Applied Mathematics & Information Sciences
An International Journal

A Logarithmic Slow-Start Algorithm of TCP Vegas in IP networks

Mao Kai[1,2]

[1] College of Computer Science, Chongqing University, Chongqing, PR China
[2] College of Computer Science and Information Engineering, Chongqing Technology and Business University,Chongqing

Received: Aug. 20, 2011; Revised Nov. 27, 2011; Accepted Dec. 9, 2011
Published online: 1 Mar. 2013

Abstract: In numerous TCP congestion control algorithms, TCP Vegas avoids congestion on its own initiative. However, its slow-start ending quickly and growing unsmoothly cause that the congestin window designing too small and bandwidth utilization declining too rapid, furthermore affect network efficiency and throughput dropping...etc. Aimed at the deficiencies, we propose a kind of Logarithmic Slow-Start Algorithm of TCP Vegas (LSSA-TCP Vegas) to conduct the dynamics of congestion window. We analyze stability condition from the theory via the Nyquist stability criterion and study the issue such as convergence of the algorithm, configuration parameters. Our goal in this paper is to study and compare the performance of LSSA-TCP Vegas and other TCP congestion control algorithms under various network scenarios. The simulation results demonstrate that the algorithm vs the bug of TCP Vegas contains obvious improvements. We believe that Logarithmic Slow-Start Algorithm can be used to provide the better performance of TCP Vegas.

Keywords: LSSA-TCP Vegas, throughput, stabilization, convergence

1. Introduction

The computer network, since the birth, specially the recent several years, obtained the fast development. The Internet has already been government, business company, social organization and individual a necessary fraction. Continuous development along with the network scale, the network congestion becomes more and more serious [1]. The primary congestion reason in network is that the users need the transmission capacity and handling ability are much more than the limit which the actual network can provide, thus causes that the data loss rate rises as well as network system application performance drops and other problems occur.

The TCP protocols are the most dominated protocols in the Internet [2], their success lies in the good congestion control strategy. Owing to outstanding of congestion control strategy ensure Internet to develop rapidly [3]. According to statistic from MCI, 90% total data quantity is transmitted by TCP protocols [4]. They are mainly used for reliable end to end delivering. This need to guarantee stability, efficient transmission,..., etc. There have been active research and increasing interest in the area of Internet congestion control [5,6]. And TCP congestion control generally adjust the congestion window size according to loss rate, this includes Reno[7], perhaps the most well-known TCP congestion control algorithm, is designed based on the rule. However, the reason of loss data is varied, not only just for transmission congestion, so TCP Vegas judge whether the network is in congestion and adjust congestion window size according to the changes of round-trip-time (RTT) value, not loss rate, thus acquired more superior than TCP Reno in congestion control performance and transmission efficiency [8].

TCP Vegas adopt active congestion avoidance and minish the congestion window size before practical loss data occurrence, pass a repeated ACK package, not the algorithm of Reno of three repeated ACK packages to judge whether the transmission is overtime, thus find congestion occurrence in time. But this kind of congestion avoidance scheme have to coexist with other TCP protocols, when it predicts that the network will take place congestion, it will enter congestion avoidance stage on its own initiative, and release its part of bandwidths. This competition mode of the bandwidth occupancy with other TCP data streaming will be in bad situation. Owing to the flaw, TCP Vegas

* Corresponding author: e-mail: shylmath@hotmail.com

finally fail to large-scale commercial application in Internet[9].

Compared to the "coarse" TCP Vegas, Srijith,etc designed the Vegas-A mechanism [10], a kind of refined TCP Vegas algorithm that doesn't need to depend on any key parameter to alleviate congestion, at first it conducts the parameters via transmission mode. While Vegas-A data streamings coexist with other TCP data streamings, the weak capability of occupancy bandwidth contains some improvements, it can adapt to various RTT and the change of path. But the TCP Vegas-A mechanism is in the process of delivering, with regard to the intrinsic difficulty, incapability of occupancy bandwidth quickly is obvious in the network.

In order to overcome the shortcoming of the TCP Vegas and Vegas-A mechanism and satisfy requirements of different types of traffic service in terms of higher throughput and less loss rate, this paper proposes the novel Logarithmic Slow-Start Algorithm of TCP Vegas (LSSA-TCP Vegas), in the process of Slow-Start stage, its congestion window traces the total of ACK packages and adopts a kind of slippery growth via the logarithm method, effectively settles congestin window steep growth, avoids to make it increase the skip step, and enters smoothly congestion avoidance stage. The design objectives of LSSA-TCP Vegas includes minimizing loss rate, stabilizing transmission, maintaining high link utilization, and removing biases against burst traffic. The LSSA-TCP Vegas algorithm increases congestion window quickly at first, avoids the sudden data streamings, slows the congestion window growth later, and defers the occurrence of the network congestion. This is very beneficial to the small data streamings connectivity, before the congestion occur, the data transmission has been already completed[19]. Theory analysis and simulation results will be verified and illustrated by Nyquist stability criteria and ns2 simulator, respectively, obtaining the practical values and merits of the stability condition and convergence efficiency for designing an effective LSSA-TCP Vegas algorithm in this note.

The rest of the paper is organized as follows. We first propose LSSA-TCP Vegas model via the linear model in Section 2. In Section 3, we analyze the stability and derive stability conditions. Furthermore, we also give a convergence efficiency study of the performance of LSSA-TCP Vegas. Section 4 shows the simulation results to validate the analysis by comparing the varying parameters and performances of LSSA-TCP Vegas with other TCP congestion control algorithms. The conclusion follows in Section 5.

2. LSSA-TCP Vegas congestion control model

The TCP congestion control strategy mainly includes 4 following processes [11]: (1) The Slow-Start stage, assures the establishment of connectivity, meanwhile, with transmission rate of congestion window raises and avoids much more data to enter into congestion window at first [12]. (2) Congestion-Avoidance stage, while the transmission rate reaches a certain value, the algorithm enters congestion avoidance stage. This step is mainly to keep network from congestion. (3) Fast-Retransmit stage, retransmit the loss data as soon as possible when congestion occurs. (4) Fast-Recovery stage, adopts a related recovery strategy and renews next Slow-Start stage [20,21].

The duration of TCP vegas arrives the best available rate from the connectivity establishment has the important impact on its whole performance [9]. The congestion window size increases too quick later in the Slow-Start stage will result in the drop of performance, also will cause longer Congestion-Avoidance stage at the same time [13]. Therefore, the urgent assignment in Slow-Start stage is to shorten time of transmission rate to occupy an available bandwidth as far as possible, meanwhile, raise throughput of the network.

In the LSSA-TCP Vegas dynamic model, we facilitate the window-based logarithmic additive increase and multiplicative decrease (LAIMD) principles to describe the basic feedback mechanism. It is assumed to increase the congestion window size, additively with logarithmic difference at each round trip time (RTT). And it shortens the congestion window size, multiplicatively by half if a packet loss is detected.

The LSSA-TCP Vegas model is defined as follows.

$$W_i(t) = \begin{cases} W_i(t-1) + \alpha(\log_2 \sum_{i=0}^{n} T_{ack} - \\ \log_2 \sum_{i=0}^{n-1} T_{ack}), & \text{non-congestion} \\ \frac{1}{2} W_i(t-1), & \text{congestion} \end{cases} \quad (1)$$

Where $W_i(t)$is congestion window size of ith loss cycle(LC_i) in time t and $\alpha(>0)$ is a scaling factor, additive rate-increase parameter. We assume that the source sends data packets and receives ACK packets all by rate $X_i(t)$, and every packet to be confirmed. For one non-congestion confirmed packets, the congestion window size increases $\alpha(\log_2 \sum_{i=0}^{n} T$ $\log_2 \sum_{i=0}^{n-1} T_{ack})$; for one congestion confirmed packed, the congestion window size is halved. Let $P(t)$ denote the congestion probability in time t, according to average statistics, the network system can receive $\alpha(\log_2 \sum_{i=0}^{n} T_{ack} - \log_2 \sum_{i=0}^{n-1} T_{ack})(1-P(t))$ non-congestion confirmed packed and $w_i(t)P(t)$ congestion confirmed packed in the LC_i. For simply $\Delta \log_2 T$ denotes $\log_2 \sum_{i=0}^{n} T_{ack} - \log_2 \sum_{i=0}^{n-1} T_{ack}$, so the LSSA-TCP Vegas model can also be described as follows:

$$W_i(t) = \alpha(\Delta \log_2 T)(1 - P(t)) - \frac{1}{2} W_i(t) P(t) \quad (2)$$

3. LSSA-TCP Vegas Performance analysis

In this section, through the evaluation of the stability and convergence of LSSA-TCP Vegas algorithm, we derive the stability condition, which is represented in terms of its control parameters and analyze the influence to the performance. Next, we provide a effective guideline for the preset parameters setting.

3.1. Stability

In order to analyze the stability of LSSA-TCP Vegas Algorithm, at first, we take Laplace transforms on (2), for simply, considering that $1 \gg P(t) = p_0$ and $W(t) = w_0$ at equilibrium point, then we can obtain

$$sW(s) = \alpha(\Delta \log_2 T)\frac{1}{s} - \frac{1}{2}w_0 P(s) \tag{3}$$

Further, consider that the aggregate rate $X(t)$ will be $w_i(t)$ and LSSA-TCP Vegas network having the number of connections N, the round trip time R_0, and the capacity of the bottleneck transmission link C, so we have $X_i(t) = \frac{W_i(t)}{R_0}N$ and $X_i(t) = \frac{C}{N}$ at equilibrium point and we can obtain the LSSA-TCP Vegas controller transfer function as follow:

$$F_{LSSA(s)} = \frac{p(s)}{x(s)} = \frac{2N}{C}(\frac{\alpha(\Delta \log_2 T)N^2}{sCR_0} - s) \tag{4}$$

Consider the generic TCP dynamic model suggested by Vishal Misra, etc in 2000 [14]

$$q(t) = \frac{W(t)}{R(t)}N(t) - C \tag{5}$$

G.V.Hollot facilitates the application of small signal analysis method and obtains the linear model in the following form at equilibrium point [15].

$$\delta q(t) = \frac{N}{R_0}\delta w(t) - \frac{1}{R_0}\delta q(t) \tag{6}$$

Similarly, consider $\delta x(s) = \frac{N}{R_0}\delta w(t)$ at equilibrium point, then we take Laplace transforms on (6) and give

$$\delta x(s) = (s + \frac{1}{R_0})\delta q(s)\delta \tag{7}$$

Thus, the closed-loop feedback control model based on queue in [15] is now changed into based on rate, which is shown in Fig. 1

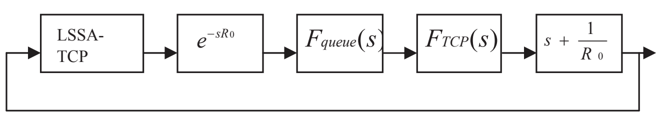

Figure 1 the closed-loop feedback control of LSSA-TCP Vegas

Note that, $F_{TCP(s)} = \frac{\frac{R_0C^2}{2N^2}}{s+\frac{2N}{R_0^2C}}$ is the delivering function of TCP features, $F_{queue(s)} = \frac{\frac{N}{R_0}}{s+\frac{1}{R_0}}$ is delivering function of queue features, and e^{-sR_0} is hysteresis phase caused by RTT.

To this end, we deduce the LSSA-TCP Vegas delivering function in the following form:

$$\begin{aligned} F(s) &= F_{LSSA}(s) \times F_{TCP}(S) \times F_{QUEUE}(s) \times e^{-sR_0} \\ &\times (s + \frac{1}{R_0}) \\ &= (\frac{\alpha(\Delta \log_2 T)N^2}{sCR_0} - s)\frac{R_0^2C^2e^{-sR_0}}{CR_0^2s + 2N} \end{aligned} \tag{8}$$

Theorem 1 *Stability Conditions: the LSSA-TCP Vegas is asymptotically stable if the control parameters satisfy*

$$\alpha(\Delta \log_2 T)N^2R_0C - \omega^2R_0^2C^2 - \omega\sqrt{(2N)^2 + \omega^2(R_0^2C)^2} \le 0$$

$$\textit{Where } \omega = \min\{\frac{2N}{R_0^2C}, \frac{1}{2R}\}. \tag{9}$$

Proof:

From (8), we have

$$F(j,\omega) = \frac{\alpha(\Delta \log_2 T)N^2R_0Ce^{-j\omega R_0}}{(2N + R_0^2Cj\omega)j\omega} - \frac{R_0^2C^2j\omega e^{-j\omega R_0}}{(2N + R_0^2Cj\omega)} \tag{10}$$

Next, we have

$$|F(j\omega)| \le \frac{\Delta \log_2 T)N^2R_0Ce^{-j\omega R_0}}{\omega\sqrt{(2N)^2 + (R_0C)^2\omega^2}} - \frac{R_0^2C^2\omega}{\sqrt{(2N)^2 + (R_0C)^2\omega^2}} \tag{11}$$

Thus, if the inequality (11) is reached, we obtain $|F(j\omega_g)| \le 1$ and

$$\begin{aligned} \angle F(j\omega) &\ge (\angle e^{-j\omega R_0} - \angle(2N + R_0^2Cj\omega)) \\ &\ge -\frac{180^\circ}{2\pi} - 90^\circ - 45^\circ > -180^\circ \end{aligned} \tag{12}$$

In summary, since the condition of Nyquist stability criterion is satisfied, the LSSA-TCP Vegas is asymptotically stable.

3.2. Convergence

In order to design that the congestion control algorithm being able to fully utilize the bandwidth as soon as possible, we expect the algorithm to have a short convergence time. To better calculate how much time the LSSA-TCP Vegas requires to reach a certain level of efficiency,

we define:

Definition 1 *For a given positive constant $\theta(0 < \theta \leq 1)$, the bandwidth resource allocation $(x_1, x_2, \ldots, x_N)$ is θ efficiency, if:*

$$e(t) = \frac{\sum_{i=1}^{n} x_i(t)}{C} \geq \theta$$

Theorem 2 *Consider N synchronous LSSA-TCP Vegas flows starting to compete for the capacity of the bottleneck transmission link C with the initial rate of each flow equal to 0, then after*

$$t_\theta = \frac{2}{p_0} \ln\left(\frac{1}{1 - \frac{\beta}{2\Delta \log_2 T}\theta}\right) \tag{13}$$

the network achieves θ efficiency. Where β denotes $\frac{CR_0P_0}{N^2(1-P_0)}$.

Proof:

Since $X(t = 0) = 0$, there is no packet queuing before the utilization reaches θ, i.e., $q(t) = 0$. Considering all flows as synchronous, and the complete control law can be described by (2).

For simply, we suppose $P(t) = p_0$ at equilibrium point, and from (2) we obtain

$$W_i(t) = \alpha(\Delta \log_2 T)(1 - p_0) - \frac{1}{2}W_i - p_0 \tag{14}$$

Consider $W_i(t = 0) = 0$, $X_i(t) = \frac{W_i(t)}{R_0}N$ and $X_i(t) = \theta\frac{C}{N}$ as the initial condition and Solving (14), we obtain

$$\begin{aligned} t_\theta &= \frac{2}{p_0} \ln\left(\frac{\alpha(\Delta \log_2 T)(1 - p_0)}{\alpha(\Delta \log_2 T)(1 - p_0) - \frac{1}{2}W_i - p_0}\right) \\ &= \frac{2}{p_0} \ln\left(\frac{1}{1 - \frac{1}{2}p_0\theta\frac{CR_0}{N^2\alpha(\Delta \log_2 T)(1-p_0)}}\right) \\ &= \frac{2}{p_0} \ln\left(\frac{1}{1 - \frac{\beta}{2\Delta \log_2 T\theta}}\right) \end{aligned} \tag{15}$$

The time of convergence to efficiency of the LSSA-TCP Vegas is $O(\ln(\frac{1}{1-\frac{\beta}{2\Delta \log_2 T\theta}}))$. For example, if we set $\alpha = 10$, and $p_0 = 0.02\%$, while choosing the target θ as 0.995, then the LSSA-TCP Vegas flows saturate $C = 20Mbps$ link in only $15s$. This case is validated by simulation experiment in Section 5.1.

4. Performance evaluation

The objective of this section is to verify, by using the ns2 simulator, that the LSSA-TCP Vegas can obtain higher performance than other TCP congestion control algorithms on the aspects of stabilization and throughput under dynamic network environments.

In the simulation case, we divide the experiments into two parts, at first we perform simulation experiments for different values of configuration parameter, α, to assess the validity of the model in diverse environmental conditions.

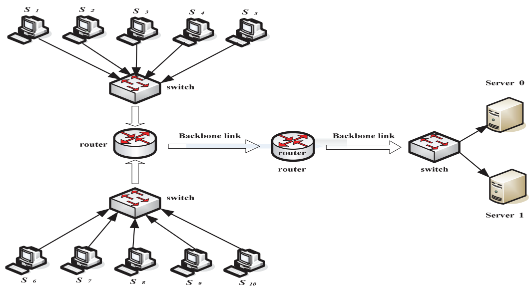

Figure 2 A network topology used for the evaluation of the LSSA-TCP Vegas

In the second we compare LSSA-TCP Vegas with RED, REM, PI controller and AVQ scheme.

we specify a network which is shown in Fig. fig:2. There are 10 source nodes and 2 destination nodes, meanwhile, the 1∼5 source nodes send data to destination node *server-0* and the 6∼10 source nodes send data to destination node *server-1*, respectively. Within these 10 branches, 1∼5 data flows start from 0s and 6∼10 data flows start from $5s$. The sending sequence have $1s$ interval and share a common backbone link, meanwhile, every node has the same buffer capacity of $10Mb$. The capacity of the bottleneck transmission link is $C = 20Mb/s$ and each link connecting to source or destination nodes has the same transmitting rate of $10Mb/s$. For convenience, we set the packet size to be $1kb$. Additionally, we set $W(0) = 0$ and $R0 = 0.1s$. Note that the values of the parameters remain unchanged throughout the whole process of simulation unless otherwise stated. We assume that the sender sources always have data to send and all links are duplex, furthermore, the drop-tail strategy is used when overflow occurs at the transmitting queue.

The simulated LSSA-TCP Vegas operates as follows: in every loss cycle, the source nodes trace the total of received ACK packages and increase the congestion window size W additively with $\alpha(\log_2 \sum_{i=0}^{n} T_{ack} - \log_2 \sum_{i=0}^{n-1} T_{ack})(1-p_0)$ at each RTT, i.e., $W(n+1) = W(n) + \alpha(\log_2 \sum_{i=0}^{n} T_{ack} - \log_2 \sum_{i=0}^{n-1} T_{ack})(1-p_0)$; and decrease the congestion window size W multiplicatively with half in case of packet loss, i.e., $W(n + 1) = 0.5 \times W(n)$.

4.1. Evaluation the LSSA-TCP Vegas with different configuration parameters

In the first case, we perform simulation experiments for the additive rate-increase parameter, α, to assess the validity of the LSSA-TCP Vegas using a dynamic traffic scenario. The simulation duration is $50s$.

We input initial a as the configuration parameter of the LSSA-TCP Vegas controller, and set $N = 100$ and $\alpha = 10, 20$. According to analysis of stability conditions in section 3.1, we guarantee that parameters of LSSA-TCP

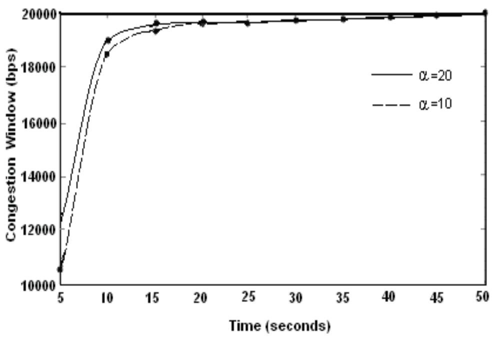

Figure 3 Congestion window achieved by the LSSA-TCP Vegas under the specified set value of α

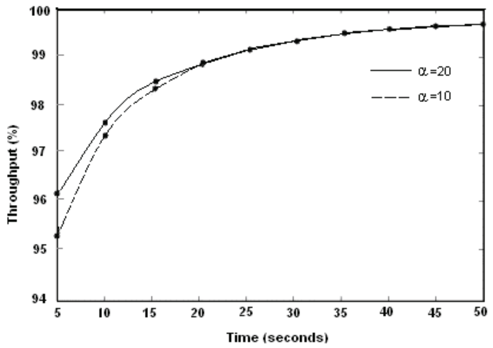

Figure 4 Throughput achieved by the LSSA-TCP Vegas under the specified set value of α

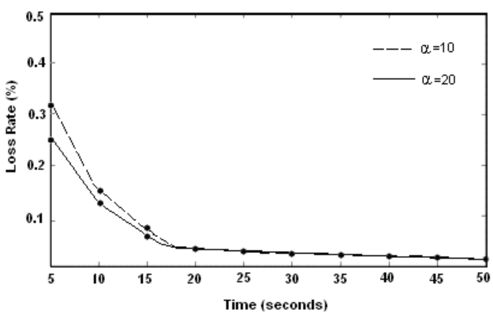

Figure 5 Loss Rate achieved by the LSSA-TCP Vegas under the specified set value of α

Vegas controller satisfy in equation (9), which is at the stability range. The congestion window, the through, and the loss rate are show in Fig. 3–5, respectively.

Observations from the simulation results of Fig. 3 suggest that the measured congestion window values with $\alpha = 10$ tracks very closely with $\alpha = 20$, especially for $t = 20s$ and afterword, there are nearly no difference between the congestion window values and provides almost full link utilization. The reason is that these values are all at the stability range according to in equation (9) no matter $\alpha = 10$ or $\alpha = 20$. Hence, the method of LSSA-TCP Vegas controller according to the varying link conditions can significantly improve the congestion window link utilization.

The similar patterns in the achieved throughput and loss rate are observed for experiments with $\alpha = 10$ and $\alpha = 20$ shown in Fig. 4 and Fig. 5. There are only some difference between the values of throughput and loss rate in the incipient stage. For example, the sources with $\alpha = 20$ can achieve 95% throughput, is more over than the sources with $\alpha = 10$ at $T = 5s$. And the same sources can achieve 0.25% loss rate less than the source with $\alpha = 10$ at the same time. Similarly, there are also nearly no difference after over a period of time, no matter the throughput or loss rate. On the other hand, a is not too large too benefit to LSSA-TCP Vegas, if we set a too large it would lead to significant network instability. There is possibility that setting very large a may lead to overshoot the available bandwidth during additive rate increase and may cause frequent rate oscillations and network instability according to in equation (9). The simulation results imply that the proposed mechanism is amenable to gradual deployment and obey the stability condition to reduce the impact of congestion. This feature may encourage the gradual adoption of LSSA-TCP Vegas on the Internet.

Table 1 Results of performance comparison under various TCP congestion control algorithms

criterion	LSSA-TCP Vegas	AVQ	PI	RED
Mean Length	50.05	53.76	48.33	67.34
STD	11.69	25.64	14.32	16.12
Throughput (%)	99.78	98.91	99.39	99.52
Loss rate (%)	0.04	0.46	0.09	2.38

4.2. Performance Comparison Under Different TCP congestion control algorithms

In the second simulation experiment, we compare LSSA-TCP Vegas with RED, PI controller and AVQ scheme. We set $N = 100$, $\alpha = 20$, and the simulation duration is $100s$. The target queue length is set to 50 packets. The control parameters of the other various TCP congestion control algorithms are summarized as follow.

We choose the parameters of AVQ that are set to the values recommended in [17], $\alpha = 0.15$, $\beta = 0.8$, $\gamma = 1.0$. In the PI controller scheme, we set the control parameters $a = 0.002445$, $b = 0.0024$ and the update interval $T_s = 10ms$ so as to satisfy the stability conditions in [18]. For RED, $\max_{th}$ and $\min_{th}$ are set to 20 packets and 80 packets, respectively, in order to keep the average queue length at around $(\max_{th} + \min_{th}) = 50$ packets, $p_{\max}$ is set to the default value of 0.1 [14, 15].

Simulation results are illustrated in Table. 1, which shows the comparison of performance under these four TCP congestion control algorithms, according to the previous parameters setting. In the case of PI, its queue length fluctuates less than RED and AVQ, but much more than LSSA-TCP Vegas. However, in the case of LSSA-TCP Vegas, the average queue length does not vary much from the target length regardless of the number of connection, in despite of TCP connections are established continuously. Table. 1 illustrates that the performance under the LSSA-TCP Vegas algorithm converges faster and fluctuates less around the target queue with smaller steady-state errors than RED, PI, and AVQ. Simulation results demonstrate that LSSA-TCP Vegas can achieve the mean queue length closer to the target queue, meanwhile, obtain more throughputs and less loss rate than the other TCP congestion algorithms. Therefore, the LSSA-TCP Vegas is an important improvement of congestion control to the window-based algorithms especially for complicated and diverse environments.

5. Conclusion

In this paper, we analyze the LSSA-TCP Vegas model from the control theory viewpoint. Furthermore, we also present the sufficient stability conditions and convergent time to the network transmission. Both of the theoretical analysis and simulation results demonstrate that LSSA-TCP Vegas algorithm can obtain the desired stability in terms of arbitrary and diverse link condition. This means that LSSA-TCP Vegas can perform better than the other TCP congestion algorithms. Although in this paper we only focus on the stability and convergence of the LSSA-TCP Vegas, the relative control technique explored in this paper can be extended to other TCP or TCP-based algorithms for their stability and convergence analysis. Our results reported can also be extended to networks with uncertain network conditions, leaving several important and interesting topics for exploratory research.

References

[1] F. Paganini, Z. Wang, John C. Doyle and Steven H. Low: "Congestion Control for High Performance, Stability, and Fairnessin General Networks". IEEE/ACM TRANSACTIONS ON NETWORKING, VOL.**13**, NO.1, FEBRUARY, 2005.

[2] T. Kelly, "Scalable TCP: Improving Performance in High-speed Wide Area Networks,". First International Workshop on Protocols for Fast Long-Distance Networks, February, 2003.

[3] David X. WEI, Cheng Jin, Steven H. Low and Sanjay Hegde "FAST TCP: MOTIVATION, ARCHITECTURE, ALGORITHMS, PERFORMANCE". IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. **14**, NO. 6, DECEMBER 2006

[4] S. Floyd, "High-speed TCP for Large Congestion Windows" RFC 3649, December 2003.

[5] R. Johari and D. Tan, "End-to-end congestion control for the internet: delays and stability". IEEE/ACM Trans. Networking, Vol **9**, 2001, pp. 818-832.

[6] L. Massoulie, "Stability of distributed congestion control with heterogeneous feedback delays". Microsoft Research, Cambridge, U.K., Tech. Rep. TR 2000-111, 2000.

[7] G. Vinnicombe, "On the stability of end-to-end congestion control for the Internet". Technical report CUED/F-INFENG/TR.398. http://www-control.eng.cam.ac.uk.

[8] Brakmo L S, Peterson L L. TCP vegas: end to end congestion avoidance on a global internet [J]. IEEE Journal on selected Areas in Communications, 2003 (10), pp. 1465-1480

[9] Vanichpun S, Feng Wuchum. On the transient behavior of TCP vegas [C] //Proceedings of Eleventh International Conference on Computer Communications and Networks, USA: Mary land Univ. IEEE, 2002, pp.504-508.

[10] Srijith K N, Jccobl L, Ananda A L. TCP vegas-a: improving the performance of TCP vegas[J], Computer Communciations, 2005, **28** (4), pp.429-440.

[11] Ramana B Venkata, Manoj B. SAR-TCP: a loss-aware adaptive rate based TCP for ad hoc wireless networks [J]. journal of high speed networks: 2006,**15** (1), pp.53-72

[12] Hsiao A Hsufeng, Hwang Jenqneng. A max-min fairness congestion control for layered streaming of scalable video [J]. IEEE transactions on circuits & systems for video technology, 2006, **16**(9), pp.1074-1085.

[13] B. Akan: "On the throughput analysis of rate-based and window-based congestion control schemes". Computer Networks **44** (2004), pp.701-11.

[14] Vishal Misra, Gong Wei-Bo, Don Towsley, Fluid-based analysi of a network of a network of AQM routers supporting TCP flows with an application To RED [A]. In Proceedings of ACM SIGCOMM [C]. Stockholm, Sweden. 2000, pp.151-160.

[15] Hollot C, Misra V, Towsley D, et al. A control theoretic analysis of RED [C]. IEEE INFOCOM. **20**

[16] T. Abdelzaher, J. Stankovic, C. Lu, R. Zhang, Y. Lu, "Feedback performance control in software systems, Proc". IEEE Control Systems **23** (3) (2003) 7490.

[17] S. Kunniyur, R. Srikant, Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management, in: Proceedings of ACM SIGCOMM, 2001, pp. 123-34.

[18] J. H. MO and J. WALRAND, "Fair end-to-end window-based congestion control". IEEE/ACM Transactions on Networking, vol. **8**, 2000, pp. 556-567.

[19] Y. P. Tian and H. Y. Yang, "Stability of the Internet congestion control with diverse delays". Automatica, Vol **40**, 2004, pp.1533-1541.

[20] S. Deb and R. Srikant, "Global stability of congestion controllers for the Internet". IEEE Transactions on Automatic Control, Vol **48**, 2003, pp. 1055-1060.

[21] S. Floyd, "High-speed TCP for Large Congestion Windows". RFC 3649, December 2003.

Mao Kai was born in Neimenggu District, China in 1968. He received his B.S. and M.S. degree in Computer Application and Computer Structure from Hunan University and Chongqing University, China, in 1990 and 1998, respectively. His research interests include network congestion control, stability analysis and management information system.