Journal of Korea Multimedia Society Vol. 13, No. 12, December 2010(pp. 1798-1804)

General AIMD with Congestion Window Upper Bound

Dang-Quang Bui†, Myeonggil Choi††, Won-Joo Hwang†††

ABSTRACT

TCP with AIMD mechanism, one of the most popular protocols in internet, can solve congestion control in wired networks. This protocol, however, is not efficient in wireless networks. This paper proposes a new mechanism namely General AIMD with Congestion Window Upper Bound in which congestion window is limited by an upper bound. By applying optimization theory, we find an optimal policy for congestion window upper bound to maximize network throughput.

Key words: TCP, AIMD, General AIMD, Wireless Networks, Optimization.

1. INTRODUCTION

Transmission Control Protocol (TCP) [1] is one of the most popular protocols in the Internet. In TCP, reliable data transfer is solved by acknowledge and retransmission mechanisms while congestion control problem is solved by Additive Increasing Multiplicative Decreasing (AIMD) mechanism. In AIMD, congestion window is increased, probing for usable bandwidth, until loss occurs. The policy of additive increase may, for instance, increase the congestion window by 1 until a loss is detected. When loss is detected, the policy is changed to be one of multiplicative decrease, which may, for instance, cut the congestion window in half after loss. Note that the deduction of congestion window can solve only the congestion problem. Losses, however, may come from wireless environment in stead of congestion in wireless networks. In this case, deduction in congestion window cannot solve the loss problem on one hand and it leads to reduction of throughput on the other hand. Therefore, many authors [1-5] pointed out that this mechanism is inefficient in wireless networks.

Number of researches investigated in how to adapt TCP in wireless networks which were surveyed in [2,3]. A TCP version for multi-hop wireless networks was presented in [4]. In the same trend, Yamamoto *et al.* [5] proposed that if we restrain congestion window then network throughput may be improved. But the question how to restrain congestion window to achieve optimal throughput is not absolutely answered because it is lack of a systematic mathematical model. To overcome this problem, we will present a mathematical model which inherits general AIMD [6,7]. The differences between general AIMD and traditional one are that congestion window will be increased by α, call additive increasing factor, and decreased by β times, call multiplicative decreasing factor. In comparison with general AIMD, we restrain congestion window by an upper bound. Basing on proposed model, we use optimization theory to find an optimal policy for congestion

※ Corresponding Author : Won-Joo Hwang, Address : (621-749) Obangdong, Gimhae, Gyeongnam, Korea, TEL: +82-55-320-3847, FAX : +82-55-322-6275, E-mail : ichwang.inje.ac.kr

Receipt date : Oct. 5, 2010, Revision date : Dec. 29, 2010
Approval date : Jan. 5, 2011

† Dept. of Information and Communications Engineering, HSV-TRC, Inje University, South Korea.
(E-mail: huynkh@gmail.com, quangbd@gmail.com)
†† Dept. of Business Administration, ChungAng University, South Korea
(E-mail : mgchoi@cau.ac.kr)
††† Dept. of Information and Communications Engineering, HSV-TRC, Inje University, South Korea

※ This work was supported by the Inje Research and Scholarship Foundation in 2009.

window upper bound to maximize network throughput.

The main contribution of the paper is as follows:

- We propose a new mechanism namely General AIMD with Congestion Window Upper Bound by extending basic AIMD mechanism in which congestion window is limited by an upper bound.
- A mathematical model is proposed to estimate throughput in general AIMD with upper bound in two cases: constant congestion window upper bound and variable congestion window upper in section 2.
- By using optimization theory, we find an optimal policy for upper bound of congestion window to achieve maximal network throughput in section 3.

2. MATHEMATICAL MODEL

Yamamoto *el al.* [5] showed that throughput may be improved if we limit congestion window by an upper bound. We apply this idea in general AIMD [6]. We consider two cases: (1) congestion window is limited by a constant upper bound and (2) congestion window is limited by a variable upper bound which decreases when traffic is high.

2.1 Constant Congestion Window Upper Bound

In this case, we limit congestion window by a constant upper bound UB. Let α be the additive increasing factor and β be the multiplicative decreasing factor. When a new ACK arrives, the congestion window *cwnd*, which is maximal packet sent without ACK in a roundtrip time, is increased by α in general AIMD [6]. But it is bounded by UB; therefore, it is updated by the following equation:

$$cwnd := \min\{UB, cwnd - \alpha\}. \qquad (1)$$

When the sender detects congestion (packet loss), it decreases its *cwnd* by

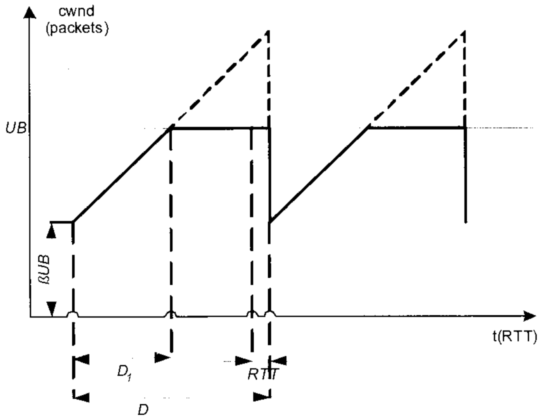


Fig. 1. General AIMD with a constant congestion window upper bound operation in a stable state.

$$cwnd := \beta cwnd. \qquad (2)$$

The operation of our protocol is shown in Figure 1. Without an upper bound, the general AIMD operation follows the dash line while our protocol operation is presented by the bold line. Let X be the number of packets transmitted in a loss cycle and D be the duration time of a loss cycle in a stable state. We assume that time is counted by roundtrip time RTT, i.e., time unit is RTT. The throughput is

$$T = \frac{X}{D}. \qquad (3)$$

Assume that D_1 is the duration from beginning of loss cycle to the time that congestion window reaches it upper bound. From the Figure 1, (1) and (2), we have:

$$\beta UB + \alpha D_1 = UB.$$

Thus,

$$D_1 = \frac{1-\beta}{\alpha} UB. \qquad (4)$$

Let n be the number of packets from beginning of loss cycle to the first packet loss. p denotes the probability that a packet is lost. Assume that this probability is equals for all packets. We have the expected number of n is computed by

$$E(n)=\sum_{i=1}^{\infty} i(1-p)^i p=\frac{1}{p}. \quad (5)$$

From the Figure 1, we can compute number of packets transferred in a loss cycle

$$X = D_1\frac{\beta UB+UB}{2}+(D-D_1)UB$$
$$=\left(D_1\frac{\beta-1}{2}+D\right)UB. \quad (6)$$

In other words, the sender only detects packet loss after a round trip time since the first packet loss. Therefore,

$$X=\frac{1}{p}+UB. \quad (7)$$

From (6) and (7), we have

$$X=\frac{1}{p}+UB=\left(D_1\frac{\beta-1}{2}+D\right)UB.$$

Then,

$$D=\frac{\frac{1}{p}+UB}{UB}-D_1\frac{\beta-1}{2}. \quad (8)$$

Substituting (4) and (8) in (3), we have

$$T=\frac{\frac{1}{p}+UB}{\frac{\frac{1}{p}+UB}{UB}+\frac{(1-\beta)^2}{2\alpha}UB}. \quad (9)$$

Our objective is to choose control parameters UB to maximize throughput. Therefore, the problem becomes

$$\max \quad T=\frac{\frac{1}{p}+UB}{\frac{\frac{1}{p}+UB}{UB}+\frac{(1-\beta)^2}{2\alpha}UB}, \quad (10)$$
$$\text{subject to} \quad UB>0.$$

2.1 Variable Congestion Window Upper Bound

In this case, we assume that in the first stage, the congestion window is increased with a slope of α_1. After it reaches the upper bound UB, it is decreased with a slope of α_2 . When the source detects packet loss, it decreases its data rate by a multiplicative decreasing factor β. The operation of the protocol is shown in Figure 2. Similar to the previous case, the expected number of packets from beginning of loss cycle to the first packet loss is determined by (5). From Figure 2, the congestion window *cwnd* at the beginning of each loss cycle is

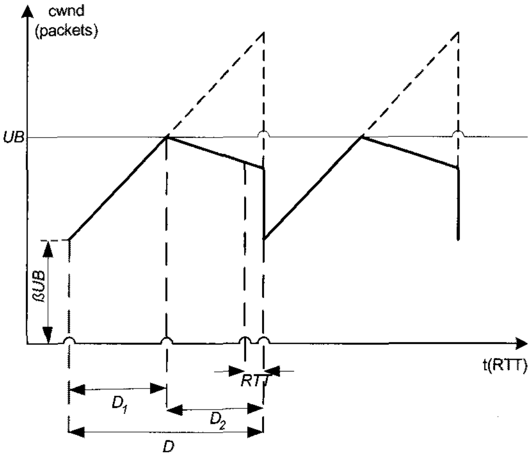


Fig. 2. General AIMD with a variable congestion window upper bound operation in a stable state.

$$cwnd_{start}=UB-\alpha_1 D_1.$$

And the congestion window at the end of each loss cycle is

$$cwnd_{end}=\beta(UB-\alpha_2 D_2).$$

Congestion window at the end of a loss cycle is also congestion window at the beginning of the next loss cycle. The operation of the protocol is assumed to be in a stable state; therefore, the congestion window at beginning of loss cycles are equal to each other's, i.e.

$$UB-\alpha_1 D_1=\beta(UB-\alpha_2 D_2). \quad (11)$$

The number of packets transferred in a loss cycle is

$$X=D_1\frac{UB-\alpha_1 D_1+UB}{2}+D_2\frac{UB-\alpha_2 D_2+UB}{2}$$
$$=D_1\frac{2UB-\alpha_1 D_1}{2}+D_2\frac{2UB-\alpha_2 D_2}{2}. \quad (12)$$

In other words, similar to (7), X can be also computed as

$$X=\frac{1}{p}+D_1\frac{UB-\alpha_2(D_2-1)+UB-\alpha_2D_2}{2}$$
$$=\frac{1}{p}+\frac{2UB-2\alpha_2D_2+\alpha_2}{2}. \tag{13}$$

From (12) and (13), we have

$$D_1\frac{2UB-\alpha_1D_1}{2}+D_2\frac{2UB-\alpha_2D_2}{2}$$
$$=\frac{1}{p}+\frac{2UB-2\alpha_2D_2+\alpha_2}{2}.$$

From (11), we obtain

$$D_1=\frac{(1-\beta)UB+\alpha_2\beta D_2}{\alpha_1}.$$

Substituting D_1 into (12), we have

$$\frac{(1-\beta)UB+\alpha_2\beta D_2}{\alpha_1}\frac{(1+\beta)UB-\alpha_2\beta D_2}{2}+D_2\frac{2UB-\alpha_2D_2}{2}$$
$$=\frac{1}{p}+\frac{2UB-2\alpha_2D_2+\alpha_2}{2}. \tag{14}$$

And a second order equation of D_2 is derived from (14)

$$AD_2^2-BD_2+C=0, \tag{15}$$

where

$$A=\frac{\alpha_2^2\beta^2}{2\alpha_1},$$
$$B=\alpha_2+\frac{\alpha_2\beta^2UB}{2\alpha_1}+UB,$$
$$C=\frac{1}{p}+\frac{2UB+\alpha_2}{2}-\frac{(1-\beta)(1+\beta)UB^2}{2\alpha_1}.$$

If $B^2-4AC<0$, (15) does not have any solution. In this case, system is not stable, implying that there is not any stable state of Fig. 2. So in this paper, assume that $B^2-4AC\geq 0$. Solving (15), we obtain

$$D_2=\frac{B-\sqrt{B^2-4AC}}{2B}. \tag{16}$$

Substituting D_2 into (11), we can find

$$D_1=\frac{(1-\beta)UB+\alpha_2\beta D_2}{\alpha_1}$$

where D_2 is given by (16).

Then we obtain the throughput as

$$T=\frac{\frac{1}{p}+UB}{D_1+D_2}. \tag{17}$$

And finally, similar to (10), our objective is to find maximize the throughput. Therefore, the problem becomes

$$\max \quad T=\frac{\frac{1}{p}+UB}{D_1+D_2}, \tag{18}$$
$$\text{subject to } UB>0.$$

Problems (10) and (18) are nonlinear optimization which can be solved by gradient method [8], which is presented in the next section.

3. CONGESTION WINDOW UPPER BOUND CONTROL ALGORITHM

In the previous section, we have throughput maximization problems for two cases in (10) and (18) respectively. To solve these two problems, we propose the following algorithm to control the congestion window upper bound UB such that the throughput is maximized.

The algorithm is implemented in each source in which general AIMD is used. In this algorithm, each source knows its general AIMD parameters. It can also monitor network status by maintaining the loss probability. Because all required information is stored locally in each source, Algorithm 1 is implemented in a distributed way. After having all required information, the algorithm starts with an arbitrary congestion window upper bound in step 2. In step 3, we choose a gradient direction. Because the problem is maximization, we follow the throughput derivative direction here.

After steps 4 and 5, we can generate a new congestion window upper bound UB which improves throughput in comparison with the previous step. The optimal UB is achieved when the stopping criteria is satisfied in step 6.

4. NUMERICAL ANALYSIS

In this section, we conduct some numerical examples. First we evaluate the dependence of throughput on congestion window upper bound. In this example, we assume that traditional TCP $(\alpha=1, \beta=1/2)$ is used. In the constant congestion window upper bound scenario, throughput, which is computed by (9), is presented in Figure 3. In the variable congestion window upper bound scenario $(\alpha_1=1, \alpha_2=1, \beta=1/2)$, throughput, which is computed by (17), is presented in Figure 4. In Figure 4, UB must be greater than a threshold to guarantee $B^2-4AC \geq 0$. We can see that with a given error rate p, there exist an optimal upper bound UB such that throughput is maximized for both two cases.

To find an optimal upper bound , we implement gradient method [8] to solve optimization problem (10) (the constant congestion window upper bound case). Figure 5 show that the iterative generated by gradient method converges to an optimal solution. The algorithm converges after several hundreds steps depending on general AIMD parameters. At the same time, throughput also

---

Algorithm 1: Congestion Window Upper Bound Control Algorithm

Step 1: Monitor packet loss to maintain the loss probability p. Maintain the general AIMD parameters (α, β - constant upper bound case; $\alpha_1, \alpha_2, \beta$ - variable upper bound case).

Step 2: Initialize the congestion window upper bound $UB = UB_0$.

Step 3: Choose a gradient direction $d=\frac{dT}{dUB}$ in which T is computed in (11) and (25) respectively for the two cases.

Step 4: Choose a enough small step size γ such that $T(UB+\gamma d) > T(UB)$.

Step 5: Update a new congestion window upper bound $UB = UB + \gamma d$.

Step 6: If $\| d \| < \epsilon$ (stopping criteria, $\epsilon \sim 10^{-3}$), then stop. UB is the optimal congestion window upper bound to maximize throughput. Otherwise, return step 3.

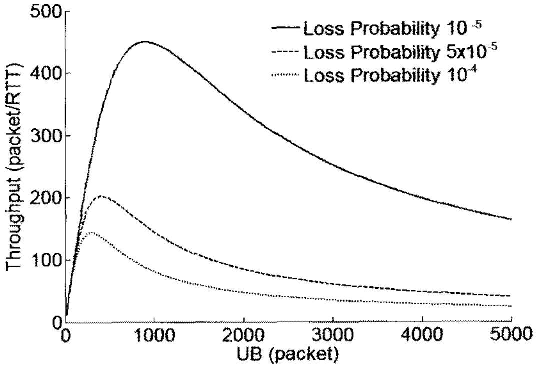


Fig. 3. Throughput over UB-Constant upper bound.

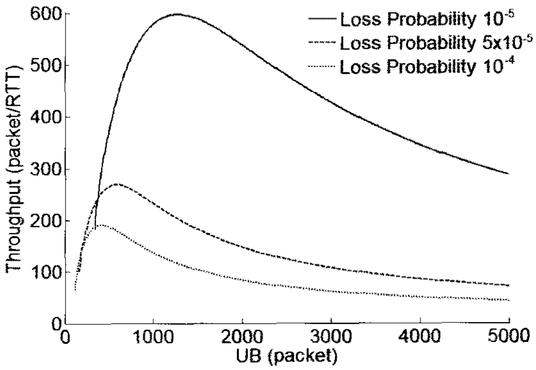


Fig. 4. Throughput over UB-Variable upper bound.

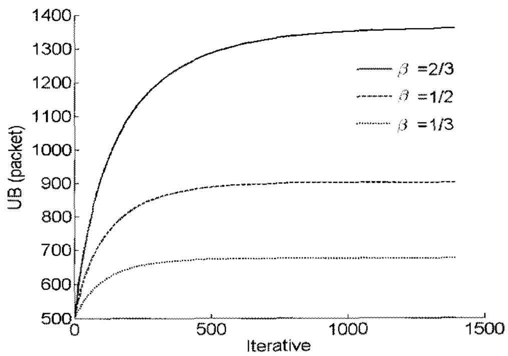


Fig. 5. Convergence of UB-Constant upper bound.

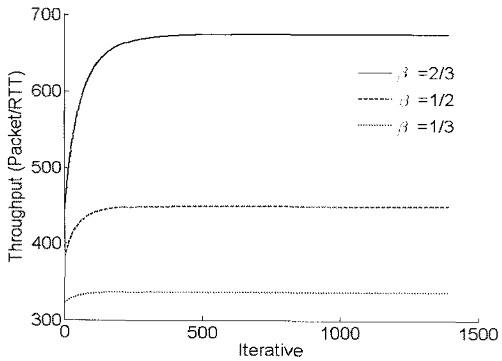


Fig. 6. Convergence of Throughput - Constant upper bound.

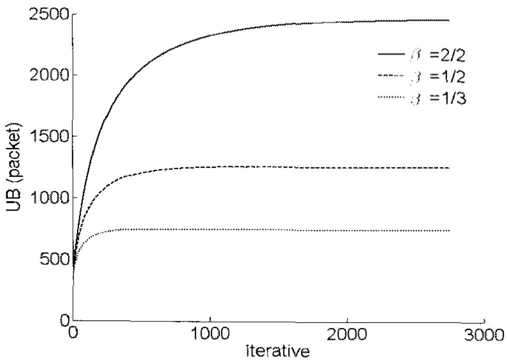


Fig. 7. Convergence of *UB*-Variable upper bound.

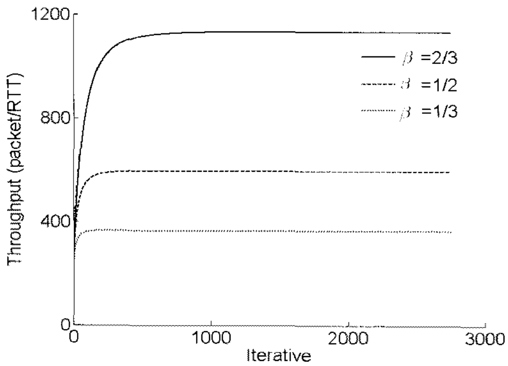


Fig. 8. Convergence of Throughput - Variable upper bound.

converges in Figure 6.

Similarly, we have the convergence results for variable congestion window upper bound in Figure 7 and for throughput in Figure 8 respectively.

5. RELATED WORKS

TCP [1] is one of the most important protocols which make the Internet popular in the world. In the Internet, a huge number of users share common resources, which leads to congestion. To avoid congestion, TCP uses AIMD mechanism whose operation is based on congestion window control. In AIMD, congestion window is additively increased by 1 if no loss is detected. When a packet loss happens, the congestion window is multiplicatively decreased by 1/2. This mechanism is based on an idea that each source slowly increases its data rate until congestion occurs and fast decreases its data rate to avoid congestion immediately. However, there is not any reason why 1 and $\frac{1}{2}$ are chosen for AIMD parameters. This problem was investigated in general AIMD [6], a generalization of traditional AIMD. In general AIMD, the additive increasing factor is not 1 but a general parameter α while the multiplicative decreasing factor is chosen as β.

Although TCP is a main factor which brings success to the Internet, it faces many challenges in wireless networks [1-5]. For example, in wireless network, packet loss mainly comes from environment noise or collision in a common channel. In this case, the TCP source considers that loss is due to congestion and reduces its congestion window. This is not a suitable behavior because it makes throughput unexpectedly decrease. To overcome the confusion between environment loss and congestion loss, each node in the transmission path may feedback the loss reason to its source [2,3]. Another method, which is proposed by Yamamoto *et al.* [5], is to limit congestion window by an upper bound. Their simulation result showed that the network throughput is improved by using this method. However, the question that how to limit congestion window has not absolutely solved because of lacking a systematic mathematical model. Thus, we develop this paper to answer the question.

5. CONCLUSION

This paper improves general AIMD mechanism to adapt it wireless environment where error rate is typically high. With presentation of an upper bound for congestion window, we can improve network throughput. We provide two scenarios: constant upper bound and variable upper bound. By applying gradient method, we find optimal policy for upper bound to achieve maximal network throughput.

REFERENCES

[1] M. Allman, V. Paxson and W. Stevens, *TCP congestion control*, RFC 2581, April, 1999.

[2] K. Leung and V. Li, "Transmission control protocol (TCP) in wireless networks: Issues, approaches, and challenges," *IEEE Communications Surveys and Tutorials*, vol.8, no.4, pp.64 79, Fourth Quarter 2006.

[3] K. Nahm, A. Helmy and C. Jay Kuo, "TCP over multi-hop 802.11 networks: Issues and performance enhancement," *ACM MobiHoc*, pp. 277-287, May 2005.

[4] S.M. El Ralabawy, A. Klemm and C. Lindemann, "TCP with adaptive pacing for multihop wireless networks," *ACM MobiHoc*, pp.288 299, May 2005.

[5] T. Yahmamoto, H. Tode and K. Murakami, "Enhanced TCP Congestion Control Realizing Higher Throughput and Inter-Session Fairness in Multihop Wireless Networks," *IEICE Trans. on Communications*, vol. E-91B, no. 7, pp. 2279 2286, July, 2008.

[6] Y. Yang and S. Lam, "General AIMD Congestion Control," Technical report, The University of Texas at Austin, May, 2000.

[7] O. Akan and I. Akyildiz, "ATL: An Adaptive Transport Layer Suite for Next-Generation Wireless Internet," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 5, pp. 802 817, June, 2004.

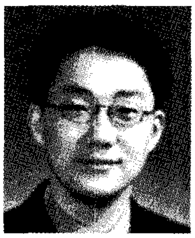
[8] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, 1999.

Bui Dang Quang

He received the B.S. degree in Information and Communication Systems from Hanoi University of Technology, in 2004 and the M.S. degree in Department of Electronic Telecommunication Engineering from Inje University. He is currently working towards his Ph.D. at Department of Information and Communication Systems, Inje University. His research interests are in Network Optimization and Robust Design.

Myeonggil Choi

He is an assistant professor t ChungAng University in Korea. He received the B.S. degree from Pusan National University and Ph.D. degree in Management Information Systems from Korea Advanced Institute of Science and Technology (KAIST) in 2004. From 1995 to 2005, he worked at Agency for Defense Department (ADD) as researcher and has worked for National Security Research Institute, Electronics and Telecommunications Research Institute (ETRI) in Korea. From 2005 to 2007, he served as an assistant professor at Inje University. His recent research issues include Information System Security Evaluation, E-Commerce Security and Information Security Management, and Industrial Security.



Won-Joo Hwang

He received the Ph.D Degree from Osaka University Japan in 2002. He received his bachelor's degree and M.S. degree in Computer Engineering from Pusan National University, Pusan, Republic of Korea, in 1998 and 2000. Since September 2002, he has been an assistance professor at Inje University, Republic of Korea. His research interests are in Network Optimization and Ubiquitous Sensor Networks.