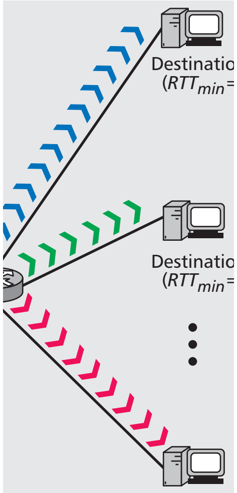
ACCEPTED FROM OPEN CALL

WIRELESS LOSS-TOLERANT CONGESTION CONTROL PROTOCOL BASED ON DYNAMIC AIMD THEORY

HIROKI NISHIYAMA, TOHOKU UNIVERSITY
NIRWAN ANSARI, NEW JERSEY INSTITUTE OF TECHNOLOGY
NEI KATO, TOHOKU UNIVERSITY



TCP, the de facto standard transport layer protocol providing reliable communication over IP networks, poses several significant performance issues. The authors examine the causes of these problems from the viewpoint of window control theory.

ABSTRACT

Recently, the use of Internet Protocol has rapidly expanded beyond the Internet, as evidenced, for example, by the construction of the next-generation network, empowering telecommunication networks by IP. A huge IP network is expected to emerge in the near future by means of convergence of various networks. However, Transmission Control Protocol, the de facto standard transport layer protocol providing reliable communication over such IP networks, poses several significant performance issues. The small buffer problem, unfair bandwidth allocation, and throughput degradation in wireless environments have been widely known issues in TCP communications. In this article we examine the causes of these problems from the viewpoint of window control theory. While TCP employs additive-increase multiplicative-decrease theory as a window control policy, the lack of flexibility of its static AIMD control is the basic cause for its performance degradation. After briefly reviewing TCP enhancements that utilize various modified AIMD control schemes, we introduce explicitly synchronized TCP, which employs a dynamic AIMD window control mechanism by employing feedback information from network nodes. By dynamically controlling AIMD procedures according to varying network conditions, ESTCP is able to achieve high performance even in hybrid wired/wireless networks.

INTRODUCTION

The tremendous and rapid development of wireless access technologies has facilitated easy connections to the Internet with broadband services. Wireless LAN, Worldwide Interoperability for Microwave Access (WiMAX), and third-generation (3G) cellular systems provide high-speed access to networks. The convergence of such networks is essential for achieving ubiquitous networking, and Internet Protocol (IP) is used for integrating these heterogeneous networks. In response to the broadbandization of the media access control (MAC) layer and the wide use of IP, the role of Transmission Control Protocol (TCP) in controlling the transmission rate over IP networks has also been expanded.

Although TCP is the de facto standard congestion control protocol in the Internet, its shortcomings in the heterogeneous environment are widely known and have been actively addressed by many researchers. Most existing congestion control protocols can be classified into two categories, additive-increase multiplicative-decrease (AIMD) and feedback control system (FCS), according to the theory used for rate control. AIMD approaches are traditional schemes that adopt AIMD theory for window control. Because static AIMD control (like in TCP) limits its utility, as described in the next section, the challenge is to design a dynamic AIMD control scheme that is able to adapt to a variety of network environments. We introduce a high-performance congestion control protocol, referred to as Explicitly Synchronized TCP (ESTCP), which is based on dynamic AIMD theory.

Meanwhile, several FCS approaches such as Explicit Control Protocol (XCP) [1] and Adaptive Congestion Protocol (ACP) [2] have recently been proposed. In these protocols a source node and network nodes exchange useful information for congestion control, and the source regulates its own window size according to the feedback from network nodes. As the window size of each flow is controlled based on the FCS theory, aggregate traffic can be stabilized and throughput of each flow becomes steady after a reasonable time lapse. In general, FCS approaches tend to achieve better performance than AIMD approaches because they can operate according to the actual traffic conditions observed at a bottleneck node. However, in FCS approaches there is a significant drawback involving the network nodes, which are required to handle the packet headers of upper layers other than the IP layer. It is not easy to read and write the corresponding TCP headers in network nodes when an IP payload is encrypted or encapsulated. For instance, the encryption mechanisms used in IP-in-IP tunneling and IP security (IPsec) popular in virtual private networks (VPNs) complicate the extraction of the IP payload. Although the IP option field may be applied in such situations, such use is generally

1536-1284/10/$25.00 © 2010 IEEE

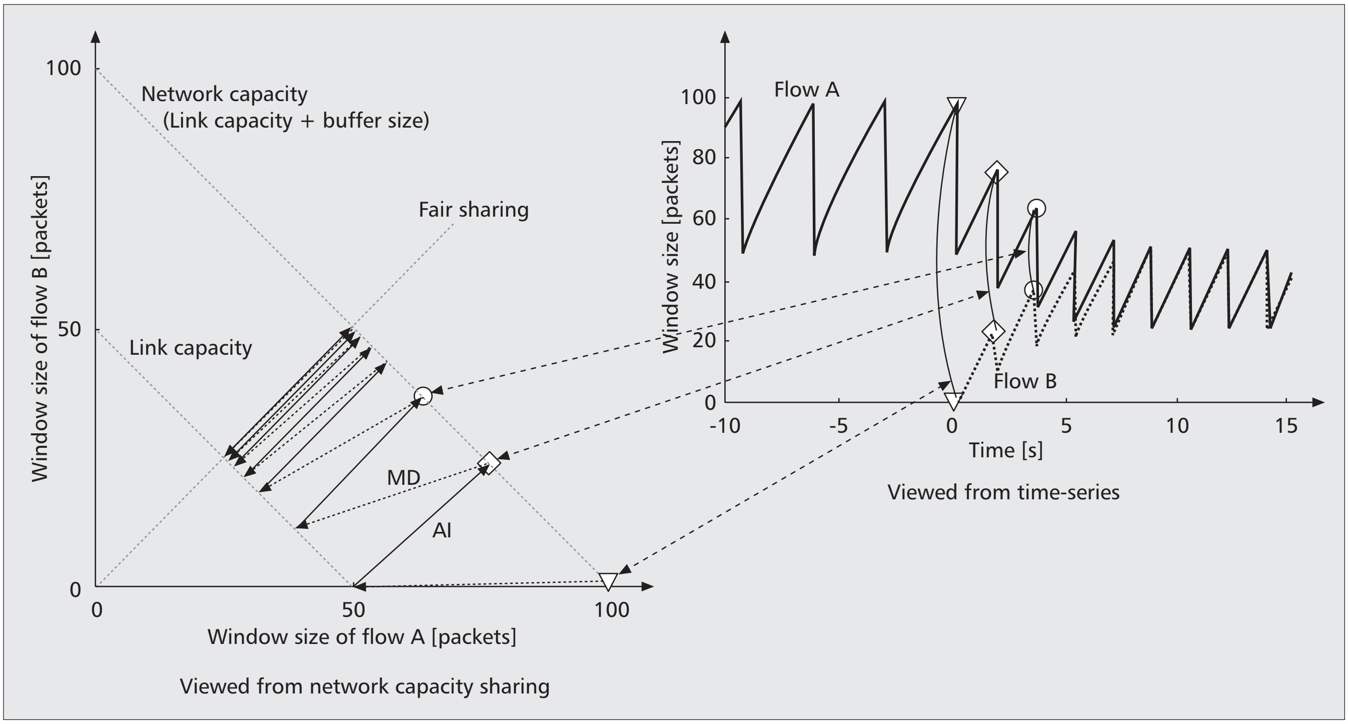


Figure 1. *Window convergence by AIMD control in TCP.*

avoided as it would substantially slow down switching. Therefore, like ESTCP, described later, information exchange between nodes should be done within the IP layer without invoking any additional fields.

In the following we primarily address AIMD approaches and do not discuss FCS approaches, which are beyond the scope of our subsequent coverage. In the next section issues of TCP window control with respect to static AIMD theory are reviewed. Then existing congestion control protocols by adapting AIMD control to improve the performance in wireless networks are discussed. Finally, ESTCP is introduced as an exemplary protocol based on dynamic AIMD designed from the traffic control perspective at network nodes.

ISSUES OF TCP RATE CONTROL

TCP employs a window-based rate control mechanism so as to be automatically clocked by acknowledgment (ACK) packets, which are paced depending on the bottleneck link speed. The TCP source agent is allowed to send as many data packets as the size of the sending window without ACKs. The maximum size of the sending window is limited by the receiver's advertised window, the size of which is defined by the TCP destination agent to prevent overflow in the receive socket buffer. With an adequate size of receive socket buffer, the sending window size is set to be the size of the congestion window (*cwnd*) computed at the source node. Hence, the congestion window control mechanism dictates the rate control in TCP.

AIMD WINDOW CONTROL MECHANISM

The congestion window control policy at a TCP source is quite different between the slow start phase and the congestion avoidance phase. In the former, corresponding to the period of connection startup or restart, the congestion window is exponentially increased to quickly ramp up the transmission rate. After that, the TCP source goes into the congestion avoidance phase where the congestion window grows linearly until a packet loss is detected. Upon the detection of a packet loss, the congestion window is immediately halved. TCP repeats these linear increase and immediate decrease in the congestion avoidance phase. Hence, the performance of TCP largely depends on the performance in the congestion avoidance phase.

The control of the increase and decrease of window size in the congestion avoidance phase is referred to as AIMD. Additive-increase (AI) is performed at the source whenever an ACK packet acknowledging the reception of new data is received. On the other hand, multiplicative-decrease (MD) is executed upon detecting a packet loss. AI and MD algorithms can be expressed as follows:

$$\text{AI: } cwnd \leftarrow cwnd + \frac{\alpha_{TCP}}{cwnd}$$

$$\text{MD: } cwnd \leftarrow \beta_{TCP} \cdot cwnd$$

where α_{TCP} and β_{TCP} are equal to 1 and 0.5 in units of a packet, respectively, for the stock TCP.

AIMD is introduced to ensure efficient utilization of network resources and fairness among competing flows sharing the same bottleneck. Figure 1 shows the ideal behavior of AIMD in a simple case where two flows share the bottleneck. Flow B initiates communication at time 0 when flow A has consumed the whole network capacity equal to 100 packets. The network capacity consists of bottleneck buffer space and end-to-end link capacity. The amount of end-to-end link capacity is given by the bandwidth-delay product (BDP). In the ideal case where the buffer size is equal to BDP, the aggregate window size of both flows never falls below the link capacity, as shown in Fig. 1, because the value of β_{TCP} is 0.5. After the window size is shrunk, both flows increase their window sizes at the same speed under the assumption that both flows have an equal round-trip time (RTT). At the moment when the amount of outstanding data exceeds network capacity, packets are lost due to buffer overflow, thus leading to the invocation of MD. By repeating AI and MD in this manner, the gap in the window size between flows becomes less and less. This is because the decreased amount of the window by MD is proportional to the absolute window size while the increasing speed in AI does not depend on its size and is the same in both flows. Thus, TCP attempts to ensure the efficiency and fairness in communications by adopting AIMD for its rate control.

Issues of Window Control Based on AIMD

Although AIMD is a simple and convenient mechanism to fairly share the bandwidth among competing flows, it makes many assumptions that are not always satisfied. For example, RTTs of flows are not equal in general; this leads to the unfair bandwidth allocation problem. On the other hand, there is no guarantee that the concerned router's buffer size is more than the value of BDP, thus causing the degradation of link utilization. This issue is widely known as the small buffer problem. To cope with these issues, AIMD parameters α_{TCP} and β_{TCP} need to be dynamically adjusted according to the situation at hand. How to estimate the network condition, and design an algorithm for controlling these parameters are the main focuses in this research area. Related work is outlined in the next section.

The drawbacks of AIMD include not only the difficulties in setting the suitable parameters but also the naive assumption that the change of the congestion window is synchronized over all flows by simultaneously experiencing a packet drop due to buffer overflow. Although the drop-tail queuing policy tends to be employed in almost all routers because of its simplicity, it does not guarantee that packet drops are distributed over all flows when a buffer is overflowed by traffic convergence. Some flows may seldom detect a packet drop and increase their sending rates, while other flows consistently experience packet loss and decrease their window sizes by running the MD algorithm. Half-synchronization derived from drop-tail queuing causes lower network utilization and promotes unfair bandwidth allocation. This issue is well known as the TCP synchronization problem. While Random Early Detection (RED) [3] can reduce the degradation of link utilization by randomly dropping packets, it almost completely breaks the synchronization, which is an essential feature of AIMD. To exploit the property of AIMD, it is preferred to synchronize all flows. As described later, ESTCP, by employing dynamic AIMD, can achieve high performance by ensuring synchronization over all concerned flows.

Non-congestive packet loss (i.e., link error-related loss) is also another key factor hampering AIMD control because AIMD does not take into account the occurrence of non-congestive losses. Bit error rates of wireless links are much higher than those of wireline ones because the wireless channel is not stationary and becomes worse due to interference, multipath fading, user mobility, and so on. When flows detect a link error-related loss, they cut down their sending rates by the MD algorithm since they misinterpret the cause of packet loss as being due to congestion. An unnecessary calling of MD leads to throughput degradation and ultimately poor link utilization. To avoid underutilization of the link capacity, it is essential to properly adjust the MD parameters or differentiate non-congestive losses from congestive ones.

TCP Enhancements for Wireless Networks

The poor TCP performance in wireless networks is due to the misunderstanding of the cause of packet losses. In TCP all packet losses are assumed to be caused by network congestion, thus leading to the reduction of window size by the MD algorithm. Therefore, the throughput of TCP can be improved if it is possible to distinguish link error-related losses from congestive ones. TCP Veno [4] and Jitter-based TCP (JTCP) [5] are typical examples of this type of solution. These schemes estimate the cause of packet losses from the changes in RTT, which reflect network congestion. By following the estimation, a source agent determines whether or not to call on MD for the loss. Another solution is to control the degree of MD in such a way as not to shrink the window more than required. Even if the MD algorithm is invoked for every packet drop, an appropriate MD parameter makes it possible to maintain high throughput. Actually, in most cases the window size is directly updated according to the network condition estimated at the source node without using the MD parameter because it is not easy to know the proper value of the MD parameter. TCP Westwood [6] and TCP New Jersey [7, 8] are examples that estimate the available bandwidth and accordingly update the window size. In addition, TCP New Jersey has a mechanism for specifying the cause of packet losses. Table 1 compares TCP enhancements for wireless networks.

In TCP all packet losses are assumed to be caused by network congestion, thus leading to the reduction of window size by the MD algorithm. Therefore, the throughput of TCP can be improved if it is possible to distinguish link error-related losses from congestive ones.

While we do not address cross-layer designs including MAC layer modification [9, 10] because our solution space is primarily constrained to the TCP and IP layers within the realm of the TCP/IP protocol suite, the cross-layer approach is certainly one of the effective means to enhance the performance of TCP in wireless environments.

TCP Veno

TCP Veno uses the amount of backlogged packets, N, buffered in the bottleneck queue as a network congestion indicator. N can be estimated from the latest congestion window size and RTT values monitored at the source node. RTTs are measured by using a millisecond resolution timestamp (TS) mechanism. In TCP Veno, AI and MD algorithms of the standard TCP are modified by using N and the predefined constant threshold, θ. The growing speed of the window size in AI is halved if N is equal to or greater than θ, because the link capacity is considered fully utilized in this

The motivation of ESTCP is to dynamically control AIMD parameters by adjusting them to network congestion. The AI parameter value is handled by a network node so as to stabilize traffic. In the equilibrium state, windows of all flows sharing the same bottleneck are synchronized.

	TCP Veno	JTCP	TCP Westwood	TCP New Jersey	ESTCP
Efficiency control	No	No	Yes	Yes	Yes
Fairness control	No	No	No	No	Yes
Loss differentiation	Yes	Yes	No	Yes	Available
Modified algorithm	AI and MD	MD	MD	MD	AI and MD
Additional mechanism	TS	TS	—	ECN and TS	ECN and TS

Table 1. *Comparison among loss-tolerant TCP enhancements based on AIMD.*

case. Otherwise, the window is increased as in the standard TCP. Similar to the AI algorithm, the cause of packet loss is estimated based on the magnitude relation between N and θ. N less than θ implies link error-related losses, and the value of the MD parameter is set to be 0.8. In other cases it is set to be 0.5. By using a larger MD parameter for non-congestive losses, throughput improvement can be expected. However, there is no basis for ensuring that the static value, 0.8, is always enough to fill the bandwidth capacity.

JTCP

JTCP is a congestion control protocol that distinguishes the cause of packet losses by observing the interarrival jitter, which is defined in the Real-Time Transport Protocol (RTP). As an increase in interarrival jitter implies an increase in queuing delay caused by network congestion, it can be used as an indicator of congestion. In JTCP, jitter ratio, *Jr*, is calculated from measured interarrival jitters. Upon detecting packet loss, *Jr* is used to determine whether or not to halve the window. With *Jr* value below (1/*cwnd*), JTCP regards the loss being caused by wireless link errors, and does not invoke MD. Although JTCP keeps its transmission rate if the loss is assumed to be non-congestive, it calls the same MD procedure for congestive losses as in standard TCP. Therefore, JTCP also exhibits the efficiency and fairness issues.

TCP Westwood

TCP Westwood is the most notable TCP variant, which aims to address random packet drops. A source node estimates the available bandwidth based on the interarrival time of ACK packets. In TCP Westwood, instead of the MD parameter, the value of the available bandwidth is used for setting a new window size in every MD procedure. By setting transmission rate to be the available bandwidth, TCP Westwood is able to achieve efficient link utilization and robustness to link error-related losses. However, its performance largely depends on the estimation accuracy. In fact, it has been widely reported that the ACK compression induced by congestion leads to overestimation of the available bandwidth, thus degrading the performance of TCP Westwood. While an enhancement of TCP Westwood, dubbed TCP Westwood+ [11], has been proposed to remove the negative effect of ACK compression, its fairness issue remains unsolved.

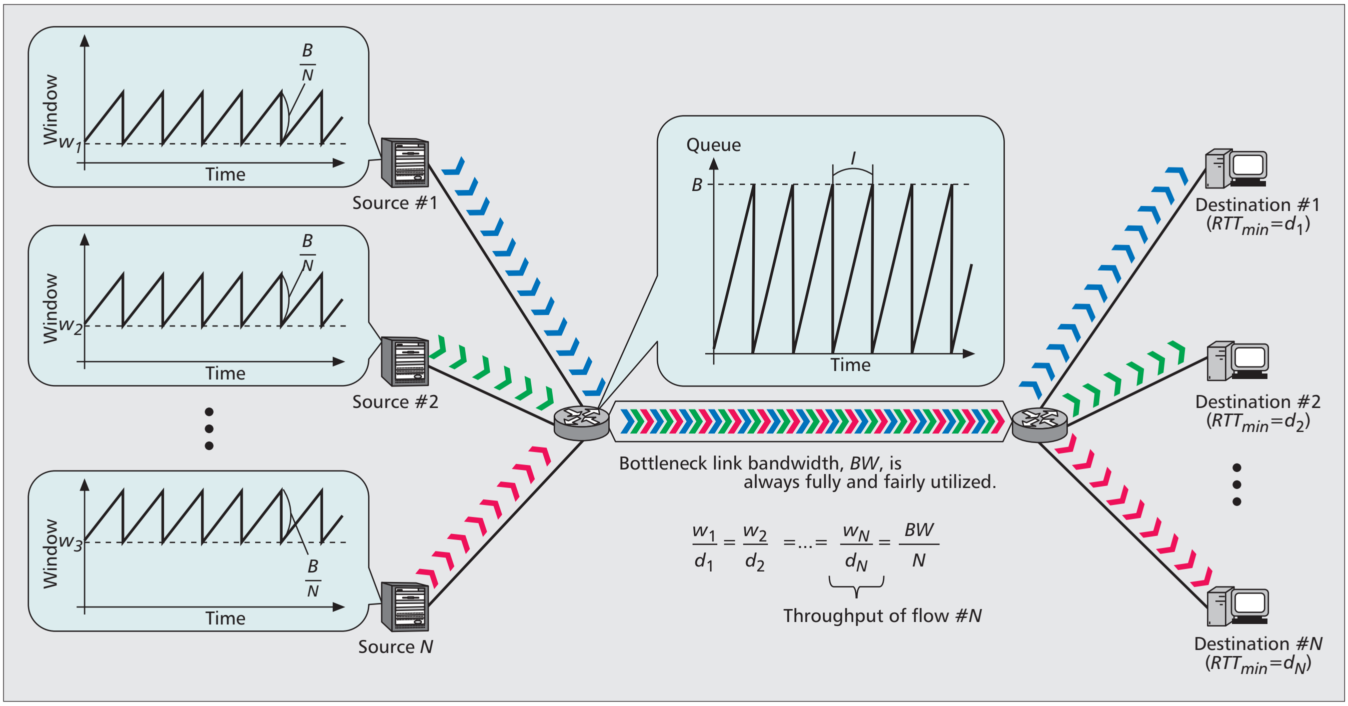
TCP New Jersey

In terms of estimating the available bandwidth and accordingly adjusting the window size in MD processing, TCP New Jersey is similar to TCP Westwood, but TCP New Jersey employs a more complex mechanism by using the TCP timestamp option to compute the available bandwidth more accurately. It should be noted that TCP New Jersey uses feedback from network nodes to acquire certain information on packet drops. In TCP New Jersey the key component, called the congestion warning (CW), is equipped in each network node. CW signals network congestion by using the explicit congestion notification (ECN) mechanism when buffer overflow may be expected. According to the feedback signal from CW, the source agent differentiates the wireless losses from the congestive ones. Window updating by MD is performed only upon confirmation from the CW signal that the network is congested. Because of loss differentiation by the CW mechanism and also availability of the intelligent bandwidth estimator, TCP New Jersey is able to outperform TCP Westwood. However, the fairness issue is still not resolved.

ESTCP

ESTCP Algorithm

The motivation of ESTCP is to dynamically control AIMD parameters by adjusting them to network congestion. The AI parameter value is handled by a network node to stabilize traffic. In the equilibrium state, windows of all flows sharing the same bottleneck are synchronized, as shown in Fig. 2. While the rate of increase of the window size in TCP depends on the flow's RTT, it is equal for all flows in ESTCP and simultaneously scaled by following the feedback from the bottleneck node. On the other hand, the MD parameter is independently adjusted by each source to keep full link utilization. By doing so, the amount of network traffic can be matched to bandwidth capacity, thus leading to almost zero buffer occupancy with synchronized MDs. The timing of running the MD algorithm in each flow is concentrically controlled by the bottleneck node to exactly synchronize all flows. Since the aggregate traffic passing through the bottleneck node is the superposition of window sizes of all traversing flows, the bottleneck queue occupancy demonstrates the change as a sawtooth wave. By



Window
Time
w_1
$\frac{B}{N}$
Source #1
Window
Time
w_2
$\frac{B}{N}$
Source #2
Window
Time
w_3
$\frac{B}{N}$
Source N
Queue
B
I
Time
Bottleneck link bandwidth, BW, is always fully and fairly utilized.
$\frac{w_1}{d_1} = \frac{w_2}{d_2} = \ldots = \underbrace{\frac{w_N}{d_N}}_{\text{Throughput of flow \#N}} = \frac{BW}{N}$
Destination #1 ($RTT_{min}=d_1$)
Destination #2 ($RTT_{min}=d_2$)
Destination #N ($RTT_{min}=d_N$)

Figure 2. *Window synchronization and buffer occupancy in the equilibrium state in ESTCP.*

combining these mechanisms, ESTCP is able to overcome the TCP problems exhibited in static AIMD. In the following we describe dynamic AIMD control in ESTCP. As it is obvious from the above, ESTCP consists of two key components: the window controller (WC), which executes the AIMD window adjustment at the source node, and the traffic controller (TC), equipped at the bottleneck node for scaling the rate of increase of the window size to appropriately stabilize the aggregate traffic.

To constantly maintain full utilization of the link capacity, the fluctuation range of the aggregate traffic must lie within the capacity of the bottleneck buffer. To derive the ideal value of the MD parameter, β_{ESTCP}, we consider the equilibrium state as shown in Fig. 2. In the equilibrium state all flows simultaneously and periodically experience packet drops due to buffer overflow, thus leading to invocation of MD, while throughputs of all flows do not change much and remain the same. The throughput at the moment prior to invoking MD can be expressed as the congestion window size, $cwnd_{cur}$, divided by the RTT value, RTT_{cur} (i.e., $cwnd_{cur}/RTT_{cur}$). RTT_{cur} is greater than the minimum value, $RTT_{\min}$, due to the queuing delay. On the other hand, the throughput at the moment just after MD is equal to $(\beta_{ESTCP} \cdot cwnd_{cur}/RTT_{\min})$ because the queuing delay is almost zero. According to the fact that the throughputs before and after invoking MD are the same in each flow, the ideal value of β_{ESTCP} can be defined as

$$\beta_{ESTCP} = \frac{RTT_{\min}}{RTT_{cur}},$$

where $RTT_{\min}$ denotes the minimum RTT measured since the beginning of the flow, and RTT_{cur} shows the RTT value prior to invoking the MD mechanism. The concept of this definition is similar to that of H-TCP [12]. To enable accurate measurements of RTT, the TCP timestamp option is enabled.

In protocols that adopt the ACK clocking mechanism like TCP, the rate of increase of the window size depends on the flow's RTT. A larger RTT results in slower growth of the congestion window because the congestion window can increase only upon receiving a new ACK packet. This is the prime cause of the unfair bandwidth allocation issue. The simplest solution to remove the effect of RTT on the rate of increase of the window size is to design the AI parameter proportional to the flow's RTT. ESTCP uses the following AI parameter:

$$\alpha_{ESTCP} = g^{-1} \cdot RTT_{\min},$$

where g is a scaling parameter. Although some window control schemes attempt to solve unfair bandwidth allocation by using the normalized RTT value in the AI processing, such as TCP Hybla [13] and Variable-Structure Congestion Control Protocol (VCP) [14], the parameter value corresponding to g is not properly adjusted according to the network condition. The noticeable feature of ESTCP in its window control mechanism is dynamically increasing or decreasing g by following the network congestion state.

In ESTCP all flows are synchronized by feedback from the same bottleneck node. In addition, the aggregate traffic fluctuates within the buffer space. Hence, it is indeed important to control and stabilize the traffic changes. Fortunately, traffic can easily be handled by controlling g. A small value of g should be avoided because it makes traffic increase rapidly, leading to significant packet drops and breaking synchronization. On the other hand, large values of g prolong the invocation of MDs by the buffer

In ESTCP, all flows are synchronized by the feedback from the same bottleneck node. In addition, the aggregate traffic fluctuates within the buffer space. So, it is indeed important to control and stabilize the traffic changes.

overflow; thus, the system takes a long time to reach the equilibrium state because fair sharing of bandwidth can be achieved by repeating AI and MD alternately. From the above discussion, ESTCP tries to control g so as to converge the duration between consecutive MDs to a suitable value of D. In this article we refer to the period between consecutive MDs as *phase*. The duration of the phase measured at the discrete time n is referred to as I_n. The parameter g is frequently updated by the following equation:

$$g_{n+1} = g_n + \gamma(D - I_n).$$

The value of the system parameter, γ, is determined from the stability analysis of the feedback system. Although the value of D needs to be much higher than the RTT values of the corresponding flows so that the system may reach an equilibrium state, exceedingly high values of D may contribute to a significant delay in reaching the equilibrium level. When a buffer overflow occurs, TC calculates I_n and updates g by the above equation. A new value of g is sent to WC from TC, and WC accordingly adjusts its window size. Since all flows follow the same signal from the same TC, aggregate traffic can be controlled as required by TC.

At the same time as the notification of a new value of g, TC informs WC about the shift of phases. The phase shift induces the execution of MD at the source node and maintains the synchronization of flows. Incidentally, it is possible to estimate the cause of packet losses at source nodes by observing the occurrence of the phase shift, which implies network congestion. Since any buffer overflow due to congestion definitely sets off a phase shift, packet losses without involving the phase shift are supposed to be noncongestive (i.e., due to wireless link errors). However, it is unsure whether WC receives the signal indicating phase shifting before detecting a congestive packet loss. So from the conservative point of view, ESTCP does not enable us to distinguish losses, and it executes the MD algorithm for all packet losses. Note that executions of MD never harm the throughput in wireless environments since the MD parameter is adequately adjusted in order not to shrink the window more than required.

Performance of ESTCP

To evaluate the performance of ESTCP, we have conducted extensive simulations by using Network Simulator version 2 (NS2). For illustrative purposes, we considered two different scenarios to investigate the tolerance for congestion and wireless losses. In both scenarios the simple dumbbell topology as shown in Fig. 2 is used, and the bottleneck bandwidth, BW, is equal to 100 Mb/s. In ESTCP the value of D is set to 5 s, which is far greater than flows' RTTs. Standard TCP and TCP Westwood+ are used for comparisons.

In the first scenario, the bottleneck buffer size, B, is set to the half of BDP, and the RTTs of each flow are distributed within the range between 40 and 200 ms; this emulates the condition to incite inefficient link utilization caused by the small buffer problem and unfair bandwidth allocation due to different RTTs. The number of flows, N, varies from 10 to 100 in order to evaluate the tolerance for network congestion. Figure 3a shows the bottleneck link utilization and fairness index, which is defined as $(\sum_i x_i)^2/(N \sum_i x_i^2)$, where x_i is the throughput of flow i. The fairness index ranges from 0.0 to 1.0, in which a larger value implies better fairness. It is clear that standard TCP employing the static AIMD algorithm cannot efficiently utilize the bandwidth capacity because the window size necessarily decreases because of the unsuitable constant MD parameter. In addition, the results show that static AIMD, designed to fairly share network resources, is not able to accomplish the objective if flows have different RTTs. Although TCP Westwood+ provides more efficient link utilization than standard TCP, it still cannot resolve the unfair bandwidth allocation issue. It should be noted that only ESTCP achieves both almost 100 percent link utilization and almost perfect fair bandwidth allocation.

In the second scenario, the last hop to each destination is configured as an error-prone link emulating a wireless link in order to evaluate the tolerance for wireless losses. To remove the impact of factors other than wireless losses, B is set to BDP, N is equal to 1, and the RTT of the flow is fixed to 100 ms. We measured throughputs with different packet error rates (PERs) between 10^{-7} and 10^{-2}. Figure 3b depicts throughput degradations in each protocol in wireless environments. Although TCP Westwood+, developed to cope with random packet drops due to lossy wireless links, drastically improves the performance of TCP, ESTCP achieves higher throughput than that of TCP Westwood+, even around PER = 10^{-3}. It is evident that ESTCP achieves high link utilization, fair bandwidth allocation, and error robustness by employing dynamic AIMD.

Implementation of ESTCP

In ESTCP two pieces of information, the value of g and the occurrence of phase shifting, need to be supplied to all sources from a network node at the same time. If it is allowed to use any field in a packet header for carrying this information, ESTCP obviously performs well. However, the use of an option field or an additional field is not preferred even in the IP header, while the use of a TCP header should be avoided as mentioned earlier.

In response to the above requirements, ESTCP can adopt a probabilistic packet marking scheme by using an ECN-like framework as a mechanism to convey information. In ESTCP two kinds of information, the phase shift timing and the value of g, need to be transmitted to each source from the bottleneck. Figure 4 shows how such information can be transmitted by adopting the probabilistic packet marking scheme using an ECN-like mechanism. ECN bits of each data packet are set at the bottleneck, copied into an ACK packet at the destination, and then transmitted to the source by the ACK packet. To detect the occurrence of phase shift, different combinations of ECN bits (i.e., 00 and 01 or 10 and 11) are used in two successive phases. In the case of Fig. 4, with $k = 0, 1, 2, \ldots$, the combination of 00 and 01 is used in $2k$th phases while the combination of 10 and 11 is used in $(2k + 1)$th phases. Since different combinations

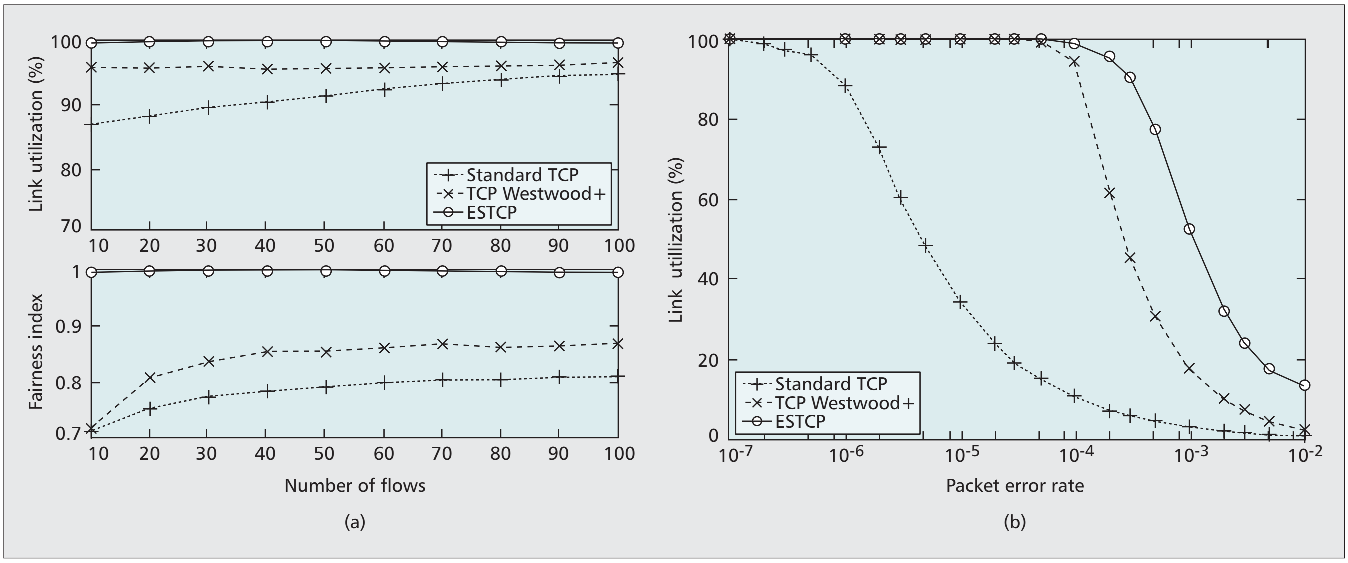


Figure 3. *Performance comparison.*

of bits are used in two successive phases, each source can detect the occurrence of phase shift by monitoring the bits in arriving ACK packets. On the other hand, in the *i*th phase the value of g_i is transmitted from the bottleneck to the source by using the probabilistic packet marking scheme, where marking probability p_i is determined at the bottleneck by the one-to-one mapping function, f, between g and p. At the source, the estimated value of g_i, referred to as g_i^e, can be obtained by gathering packets and estimating the marking probability, p_i^e, during the phase. The estimation of the marking probability can be done by using simple formulas, as shown in Fig. 4 where $n(x)$ stands for the number of packets marked with x. While the information feedback via the ECN field makes ESTCP more acceptable for the current TCP/IP framework, it is uncertain whether it can perform well in multiple bottleneck environments. This will be our future work from the aspect of implementation.

In ESTCP fairness among competing flows sharing the same bottleneck is ensured because window control in all flows is handled by the same TC at the bottleneck. In other words, it is very difficult to ensure fairness between ESTCP flows and non-ESTCP flows such as other TCP variants and nonresponsive traffic such as User Datagram Protocol (UDP). However, by using per-class queuing based on the protocol type, it is possible to prevent such background traffic from degrading the performance of ESTCP.

Conclusion

The purpose of this article is to point out the potential of dynamic AIMD window control. In the past, the drawbacks of TCP due to its inflexible and static AIMD control have been addressed. Also, many congestion control protocols employing modified AIMD have been proposed. However, drastic performance improvement is not expected without support from network nodes, which can directly observe traffic conditions. This is the motivation of developing a dynamic AIMD theory. The distinctive feature of dynamic AIMD control is the cooperation between the AIMD window controller at the source side and the traffic controller at the network side. The results of extensive simulations confirm the robust performance of ESTCP employing dynamic AIMD control. While it is clear that assistance from the network side dramatically enhances performance, additional packet headers or increased complexity in the protocol stack should be avoided, as not only the performance advantage but also easy implementation over IP is essential to a new congestion control protocol. Future endeavors should be dedicated to developing a high-performance and feasible transmission control protocol compatible with the present TCP/IP protocol suite.

References

[1] D. Katabi, M. Handley, and C. Rohrs, "Congestion Control for High Bandwidth-Delay Product Networks," *ACM SIGCOMM Comp. Commun. Rev*., vol. 32, no. 4, Oct. 2002, pp. 89–102.

[2] M. Lestas *et al*., "Adaptive Congestion Protocol: A Congestion Control Protocol with Learning Capability," *Comp. Net*., vol. 51, no. 13, Sept. 2007, pp. 3773–98.

[3] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Trans. Net*., vol. 1, no. 4, Aug. 1993, pp. 397–413.

[4] C. P. Fu and S. C. Liew, "TCP Veno: TCP Enhancement for Transmission over Wireless Access Networks," *IEEE JSAC*, vol. 21, no. 2, Feb. 2003, pp. 216–28.

[5] E. H.-K. Wu and M.-Z. Chen, "JTCP: Jitter-Based TCP for Heterogeneous Wireless Networks," *IEEE JSAC*, vol. 22, no. 4, May 2004, pp. 757–66.

[6] C. Casetti *et al*., "TCP Westwood: End-to-End Congestion Control for Wired/Wireless Networks," *Wireless Net*., vol. 8, no. 5, Sept. 2002, pp. 467–79.

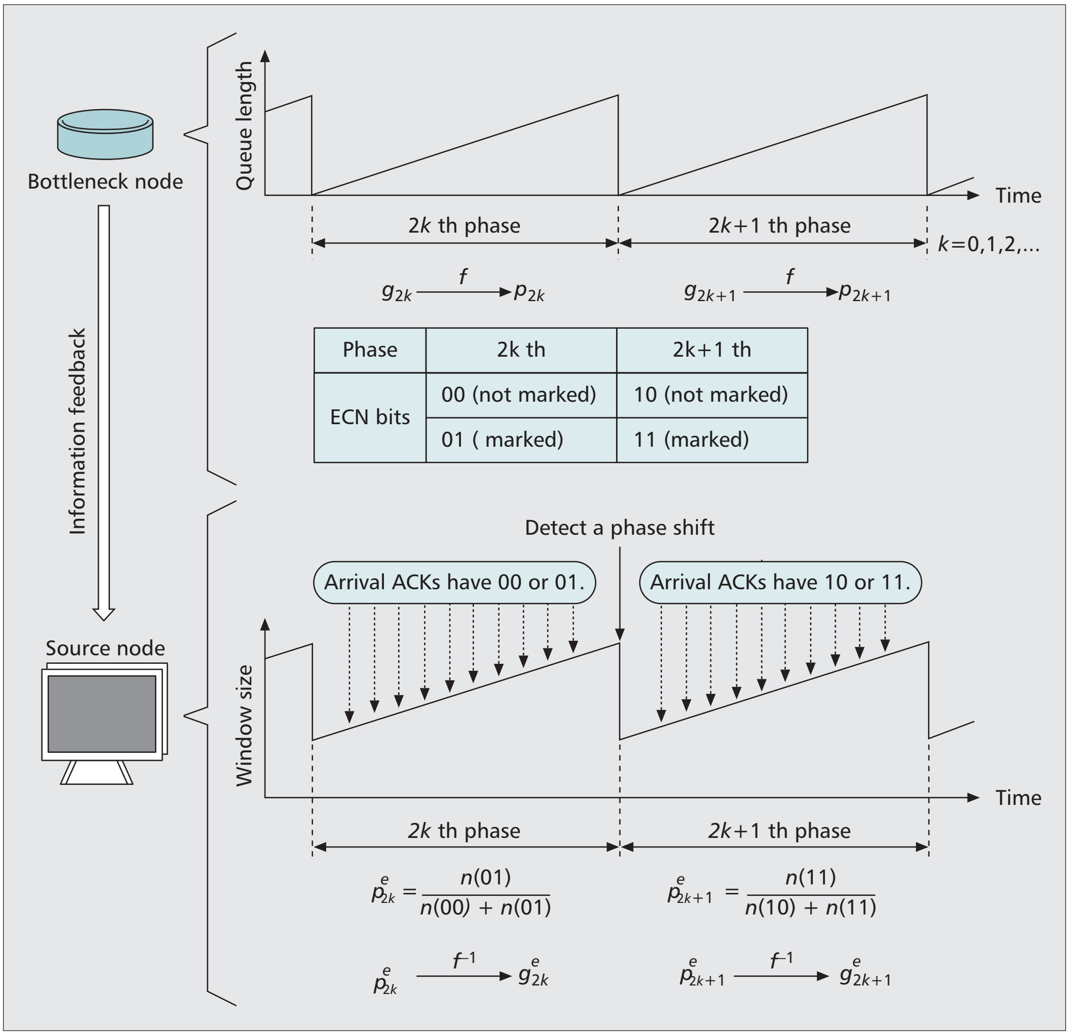
[7] K. Xu, Y. Tian, and N. Ansari, "Improving TCP Performance in Integrated Wireless Communications Networks," *Comp. Net*., vol. 47, no. 2, Feb. 2005, pp. 219–37.

[8] K. Xu, Y. Tian, and N. Ansari, "TCP-Jersey for Wireless IP Communications," *IEEE JSAC*, vol. 22, no. 4, May 2004, pp. 747–56.

[9] H. Balakrishnan *et al*., "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," *IEEE/ACM Trans. Net*., vol. 5, no. 6, Dec. 1997, pp. 756–69.

[10] H. Wu *et al*., "Performance of Reliable Transport Protocol over IEEE 802.11 Wireless LAN: Analysis and Enhancement," *Proc. IEEE INFOCOM*, New York, NY, June 2002, pp. 599–607.

While it is clear that the assistance from the network side dramatically enhances the performance, additional packet headers or increased complexity in protocol stack should be avoided, i.e., not only performance advantage but also easy implementation over IP are essential to a new congestion control protocol.


Bottleneck node
Queue length
Time
2k th phase
2k+1 th phase
k=0,1,2,...
$g_{2k} \xrightarrow{f} p_{2k}$
$g_{2k+1} \xrightarrow{f} p_{2k+1}$
Information feedback
Source node
Detect a phase shift
Arrival ACKs have 00 or 01.
Arrival ACKs have 10 or 11.
Window size
Time
2k th phase
2k+1 th phase
$p^e_{2k} = \frac{n(01)}{n(00) + n(01)}$
$p^e_{2k+1} = \frac{n(11)}{n(10) + n(11)}$
$p^e_{2k} \xrightarrow{f^{-1}} g^e_{2k}$
$p^e_{2k+1} \xrightarrow{f^{-1}} g^e_{2k+1}$

Phase	2k th	2k+1 th
ECN bits	00 (not marked)	10 (not marked)
	01 (marked)	11 (marked)

Figure 4. *Information feedback by probabilistic packet marking scheme using ECN bits in ESTCP.*

[11] S. Mascolo *et al.*, "Performance Evaluation of Westwood+ TCP Congestion Control," *Performance Evaluation*, vol. 55, no. 1/2, Jan. 2004, pp. 93–111.

[12] R. N. Shorten and D. J. Leith, "H-TCP Protocol for High-Speed Long Distance Networks," *Proc. Protocols Fast Long-Distance Net.*, Argonne, IL, Feb. 2004.

[13] C. Caini and R. Firrincieli, "TCP Hybla: A TCP Enhancement for Heterogeneous Networks," *Int'l. J. Satellite Commun. Net.*, vol. 22, no. 5, Aug. 2004, pp. 547–66.

[14] Y. Xia *et al.*, "One More Bit Is Enough," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 35, no. 4, Oct. 2005, pp. 37–48.

BIOGRAPHIES

HIROKI NISHIYAMA [M] (bigtree@it.ecei.tohoku.ac.jp) received his M.S. and Ph.D. in information science from Tohoku University, Japan, in 2007 and 2008, respectively. He also worked as a research fellow of the Japan Society for the Promotion of Science (JSPS) for one and a half years starting in 2007. He has been an assistant professor at the Graduate School of Information Sciences, Tohoku University since October 2008. He received the IEEE Sendai Section Student Award, "The Best Paper Prize," in December 2006. He has been engaged in research on congestion control, transport layer protocols, ad hoc and sensor networks, and network security. He is a member of the Information Processing Society of Japan (IPSJ).

NIRWAN ANSARI [F] is a professor of electrical and communications engineering at New Jersey Institute of Technology. His current research focuses on various aspects of broadband networks and multimedia communications. He has contributed over 300 technical publications. He is a Senior Technical Editor of *IEEE Communications Magazine*, and also serves on the Advisory/Editorial Boards of five other journals. He is an IEEE Communications Society Distinguished Lecturer.

NEI KATO [SM] received his M.S. and Ph.D. degrees from Tohoku University, Japan, in 1988 and 1991, respectively. He has been working with Tohoku University since then and is currently a full professor in the Graduate School of Information Sciences. He has been engaged in research on computer networking, wireless mobile communications, image processing, and neural networks. He has published more than 130 papers in journals and peer-reviewed conference proceedings. He has served as a symposium cochair for GLOBECOM '07 and ChinaCom '08, and TPC member for a large number of IEEE international conferences, including ICC, GLOBECOM, WCNC, and HPSR. He is a technical editor of *IEEE Wireless Communications* since 2006, an editor of *IEEE Transactions on Wireless Communications* since 2008, and has been a co-guest editor for *IEEE Wireless Communications* (Special Issue on Wireless Communications for E-Healthcare.) He is a co-recipient of the 2005 Distinguished Contributions to Satellite Communications Award from the IEEE Communications Society, Satellite and Space Communications Technical Committee, the co-recipient of the FUNAI information Science Award, 2007, and the co-recipient of the 2008 TELCOM System Technology Award from Foundation for Electrical Communications diffusion. He is serving as an expert member of Telecommunications Council, Ministry of Internal Affairs and Communications, Japan. He is a member of the Institute of Electronics, Information, and Communication Engineers (IEICE).