International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

An Extremely Elevated Velocity Lan /Wan Using A New Blocking Control Algorithm

M.Laskhmi & S.Amutha
Assistant – professor Department of computer science Bharathidasan University College Of Arts And Science College For Women Orthanadu, Thanjavur, Tamil Nadu
Assistant – professor Department of computer science Bharathidasan University College Of Arts And Science College For Women Orthanadu, Thanjavur, Tamil Nadu

Abstract

In a percent more applications require fast transfer of massive data over networks, and the emergency of high-speed networks provides an ideal solution to this challenge. Due to the limitations of the conservative congestion control algorithm, the standard TCP is no longer appropriate for high- speed networks to efficiently utilize the bandwidth resources. A new congestion control mechanism for high-speed networks is introduced. It uses packet loss information to determine whether the window size should be increased or decreased, and uses queuing delay information to determine the amount of increment or decrement.

Introduction

The maladies of congestion collapse from undelivered packets and of unfair bandwidth allocations have not gone unrecognized. Some have argued that there are social incentives for multimedia applications to be friendly to the network, since an application would not want to be held responsible for throughput degradation in the Internet. However, malicious denial-of-service attacks using unresponsive UDP flows are becoming disturbingly frequent in the Internet and they are an example that the Internet cannot rely solely on social incentives to control congestion or to operate fairly. Some have argued that these

maladies may be mitigated through the use of improved packet scheduling [1] TCP-New Reno [2], and SACK TCP [3] are the standard versions of TCP congestion control protocols currently deployed in the Internet, and they have achieved great success in performing congestion avoidance and control.

The key feature of standard TCP is its congestion avoidance phase, which uses the additive increment multiplicative decrement (AIMD) algorithm [4]. Being a window-based algorithm, TCP controls the send rate by maintaining a window size variable W, which limits the number of unacknowledged packets in the network from a single user. This window size is adjusted by the AIMD algorithm in the following manner: W is increased by α /W (α = 1 for standard setting) for each ACK, and thus is increased by a constant α/b per round trip time (RTT) if all the packets are acknowledged within an RTT, where b is the number of packets acknowledged by each ACK (b = 1for original

TCP, and b = 2 for delayed ACK [7]). On the other hand, W is decreased by a fixed proportion βW (β = 1/2 for standard setting) once some packets are detected to be lost in the last RTT. Under this algorithm, senders gently probe the network for spare bandwidth by cautiously increasing their send rates, and sharply reduce their send rates when congestion is detected. Along with other features like slow start, fast recovery, and fast retransmission, TCP achieves congestion control successfully in the current low-speed networks [5-6].

The rapid evolution of high-speed networks is significantly supporting the international collaborations with massive data transfer and computing resource sharing, and the networks integrated with 1–10 Gbps bandwidths have been developed and deployed over numbers of research institutions. In order to efficiently utilize the large bandwidths at the physical layer, researchers have focused on the developments of protocols at transport and network layers.

The standard TCP has been remarkably successful in performing congestion avoidance and control to prevent severe congestion in the current low-speed networks. However, it is well-known that the standard TCP is not appropriate for high-speed networks in terms of the additive increment multiplicative decrement (AIMD) algorithm is too conservative to rapidly achieve full bandwidth utilization while is too drastic to recover from per packet loss event. In order to conquer the poor performance problem, the standard TCP together with the AIMD algorithm should be modified in high-speed networks. So far, a number of high-speed TCP variants have been proposed, including the end-to-end approaches, e.g. High Speed TCP (HSTCP), Scalable TCP (STCP), CUBIC TCP, FAST TCP, Compound TCP (CTCP), TCP-Illinois and the router-based approaches, e.g. XCP (8), VCP (9).

In addition, some researches focus on the application-level schemes on top of UDP to realize the congestion control functions for high-speed net- works, such as UDT (10). Although these approaches achieve higher throughput over the standard TCP in high-speed networks, most of them also have shortcomings in various aspects such as fairness, TCP-friendly, responsiveness, robustness, etc. Since none of the existing approaches is overwhelmingly better than the other protocols and has the convincing evidence that could be generally deployed, the development of new high-speed TCP variants is still needed. In this paper, a new congestion control algorithm for high-speed networks.

It uses packet loss information to determine whether the window size should be increased or decreased, and uses queuing delay information to determine the amount of increment or decrement. A new congestion control TCP, a new congestion control algorithm using the delay-based and loss-based approach for the adaptation to high speed and long distance network environment. The algorithm uses queuing delay as the primary congestion indicator, and adjusts the window to stabilize around the size which can achieve the full utilization of available bandwidth. On the other hand, it uses packet loss as the second congestion indicator, and a loss-based congestion control strategy is utilized to maintain high bandwidth utilization in the cases that the delay-based strategy performs inefficiently in the networks. The two approaches in the algorithm are dynamically transferred into each other according to the network status.

The protocol utilizes the delay information as the primary congestion indicator and utilizes the loss information as the second congestion indicator to jointly adjust the window size so as to satisfy the design requirements on efficiency, fairness, TCP-friendliness and robust, and outperforms the standard TCP and other TCP variants in high-speed networks. Due to the delay-based strategy and loss-based strategy, new Congestion Control TCP is a hybrid scheme of congestion control. Finally perform simulations to verify the properties of the proposed new congestion control TCP. The simulation results demonstrate new congestion control TCP satisfies the requirements for an ideal TCP variant in high-speed networks, and achieves efficient performance on throughput, fairness, TCP-friendliness, robustness, etc.

As the aforementioned content, numbers of new protocols have been developed to replace the standard TCP and achieve efficient bandwidth utilization in high-speed networks. The router-based protocols, such as XCP and VCP, require the explicit feedback information from routers to guide their control strategies [11]. However, it is impractical to modify all the existing routers in a real world. Therefore, a majority of the existing protocols focus on the end-to-end method rather than the router-based method for the performance improvement of high-speed networks [12].The end-to-end protocols can be mainly classified into two categories: loss-based congestion control algorithms, e.g. HSTCP, STCP, HTCP, BIC TCP, CUBIC TCP, etc. and delay-based congestion control algorithms such as FAST TCP. The loss-based congestion control algorithms utilize packet loss as the congestion measure, the window size increases for each ACK and decreases per packet loss. HSTCP and STCP are the early works along the loss-based methods.

To quickly catch up the available bandwidth, HSTCP uses step-wise functions for the increase and decrease of window size while STCP sets the increasing and decreasing values proportional to the current window size. However, both the two protocols have a serious problem on RTT-fairness performance. Using these protocols, as the multiple flows competing for the bottleneck bandwidth have different RTT delays, the fair utilization of the bandwidth cannot be achieved. HTCP sets a function of the elapsed time since last packet loss for increase parameter and uses an adaptive back off strategy at congestion events so as to achieve a perfect performance on responsiveness and efficiency in high-speed networks. For the above three protocols, the increment of window size is still fast even the network is close to the congestion event, thus the congestion in network will be caused more easily among the competing flows and result in the degradation of throughput. BIC TCP is an effective protocol that has drawn much attention in research area.

The protocol adjusts the window size using a binary search method to reach a reference value. When updating the window size, it sets the reference value as the midpoint between the maximum reference value WMAX and the minimum reference value WMIN. If the length between Wmin and the midpoint is larger than a maximum value SMAX, the window size increases linearly by the value Smax. Hence, the increment of the window size is linear at the initial stage and then becomes logarithmic when approaching to the reference point. BIC TCP performs better than the earlier approaches. However, it also suffers the RTT unfairness problem. Subsequently, an enhanced version, CUBIC TCP, is developed to improve the RTT-fairness performance of BIC TCP. On the other hand, fundamentally different from loss-based congestion control algorithms, delay-based congestion control algorithms use queuing delay as the congestion measure. FAST TCP is a typical delay-based high-speed TCP variant derived from TCP Vegas.

The protocol maintains queue occupancy at routers for a small but not zero value so as to lead the network around full bandwidth utilization and achieve a higher average throughput. On the contrary, the throughput of loss-based algorithms oscillates between full utilization and under utilization in terms of the probing action purposely generates packet losses. In addition, FAST TCP is able to rapidly converge to the equilibrium state and does not suffer the RTT unfairness problem. However, despite the unique advantages mentioned above, it also has some inherent limitations. Since FAST TCP is a delay-based approach and uses the RTTs for congestion measure, its throughput performance is significantly affected by the reverse traffic, and the throughput of the source traffic decreases as the queuing delay increases on the reverse path. Some works have focused on the reverse traffic problem in delay-based congestion control algorithms and uses a variety of schemes that relies on the measurement of one-way delay to conquer this problem. However, these schemes are not designed for high-speed networks. In addition to the reverse traffic problem, FAST TCP requires the buffer size to be larger than the specified value which indicates the total packet amount maintained in routers along the flow's path. Although any of the loss-based and delay-based approaches can achieve higher throughput than the standard TCP. In order to perform more efficiently and effectively in high-speed networks, some approaches, like CTCP and TCP-Illinois, focus on the synergy of loss-based and delay-based approach.

CTCP utilizes loss and delay information as the primary congestion indicators in different stages to determine direction of window size change, and keeps the traditional Slow-Start at the start-up period while uses a delay-based component derived from TCP Vegas in congestion avoidance phase. TCP-Illinois uses loss information as the primary congestion indicator and uses delay information to be the second congestion indicator. During the operation, it utilizes the loss information to determine the direction of window change and the delay information is used for adjusting the pace of window size change. In order to achieve a concave window size curve and a high throughput, TCP-Illinois set two parameters, α and β, in the protocol operation. When network is far from congestion, it set α to be large and β to be small. Otherwise, α is small and β is large when network is close to congestion. These approaches inherit the advantages from both the loss-based and delay-based approaches. However, due to the delay-based components still uses RTTs to measure the congestion, their throughput performance is also affected by the reverse traffic. In (14), it presented an end-to-end Enhanced FAST (EEFAST) congestion control algorithm,

Which dynamically adjusts the window size according to the measurements of one-way delay, to remove the effect of reverse traffic for delay-based congestion measurement in high-speed networks? However, EEFAST is a delay-based congestion control algorithm and suffers the same problem as FAST in which the packet amount maintained in routers should be smaller than the router buffer size.

TCP congestion control has mainly two phases:

Slow Start and Congestion avoidance. A new connection begins in Slow-start, setting its initial cwnd to 1 packet, and increasing it by 1 for every received Acknowledgement (ACK). After cwnd reaches ssthresh, the connection switches to congestion-avoidance where cwnd grows linearly. A variety of methods have been suggested in the literature recently aiming to avoid multiple losses and achieve higher utilization during the startup phase. A larger initial cwnd, roughly 4K bytes, is proposed in.

This could greatly speed up transfers with only a few packets. However, the improvement is still inadequate when BDP is very large and the file to transfer is bigger than just a few packets. Fast start uses cached cwnd and ssthresh in recent connections to reduce the transfer latency. The cached parameters may be too aggressive or too conservative when network conditions change Smooth start has been proposed to slow down cwnd increase when it is close to ssthresh. The assumption here is that default value of ssthresh is often larger than the BDP, which is no longer true in large bandwidth delay networks. Proposes to set the initial ssthresh to the BDP estimated (Packet Network Discovery) has been proposed to derive optimal TCP initial parameters. SPAND needs leaky bucket pacing for outgoing packets, which can be costly and

Problematic in practice.

TCP Vegas detects congestion by comparing the achieved throughput over a cycle of length equal to RTT, to the expected throughput implied by cwnd and base RTT (minimum RTT) at the beginning of a cycle. This method is applied in both Slow-start and Congestion-avoidance phases. During Slow-start phase, a Vegas sender doubles its cwnd only every other RTT, in contrast with Reno's doubling every RTT. A Vegas connection exits slow-start when the difference between achieved and expected throughput exceeds a certain threshold. However, Vegas are not able to achieve high utilization in large Band width delay networks as we will, due to its over-estimation of RTT.

We believe that estimating the eligible sending rate and properly using such estimate are critical to improving bandwidth utilization during Slow-start.TCP Westwood and Eligible Rate Estimation Overview in TCP Westwood (TCPW), the sender continuously monitors ACKs from the receiver and computes its current Eligible Rate Estimate (ERE). ERE relies on an adaptive estimation technique applied to ACK stream. The goal of ERE is to estimate the connection eligible sending rate with the goal of achieving high utilization, without starving other connections. We emphasize that what a connection is eligible for is not the residual bandwidth on the path. The connection is often eligible more than that. For example, if a connection joins two similar connections, already in progress and fully utilizing the path capacity, then the new connection is eligible for a third of the capacity.

Problem Methodology

System Flow diagram are directed graphs in which nodes specify processing activities and arc specify data item transmitted between processing nodes .Data Flow diagrams represent the system between individual items in fig:1,

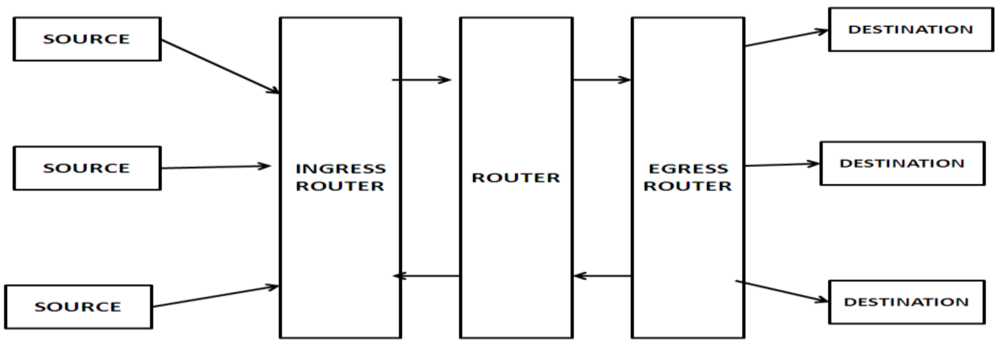

Fig: 1, Backward Feed Back

System implementation

Egress module- Input parameters: (I) Data packets from router. (II)Forward feedback from the router.

Egress module- Output parameters: (I) Data packets. (II)Backward feedback.

Destination module: (I) Message received from the egress router will be stored in the corresponding folder as a text file depends upon the source machine name.

Delay-based congestion control

From the perspective of a delay-based congestion control approach, fig:2 such as FAST TCP, if the queuing delay on the reverse path is heavy, the full utilization of available bandwidth will never be achieved and thus lead to potentially serious degradation of throughput on the forward path. In (13), it presented EEFAST congestion control algorithm to remove the effect of the reverse traffic in high-speed networks. For the design of the delay-based estimation component, the mechanisms of the EEFAST algorithm are used to estimate the congestion in a network. In addition, based on this algorithm, it also adopts a set of new control strategies for adjusting the window size in order to achieve a further performance improvement.

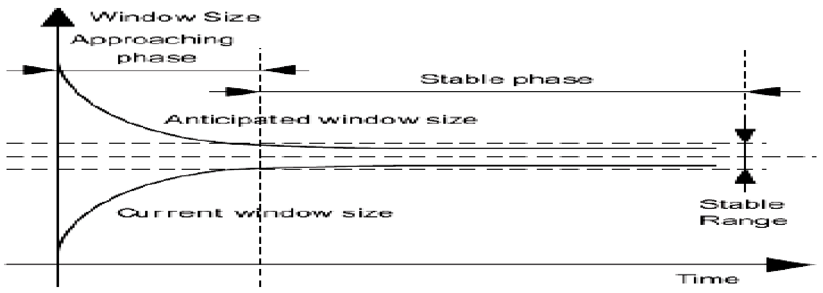

Fig.2.Window size changes by the delay-based strategy.

Loss-based congestion control

Delay-based congestion control algorithms require a specified number of packets queued in routers so as to keep the average throughput around the full utilization. Therefore the buffer size of routers should be larger than the specified value in the delay- based algorithms, and the specified value for a network increases as the increment of source numbers. However, if the buffer size of the routers is not large enough for the specified value, packet loss might happen in the networks. To tackle this, it use packet loss as the second congestion indicator and design a loss-based congestion control strategy for the operation of new Congestion Control TCP. For the loss-based congestion control, when the network is close to the congestion status, the fast increment of window size could lead to the congestion event more easily and cause a heavy oscillation of window size so that degrade the throughput performance for each traffic source.

The linear to logarithmic increase function, as described in (13) and (14), is an efficient way to avoid the heavy congestion induced by fast increment of window size. The approaches increase the window linearly at the initial stage and then increase logarithmically to get close to the reference point that a congestion event may happen. Fundamentally, the change of the window size is from fast to slow. Therefore, using such mechanisms, the traffic source can rapidly catch up the available bandwidth and also prolong the time interval between two successive congestion events so as to achieve better performance on average throughput.

Conclusion

A new congestion control algorithm for high-speed networks. In this paper, we have presented a novel congestion avoidance mechanism for the Network Border Patrol. Unlike existing Internet congestion control approaches, which rely solely on end-to-end control, NBP is able to prevent congestion collapse from undelivered packets. It uses packet loss information to determine whether the window size should be increased or decreased, and uses queuing delay information to determine the amount of increment or decrement. It does this by ensuring at the border of the network that each flow's packets do not enter the network faster than they are able to leave it. NBP requires no modifications to core routers or to end systems. Only edge routers are enhanced so that they can perform the requisite per-flow monitoring, per-flow rate control and feedback exchange operations. it has considered some natural requirements for a new TCP protocol for high-speed networks. A novel congestion control algorithm using the delay-based and loss-based strategies is presented for performance enhancement of data transfer in high-speed networks. The idea is rooted in the following two assumptions or understanding of the entire congestion control system: (i) delay is indeed a useful signal, i.e., congestion or packet loss is indeed correlated to delay information; (ii) delay is not an accurate signal, i.e., the correlation between loss and delay is weak. Combining these two, it should use loss as the primary signal and delay as the secondary signal.

References

[1] B. Suter, T.V. Lakshman, D. Stiliadis, and A. Choudhury, "Design Considerations for Supporting TCP with Per-Flow Queueing," in Proc. Of IEEE Infocom '98, March 1998, pp. 299–305.

[2] S. Floyd, T. Henderson, The new reno modification to TCPs fast recovery algorithm, RFC 2582, 1999.

[3] M. Mathis, J. Mahdavi, S. Floyd, A. Romanow, TCP selective acknowledgement options. (2002)

[4] V. Jacobson, Congestion avoidance and control, ACM Computer Communication Review 18 (August) (1988) 314–329.

[4] V. Jacobson, Berkeley TCP evolution from 4.3-tahoe to 4.3-reno, in: Proceedings of the Eighteenth Internet Engineering Task Force, July1990.

[5] S. Liu, T. Basar, R. Srikant, and TCP-Illinois: A loss and delay-based congestion control algorithm for high-speed networks, in: Proc. First International Conference on

Performance Evaluation Methodologies and Tools, VALUETOOLS, Pisa, Italy, October 2006.

[6] Kolawole I. Oyeyinka1, Ayodeji O. Oluwatope2, Adio. T. Akinwale3, Olusegun Folorunso3, Ganiyu A. Aderounmu2, Olatunde O. Abiona, TCP Window Based Congestion Control Slow-Start Approach, *Communications and Network*, 2011, 3, 85-98.

[7] W. Stevens, TCP/IP Illustrated, in: The Protocols, vol. 1, Addison-Wesley, 1994.

[8] Kolawole I. Oyeyinka1, Ayodeji O. Oluwatope2, Adio. T. Akinwale3, Olusegun Folorunso3, Ganiyu A. Aderounmu2, Olatunde O. Abiona, TCP Window Based Congestion Control Slow-Start Approach, *Communications and Network*, 2011, 3, 85-98.

[9] Xiaolong Qian, Yuanwei Jing, Jing Tian, "Network congestion avoidance strategy with particle filter", Computer Communications 31 (2008) 1723–1726.

[10] B. Suter, T.V. Lakshman, D. Stiliadis, and A. Choudhury, "Design Considerations for Supporting TCP with Per-Flow Queueing," in Proc. Of IEEE Infocom '98, March 1998, pp. 299–305.

[11] M. Scharf, Comparison of end-to-end and network-supported fast startup congestion control schemes, Computer Network Journal ,Feb 2011

[12] K. Tan, J. Song, Q. Zhang, M. Sridharan, A compound TCP approach for high-speed and long distance networks, in: Proceedings of IEEE Infocom, April 2006.

[13] Wenjun Xu, ZudeZhou, D.T.Pham, C.Ji, M.Yang, QuanLiu, Hybrid congestion control for high-speed networks, Journal of Network and Computer Applications 34 (2011) 1416–1428.

[14] Wei DX, Cao P. A Linux TCP. Implementation for NS2 2007.