, the bandwidth estimated at the source is:

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

$$b_{bnl} = \frac{S_1}{t_0^1 - t_0^0} \qquad (10)$$

III. Results and Discussions

TCP Reno is used to test the new slow-start technique due to Reno congestion control mechanism w which support fast retransmit and fast recovery technique to cover the packet loss and for that, the effect of the proposed technique appears clearly with Reno. Fig. 4 illustrates the behavior of congestion window before and after supporting the proposed mechanism. The period (35 sec) is enough to prove the effect of using the new approach on the congestion window behavior, where the gap between standard and new continues to increase with the simulation session because of the collected time variation obtained from every slow-start phase.

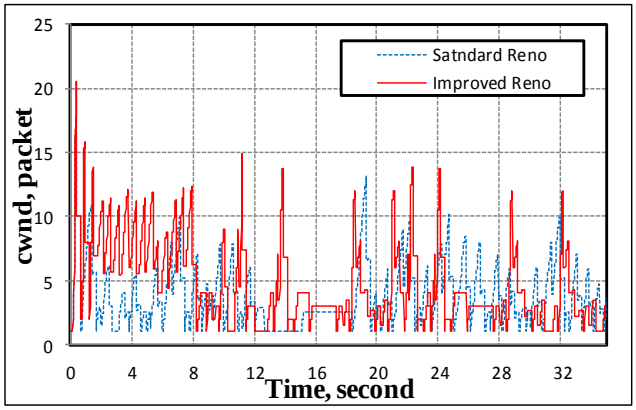

Fig. 4 Congestion window comparison between standard and improved TCP Reno

The efficient behavior of congestion window for the improved Reno in Fig. 4 is clear and the high beam which appeared in improved Reno was due to the estimation to the available bandwidth to upgrade the stress during the simulation period. The usage of the available bandwidth can push the congestion control algorithm to inject a suitable amount of packets in network pipe and avoid exceeding the limits to avoid the packet dropping. This operation will clearly affect the throughput of the network as shown in Fig 5. In this figure, the rational throughput is calculated to show the real performance of the proposed technique.

The adaptive behavior of the new slow-start gave high throughput compared with standard slow-start mechanism and the throughput is duplicated and kept a stable goodput within network simulation scenario.

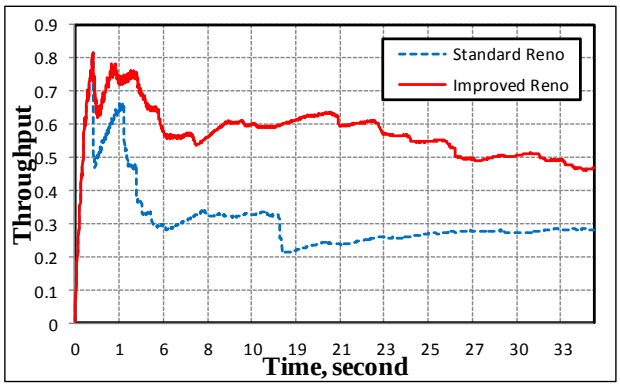

Fig. 5 Congestion window comparison between standard and improved TCP Reno

In fact, the effect of the proposed technique to estimate the bandwidth and upgrade the ssthresh is not showed large difference compared with standard Reno, but the real effect appeared in throughput showed in Fig. 5. Additionally, the expected packet loss with this technique expected to be less than the high loss with standard Reno, due to it know that Reno does not perform well over high congested networks and with a high loss rate level.

Actually, the packet-pair technique can serve the network with large propagation delay and long RTT, but it suffers from serious problems when it applies over low-latency networks, due to the packet-pair requires suitable time difference to detect the bandwidth precisely. Also, when the RTT becomes short, the estimation error reaches to unacceptable levels so this technique becomes insufficient when it applied.

IV. Conclusion

In this article, the authors focused on developing a new slow-start mechanism based on estimation of the available bandwidth of the network pipe to upgrade ssthresh value especially in large-bandwidth links such as industrial application and satellite systems. The proposed solution used packet-pair technique to estimate the bandwidth and employing other factors to calculate the ssthresh. The proposed slow-start algorithm grounded on the upgraded ssthresh where this threshold reflects the available capacity of the network pipe during estimation operation. The evaluation of the proposed technique used TCP Reno over high congested network topology with wired-wireless nodes. The experimentation proved that the improved slow-start performed better than the standard algorithm and the throughput is duplicated with the new algorithm compared with the traditional Reno version.

ISSN:2413-6999
Journal of Information, Communication, and Intelligence Systems (JICIS)
Volume 2, Issue 1, February 2016

References

[1] Y. Nishida, "Smooth slow-start: refining TCP slow-start for large-bandwidth with long-delay networks," 1998, pp. 52-60.

[2] B. Kumar, D. Manjunath, and S. Chakraborty, “Estimating network link characteristics using packet-pair dispersion: A discrete-time queueing theoretic analysis”, Computer Networks, Volume 55, Issue 5, 2011, pp 1052-1068.

[3] G. Karame, B. Danev, C. Bannwart, and S. Capkun, "On the Security of End-to-End Measurements Based on Packet-Pair Dispersions," Information Forensics and Security, IEEE Transactions on , vol.8, 2013, pp.149- 162.

[4] N. Simulator, "ns-2," ed, 1989.

[5] B. Melander, et al., "A new end-to-end probing and analysis method for estimating bandwidth bottlenecks," 2000, pp. 415-420 vol. 1.

[6] A. Capone, "Bandwidth Estimation Schemes for TCP."

[7] K. Lai and M. Baker, "Measuring link bandwidths using a deterministic model of packet delay," 2000, pp. 283-294.

[8] I. Kevin, "Measuring the Bandwidth of Packet Switched Networks," Citeseer, 2002.

[9] K. Lai and M. Baker, "Measuring bandwidth," 1999, pp. 235-245 vol. 1.