ISSN: 1813 – 1662 (Print)
E-ISSN: 2415 – 1726 (On Line)

IMPROVEMENT OF TCP PERFORMANCE TO AVOIDING NETWORK CONGESTION BASED ON LARGE-BANDWIDTH AND LOW-LATENCY

Maaeda Mohsin Rashid
College of Medicine, Kirkuk University , Kirkuk , Iraq
E-mail: maeda_hiderm@yahoo.com

Abstract
In recent days, the need to provide reliable data transmission over Internet traffics or cellular mobile systems becomes very important. Transmission Control Protocol (TCP) represents the prevailing protocol that provide reliability to data transferring in all end-to-end data stream services on the Internet and many of new networks. TCP congestion control has become the key factor manipulating the behavior and performance of the networks. TCP sender can regulates the size of the congestion window (CWND) using the congestion control mechanism and TCP dynamically adjust the window size depending on the packets acknowledgment (ACK) or by indicates the packets losses when occur. TCP congestion control includes two main phases, slow-start and congestion avoidance and these two phases even work separately, but the combination of them controls CWND and the packet injection to the network pipe. Congestion avoidance and slow-start are liberated mechanisms and using unlike objectives, but if the congestion happens, they are executed together. This article provides an efficient and reliable congestion avoidance mechanism to enhancing the TCP performance in large-bandwidth low-latency networks. The proposed mechanism also includes a facility to send multiple flows over same connection with a novel technique to estimate the number of available flows dynamically, where the all experiments to approving the proposed techniques are performed over the network simulation NS-2.

Keywords: *TCP; congestion control; CWND.*

Introduction
The congestion avoidance algorithm starts to run if TCP in sender side is detect a packet dropping in network link, then TCP sets the value of slow-start threshold (ssthresh) to one-half of the current CWND size[1].

When the capacity of CWND becomes equal or less than the size of ssthresh, that means TCP transmitter side will be in slow-start mode and if its bigger than the ssthresh, then TCP must be in congestion avoidance mode, as shown in formula below[2]:

if cwnd(t) ≥ ssthresh/2

$$cwnd\ (t+\tau) = cwnd\ (t) + \frac{1}{cwnd\ (t)} \qquad (1)$$

For each new RTT

$$cwnd(t)+1$$

For every packet loss (Timeout):

$$\frac{cwnd\ (t)}{2} \rightarrow ssthresh$$
$$cwnd\ (t+\tau) \rightarrow 1 \qquad (2)$$

In this congestion control mechanism, CWND(t) represents the present window value, while CWND(t+τ) represents the next window value.

TCP performance of typical TCP versions with standard congestion control mechanism decreases expressively when used with wireless networks or cellular systems. Additionally, the typical TCP's doesn't accomplish correctly[3] over high speed channels. Let's say, to getting a channel throughput of about 7Gbps if RTT sets to 100msec with 1.5 KB for packet size, and estimating the throughput equation shown below[3][4]:

$$Throughput = \frac{MSS}{RTT\ \sqrt{PPL}} * C \qquad (3)$$

When C is a constant and is equal to √3/2 [3], the implementation becomes as follows:

$$7 \times 10^{9} = \frac{1500 * 8}{100 \times 10^{-3}\sqrt{PPL}} * \sqrt{3/2}$$

$$PPL = 4.408 \times 10^{-10}$$

This improves, that the predictable PPL - Probability of Packet Loss must not touch 4.408 x 10^{-10} because this is unacceptable packet loss rate. Furthermore, the standard TCP version requires a very large number of RTT to achieving the required 40,000for one packet loss[5].

Additive increase multiple decrease mechanism
Many standard TCP variants used a congestion control mechanism called Additive Increase Multiple Decrease (AIMD). AIMD[6] mechanism represents a feedback congestion window method is used in TCP congestion avoidance phase, and this method acquaintances a linearly window incrementing while exponentially decreasing the window if congestion status detected [7]. The AIMD algorithm recommends that a dropping in packets occurs when CWND touches the value of *w* (size of packet), as shown in Fig. 1[8].

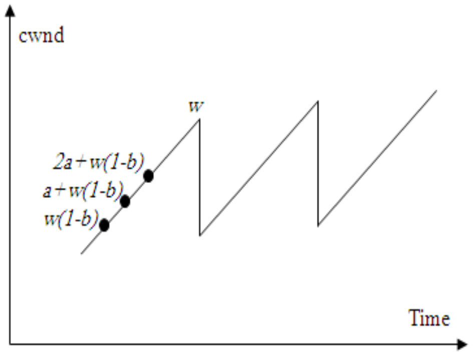

Fig. 1. AIMD congestion control technique

The modeling of AIMD mechanism is based on the using decrementing of CWND in multiplicative manner and back to the value of *w(1-b)* and for every congestion situation CWND increases by *w(b/a)+1* per each RTT[9].

Besides, when *S* means the transmitted packets number per, while *T* means the transmitted packets number per second, then the middling transmitted packet rates per each congestion cycle becomes [10]:

$$S = w\frac{2-b}{2} \quad , \quad T = w\frac{2-b}{2\,RTT} \qquad (4)$$

Then the total number of packets for each congestion epoch can be given as:

$$S\left(w\frac{b}{a}+1\right) = w\frac{2-b}{2}\left(w\frac{b}{a}+1\right)$$

$$\approx \frac{w^2(2-b)b}{2a}$$

At the end of congestion epoch and with one packet loss, the rate of packets drop p becomes:

$$p = \frac{2a}{w^2(2-b)b + w(2-b)a} \qquad (5)$$

or by approximation:

$$p \approx \frac{2a}{w^2(2-b)b} \qquad (6)$$

After approximate equation (6) corresponding to cwnd, we can get the simplified formula as shown below:

$$w \approx \sqrt{\frac{2a}{p(2-b)b}} \qquad (7)$$

Currently, by substituting Eq. (7) in Eq. (2) to indicating the transmitted packet *S* for each second, then we can obtain:

$$S = \frac{\sqrt{a}\,\sqrt{(2-b)}}{RTT\,\sqrt{p}\,\sqrt{2b}} \qquad (8)$$

When assume a=1 and b=0.5 and apply that in equation (7) we can get the following:

$$S = \frac{\sqrt{1.5}}{RTT\,\sqrt{p}} \qquad (9)$$

$$\frac{\sqrt{1.5}}{RTT\,\sqrt{p}} = \frac{\sqrt{a}\,\sqrt{(2-b)}}{RTT\,\sqrt{p}\,\sqrt{2b}} \qquad (10)$$

Equivalently, equation (10) can be written as following:

$$a = \frac{3b}{2-b}$$

And:

$$b = \frac{2a}{3+a} \qquad (11)$$

Eq. (11) denotes the estimated determination of AIMD mechanism that can providing a proposed values of the pair *a,b*, for example : 1/5- 1/8 and 3/7 - 1/4) and every value must contend properly in AIMD mechanism if *a*=1 and *b*=½.

Enhanced aimd mechanism

The objective of enhancing AIMD mechanism is by deriving a new association between *a* , *b* parameters, to making this pair capable to create an extensive range of desired packets rate at the end of link, because the typical AIMD that are used over current TCP versions unable to send enough packet rates over the available bandwidth and that leaves a reasonable capacity over unused channels. So, the efficient technique to sending an extra data over these unused channels is to rearranging the formula and relationship between the control parameters in AIMD.

The association between *a* and *b* would be used to controlling the parameters *(a,b)* and *k* to producing the requested packet rate affording to the transfer data formula of TCP as a function of *a* and *b*.

Initially, the size of CWND in a steady situation state for each *k* packet flows is calculated by:

$$cwnd_k = k * cwnd_{aimd} = \frac{k\sqrt{3/2}}{\sqrt{p}} \qquad (18)$$

While the equation that used to estimating CWND size rendering to the packets losses is stated as shown below[11]:

$$cwnd = \sqrt{\frac{2a}{p(2-b)b}} \qquad (19)$$

$$T_{(a,b,p,RTT)} = cwnd\left(\frac{2-b}{2\,RTT}\right) \qquad (20)$$

In Eq. (20), T assigns to the transmitted packets rate, while CWND is calculated estimated at the finish of congestion period. In theory, if throughput of enhanced TCP is increases by k compared with the typical TCP, then it means:

$$T_{(a,b,p,RTT)} = k * T_{(1,1/2,p,RTT)} \qquad (21)$$

Now, after substitution a equal to one and b equal to half one in Eq. (19) and Eq. (20), it can obtain the value of T using the formula shown below:

$$T_{(a,b,p,RTT)} = \sqrt{\frac{2a}{p(2-b)b}}\left(\frac{2-b}{2RTT}\right) \qquad (22)$$

$$T_{(1,1/2,p,RTT)} = \sqrt{\frac{2}{p(1.5)0.5}}\left(\frac{1.5}{2RTT}\right)$$

$$T_{(1,1/2,p,RTT)} = \sqrt{\frac{1.5}{p}} * \frac{1}{RTT} \qquad (23)$$

Then, by substitution of Eq. (23) in Eq. (21), we obtain:

$$k * \sqrt{\frac{1.5}{p}} * \frac{1}{RTT} = \sqrt{\frac{2a}{p(2-b)b}} * \frac{2-b}{2RTT} \qquad (24)$$

If p and the period of RTT are equal in two sides of previous equation, then the simplified equation becomes:

$$k\sqrt{1.5} = \sqrt{\frac{a}{b}} * \frac{\sqrt{2-b}}{\sqrt{2}}$$

$$k^2\,\frac{3}{2} = \frac{a}{b} * \frac{2-b}{2}$$

$$3k^2 = \frac{a(2-b)}{b}$$

Lastly:

$$a = \frac{3bk^2}{2-b} \qquad (25)$$

$$b = \frac{2a}{a+3k^2} \qquad (26)$$

If a is set to be equal k , then the estimation of b value can be:

$$b = \frac{2}{1+3k} \qquad (27)$$

So, for every RTT:

$$cwnd = cwnd + (k / cwnd) \qquad (28)$$

and for every packet drop:

$$cwnd = cwnd\left(1 - \frac{2}{3k+1}\right) \qquad (29)$$

It's worth mentioning that all the experiments, modeling, simulation, and evaluations of this paper executed by using Network Simulator 2 (NS-2), where NS-2 provides environments to simulate and modeling multicast protocols; networks traffic, handovers, and other networks resources and conditions for wireless and wired channels.

Fig. 2, shows that the improved TCP (Reno) do better than TCP with standard features, even we still use the standard algorithm in slow-start phase, due to we aimed to seeing the performance and the outcome of the enhanced congestion avoidance mechanism only.

Fig. 3, shows that the CWND of enhanced TCP is missing the fast-retransmit mode because TCP Tahoe do not contains fast-retransmit phase. Furthermore, the consequence of four virtual flows (k=4) of the enhanced TCP clearly showed the difference in the behavior of the congestion window in both TCPs, and that return to the multiple flows that are used in the improved congestion control of the new TCP. Besides, in our experiments, we proposed high congestion network model to force the TCP congestion control to always enter congestion status to ensuring if the new mechanism can perform well and if the new TCP fulfill the congestion control mechanism can run properly over large bandwidth and high latency channels. Finally, the new technique, shows high performance and deliver reasonable throughput under high congestion link where we used the network simulator NS-2 for the performance evaluation as shown in Fig. 4.

The procedures are used in these simulation scenarios is based on different congestion condition and creating congestion event through each experiment to force the improved AIMD to run over congested link and to applying the new technique with high congestion level. All experiments used the improved algorithms in NS-2 script that used the enhanced TCP to trying see the performance clearly and independently.

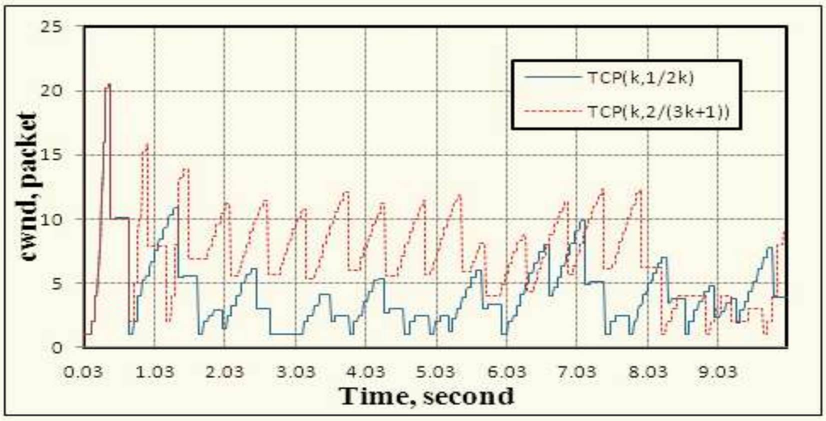

Fig. 2. The comparison of CWND between standard and improved TCP (*k*=2)

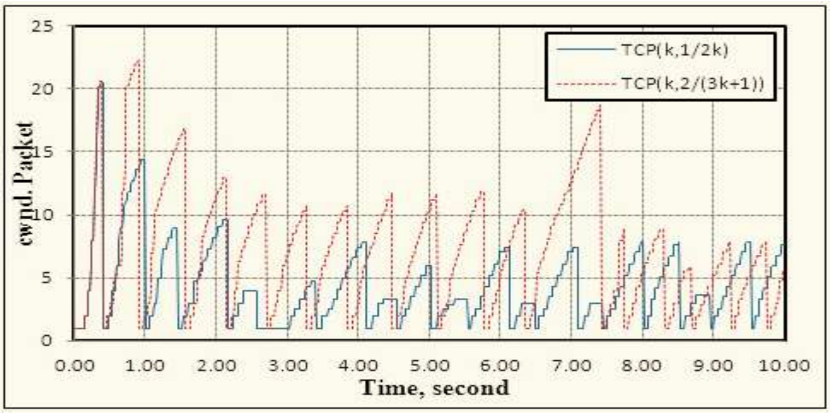

Fig. 3. The comparison of CWND between enhanced and standard TCP when *k* sets to *2*

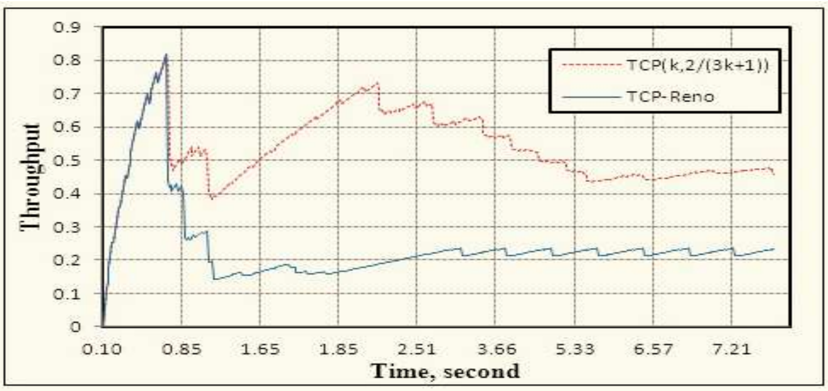

Fig.4. The comparison of between enhanced and standard TCP Reno when *k* set to 4

Conclusions

This manuscript, offered an enhanced TCP with an innovative mechanism in congestion avoidance situation, that are based on using multiple TCP flows and then control the flows number to adjusting the size of CWND during congestion status. Initially, the suggested technique is castoff the common AIMD procedure to controlling the growing in congestion window for per RTT in parallel with packet dropping. Also, the association of the parameters *a,b* is established by originating an enhanced association with ability to delivering additional effective flows in the similar TCP linking. The novel *a,b* association can deliver an extensive and varied series of favorite of *a* and *b* rendering to the simulated route that are supposed. The effort that are delivered in this research can be used in other TCP source variants such as Newreno, Tahoe, Sack and Fack by modifying the congestion control technique and then test it over network simulator to achieve the required throughput or to avoid packet drop as possible. Truly, the setting of virtual flow connection, either manually by user or dynamically by special algorithms still represents a major challenge to the parallel TCP's and for multipath TCP's too. Also, it's not acceptable to assume the number of virtual flows with one set of connection to be large because of the probability in unfairness between the parallel flows become more practicable. Despite the side effect of using a modified congestion control still degraded the theoretical performance, but the using of multipath TCP's remains the best choice in high speed networks where the available bandwidth reaches to several hundreds of Mbps. Besides, this paper shows that the improved TCP (Reno) do better than TCP with standard features, furthermore, the new technique, shows high performance and deliver reasonable throughput under high congestion links.

References

[1] Welzl M. Parallel TCP Data Transfers: A Practical Model and its Application: University of Innsbruck; 2010.

[2] Nabeshima M. Performance evaluation of multcp in high-speed wide area networks. IEICE transactions on communications. 2005;88:392-6.

[3] Mathis M, Semke J, Mahdavi J, Ott T. The macroscopic behavior of the TCP congestion avoidance algorithm. ACM SIGCOMM Computer Communication Review. 1997;27(3):67-82.

[4] Fu CP, Liew SC. TCP Veno: TCP enhancement for transmission over wireless access networks. Selected Areas in Communications, IEEE Journal on. 2003;21(2):216-28.

[5] Ghassan A. Abed, M. Ismail, K. Jumari, Modeling and Performance Evaluation of LTE Networks with Different TCP Variants, World Academy of Science, Engineering and Technology. International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering Vol:5, No:3, 2011.

[6] Wang J, Wen J, Zhang J, Han Y, editors. TCP-FIT: An improved TCP congestion control algorithm and its performance2011: IEEE.

[7] Floyd S, Jacobson V. Traffic phase effects in packet-switched gateways. ACM SIGCOMM Computer Communication Review. 1991;21(2):26-42.

[8] Demers A, Keshav S, Shenker S, editors. Analysis and simulation of a fair queueing algorithm1989: ACM.

[9] Ghassan A. Abed, Mahamod Ismail, and Kasmiran Jumari. Improvement of TCP Congestion Window over LTE-Advanced Networks, International Journal of Advanced Research in Computer Communication Engineering 1.4 (2012).

[10] Ghassan A. Abed, Mahamod Ismail, Jumari Kasmiran, Improvement of TCP Congestion Window over LTE-Advanced Networks, *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 1, no. 4, June 2012.

[11] Raghad T. Abdulrazzaq Al-Hassani, Methodical Improvement of Congestion Control Mechanism in Distinctive Data Transmission, Journal of Information, Communication, and Intelligence Systems (JICIS), Volume 2, Issue 3, September 2016.

تقنية فعالة لتجنب تزاحم البيانات ولتحسين اداء بروتوكول السيطرة على الارسال في الشبكات ذات الحزم العريضة والتأخير المنخفض

مائدة محسن رشيد

كلية الطب ، جامعة كركوك ، كركوك ، العراق

الملخص

يعتبر بروتوكول السيطرة على الارسال (TCP) هو البروتوكول الاكثر شيوعاً واستخداماً للعمل على اغلب انواع الشبكات وهو المسؤول عن نقل البيانات بين طرفيات الشبكات مع ضمان وصول تلك البيانات او حتى تامين اعادة ارسالها في حال حدوث فقدان في الحزم او البيانات عبر الشبكات المختلفة. يتم السيطرة على نقل البيانات بواسطة خوارزميات السيطرة على التزاحم عبر فتح نافذة الازدحام (CWND) من خلال آلية تسمى التحكم في الازدحام، وأن لبروتوكول الـ TCP القدرة على السيطرة على كيفية تدفق البيانات عن طريق تحديد حجم نافذة الازدحام CWND وفقا لوصول البيانات للطرف الاخر من الارسال بعد تلقي اشعار بوصول او عدم وصول حزمة البيانات. ان آلية التحكم في الازدحام تحتوي على طورين او مرحلتين رئيسيتين للسيطرة على حالة التزاحم اثناء تدفق البيانات، الاولى هي مرحلة البداية البطيئة (Slow-Start) بينما الثانية هي مرحلة تجنب الازدحام (Congestion Avoidance) وهذه المراحل مع بعضها تعمل بشكل تسلسلي أو بشكل منفصل. هذا البحث يقدم تقنية جديدة تعتمد على توفير نواقل افتراضية جديدة خلال مرحلة تجنب الازدحام من اجل زيادة أداء الـTCP في النواقل ذات الحزم العريضة والتأخير المنخفض.