This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2010 proceedings.

Extensions of VCP to Enhance the Performance in High BDP and Wireless Networks

Yoshihiro Ikeda†, Hiroki Nishiyama†, Nirwan Ansari‡, Yoshiaki Nemoto*, and Nei Kato†
†*Graduate School of Information Sciences, Tohoku University, Japan
‡Advanced Networking Laboratory, ECE Department, New Jersey Institute of Technology, USA
Email: †{yosshi, bigtree, kato}@it.ecei.tohoku.ac.jp, ‡Nirwan.Ansari@njit.edu, *nemoto@nemoto.ecei.tohoku.ac.jp

***Abstract*—While Transmission Control Protocol (TCP) is the most popular transport protocol used in terrestrial networks, its performance is not adequate in wireless networks with long delay, e.g., satellite networks. Though some improvements of TCP and new transport protocols have been proposed, we focus on Variable-structure congestion Control Protocol (VCP) designed for high Bandwidth Delay Product (BDP) networks. In VCP, Explicit Congestion Notification (ECN) is used to generate the feedback signal from routers to sources in order to notify the utilization ratio of a bottleneck link. By adjusting its congestion window according to the network traffic conditions, VCP is able to achieve high link utilization even in high BDP networks. However, VCP requires a long time to fill the link capacity due to its more conservative window control mechanism than the slow start phase in TCP. In addition, throughput is unnecessarily degraded in VCP due to packet losses in wireless environments. In this paper, to address these issues, we propose two extensions of VCP, namely Bandwidth-Independent Start-up Extension (BISE) and Wireless Loss Tolerant Extension (WLTE). BISE can quickly increase its congestion window in the start-up phase. WLTE can maintain high throughput in wireless environments. The performance of the proposed schemes is evaluated through computer simulations. The results demonstrate that the proposed schemes dramatically improve the performance of VCP in the initial phase and also in wireless environments.**

I. Introduction

The convergence of different kinds of networks, such as wired, wireless, and satellite networks, by using the Internet Protocol (IP) allows us to easily access to the Internet from anywhere and at anytime. In addition, the broadbandization of such networks provides us with an opportunity to receive high-speed communication services. However, it is widely recognized that the performance of Transmission Control Protocol (TCP), a de-facto standard transport layer protocol, can be degraded in networks with high Bandwidth Delay Product (BDP) and/or error-prone links. The cause of TCP performance degradation is attributed to its window control mechanism. In the congestion avoidance phase, TCP decreases its congestion window by half upon detecting a packet loss regardless of whether the loss is due to network congestion or wireless errors. Therefore, TCP takes a long time to recover its throughput after each packet loss in high BDP environments, and cannot maintain high throughput in wireless environments with high Bit Error Rates (BERs).

Although many TCP variants and non-TCP transport layer protocols have already been proposed to improve the performance of TCP, we focus on Variable-structure congestion Control Protocol (VCP) [1] because of its high performance and potential. Since the VCP's window reduction rate upon detecting network congestion is much less than that of TCP, VCP is able to keep high throughput even in high BDP networks. In addition to this advantage, it is not so difficult to enhance the VCP performance in error-prone networks by introducing a new mechanism into its window control method as described later in details. However, VCP also has its shortcomings. For instance, the speed of increasing the congestion window in the start-up phase is much less than that of TCP's slow start, which means that VCP requires a long time to fill the link capacity. Accordingly, in this paper, we propose a new window control algorithm to quickly increase the throughput in the start-up phase.

The remainder of this paper is structured as follows. In Section II, major existing congestion control protocols including VCP are introduced. The detailed description on VCP's window control algorithm is provided in Section III. The proposed algorithm is described in Section IV and its performance is evaluated by computer simulations in Section V. Section VI concludes this paper.

II. Related Works

As mentioned at the previous section, we here focus on two major issues in TCP; one is the poor throughput in environments with highspeed and/or long propagation delays, and the other is the throughput degradation in wireless environments with high error rates. To cope with these issues, tremendous amount of research has been carried out and many congestion control protocols have been proposed.

HighSpeed TCP (HSTCP) [2], Binary Increase Congestion Control (BIC) [3], and TCP Hybla [4] are examples of protocols designed for high BDP networks. In HSTCP, the values of TCP parameters dominating the increasing speed and reduction rate of window size are adjusted by using a Round Trip Time (RTT) value and a packet drop rate so that the appropriate throughput can be achieved even in ten gigabit networks. BIC is also developed to efficiently utilize network capacity even in fast and long-distance networks. After cutting down the congestion window to avoid additional network congestion, BIC invokes binary increase algorithm which increases congestion window more aggressively than the standard TCP, thus making it possible to quickly recover the throughput. TCP Hybla proposed for satellite networks is designed to remove the impact of RTT on the performance. In cases of ACK-clocked protocols like TCP, the speed of increasing the congestion window depends on the RTT value of each flow; a small RTT contributes to rapid increase while a large RTT leads to relatively slower increase, which is not preferred in satellite networks having long propagation delays. Moreover, the difference in the speed of growing the congestion window among competing flows sharing the same bottleneck link results in the unfair bandwidth allocation, which is referred to as the RTT unfairness issue. TCP Hybla can mitigate these two issues by modifying the TCP's window control algorithms in both the slow start and congestion avoidance phases so as to counteract the effect of RTT.

On the other hand, enhancing the performance of TCP in high error rate is also a significant issue. While many schemes have been developed to solve this problem, we introduce two notable instances, TCP Westwood [5] and TCP New Jersey [6], [7]. To keep high throughput even in error-prone networks, TCP Westwood avoids reducing congestion window more than necessarily in the fast recovery process. Since TCP Westwood has an intelligent algorithm to estimate the available bandwidth based on the inter-arrival time of ACK packets, throughput can be appropriately maintained by setting its congestion window according to the estimated available bandwidth. Similar to TCP Westwood, TCP New Jersey also employs an Available Bandwidth Estimator (ABE) to adjust its congestion window in accordance with the network condition. However, there is a significant difference between these schemes, i.e., only TCP New Jersey has an advanced mechanism, referred to as Congestion Warning (CW), to notify an occurrence of network congestion from intermediate nodes to sources. Appropriate throughput can be achieved by distinguishing non-congestive losses from the congestive ones according to the feedback by CW and by adjusting congestion window based on the estimation by ABE.

A new approach to drastically improve the performance of TCP is to move away from the end-to-end principle. For instance, eXplicit Control Protocol (XCP) [8] is a well-known non-TCP transport layer protocol in which intermediate nodes play an important role in controlling traffic. The XCP router informs XCP senders the suitable amount of increase or decrease in the congestion window, and each XCP sender updates its congestion window accordingly. Since the XCP router employs an efficient and fair controller to fully utilize the network capacity and to equally assign the limited bandwidth to all competing flows, it mitigates the two major issues in TCP, i.e., inefficient link utilization and unfair bandwidth allocation. In addition, Proportional XCP (P-XCP) [9], an enhanced variety of XCP, is able to accomplish high throughput even in high BER environments while the original XCP has no mechanism to deal with wireless error-related losses. However, most of the network supported approaches such as XCP require installing additional complex functional components into intermediate nodes and inserting additional bits in the packet header, thus they are not readily deployed.

To achieve high performance as much as that of XCP without requiring a significant modification to the current networks, VCP has been developed. While the performance of VCP is slightly lower than that of XCP, VCP is superior to XCP in terms of feasibility because VCP adopts the Explicit Congestion Notification (ECN) [10] mechanism to convey information from routers to sources instead of introducing additional fields in the packet header. Furthermore, VCP has further potential to enhance its tolerance for wireless error-related losses with just a few modifications of the congestion window control mechanism. Meanwhile, the excessive conservative behavior in the ramp-up period, i.e., slower increase of throughput than the slow start in TCP, is a remaining issue. Therefore, in this paper, we focus on the significant potential of VCP to achieve high performance even in high BDP and error-prone environments, and aim at developing a VCP enhancement to mitigate the weak start-up problem.

TABLE I
ALGORITHMS AND ECN ASSIGNMENTS IN VCP

Network condition	ECN bits	Window control algorithm
overload ($1.0 \leq \rho$)	11	MD
highload ($0.8 \leq \rho < 1.0$)	10	AI
lowload ($0 \leq \rho < 0.8$)	01	MI

III. CONGESTION CONTROL MECHANISM OF VCP

A VCP source adjusts its congestion window according to the network load condition which is informed by the VCP router via ECN. In VCP, three different network states are defined, namely, low load, high load, and overload conditions. These network conditions are based on the link utilization estimated at each router as listed in Table I. In order to differentiate these network conditions, different ECN bits are assigned to the different conditions. If the network is overloaded, the VCP source quickly decreases its sending rate by invoking Multiplicative Decrease (MD). On the other hand, when the network is in low or high load conditions, the VCP source increases its congestion window by using Multiplicative Increase (MI) or Additive Increase (AI), respectively. VCP routers always monitor input traffic and periodically estimate link utilization, ρ, for every time interval, t_ρ, by using the following equation:

$$\rho = \frac{\lambda + \kappa \cdot q}{\gamma \cdot C \cdot t_\rho}, \tag{1}$$

where λ and q denote the amount of input traffic and the persistent queue length during the period, t_ρ, respectively. C indicates the real link capacity and γ represents the control parameter for determining the apparent link capacity. κ is also the parameter for controlling how quickly the persistent queue length can be reduced to zero. Note that VCP attempts to achieve an empty queue. In Reference [1], κ and γ are set to 0.5 and 0.98, respectively. Based on the value of ρ, the VCP router recognizes the current network condition and accordingly performs packet marking. VCP sources can infer the network condition by reading the ECN bits in the ACK packets. For each network condition, the congestion window is updated as follows:

$$\begin{aligned} \text{MD:} \quad cwnd(t+\delta t) &= \beta \times cwnd(t) & (2)\\ \text{AI:} \quad cwnd(t+rtt) &= cwnd(t) + \alpha_s & (3)\\ \text{MI:} \quad cwnd(t+rtt) &= (1+\varepsilon_s) \times cwnd(t), & (4) \end{aligned}$$

where $cwnd(t)$ denotes the congestion window size at time t. β, α_s, and $(1+\varepsilon_s)$ are MD, AI, and MI parameters, respectively. Although the congestion window is smoothly incremented over one RTT in AI and MI, the reduction of congestion window by MD is suddenly executed upon receiving a signal implying that the network is overloaded. To eliminate the difference in the increase rate of congestion window among flows having different RTT values, α_s and $(1+\varepsilon_s)$ are formulated as follows:

$$\begin{aligned} \alpha_s &= \alpha \times (rtt/t_\rho)^2 & (5)\\ \varepsilon_s &= (1+\varepsilon)^{rtt/t_\rho} - 1, & (6) \end{aligned}$$

where α and $(1+\varepsilon)$ are AI and MI parameters before eliminating it. The recommended values of constant parameters, α, ε, and β, are 1.0, 0.0625, and 0.875, respectively [1]. Although VCP can potentially utilize network capacity even in high BDP environments, there are a few remaining issues. First, it takes a long time to satisfy the fair share of link capacity. While we do not address this issue in this paper, several enhancements of VCP, such as Extended VCP (EVCP) [11] and VCP-Fast Convergence (VCP-FC) [12], have been proposed to address this issue. The second issue is the excessively slow window increase in the MI phase. Since the impact of this issue becomes more substantial in high BDP environments such as satellite networks, we aim at developing the window control scheme to quickly increase transmission rates without impairing advantages of the original VCP. We also propose a simple and effective modification to improve the performance in wireless environments.

IV. Envisioned Extensions of VCP

In order to enhance the performance of VCP, two functional extensions to the original VCP are proposed. The first extension consists of an intelligent window increase mechanism called Bandwidth-Independent Start-up Extension (BISE). The second extension is a simple modification of the response to packet losses due to wireless errors, referred to as Wireless Loss Tolerant Extension (WLTE).

A. Bandwidth-Independent Start-up Extension (BISE)

BISE is adopted in the low-load condition instead of executing the MI algorithm. The key concept of BISE is to generate medium window increase rates between AI and MI by mixing them in the millisecond time scale, referred to as Hybrid Increase (HI). The mixing ratio is controlled by the traffic controller at the router side so that the aggregate traffic traversing the router equal to the link capacity. Table II is the assignment of ECN bits and window control algorithm in each condition in the case of VCP with BISE.

TABLE II
Algorithms and ECN assignments in VCP with BISE

Network condition	ECN bits	Window control algorithm
overload ($1.0 \le \rho$)	11	MD
highload ($0.8 \le \rho < 1.0$)	10	AI
lowload ($0 \le \rho < 0.8$)	01, 10	HI

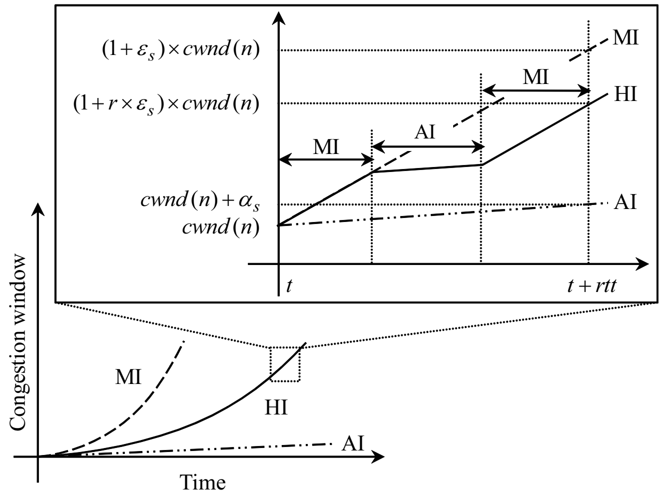


Fig. 1. Congestion window increase by HI

1) Hybrid Increase Algorithm: Although the reason behind the slow start-up of VCP in low-load conditions is the rather small value of the MI parameter, a large value of this parameter may lead to sudden increase of queue length. This may, in turn, result in numerous congestive losses as observed in the slow-start phase of TCP. To overcome this dilemma, the MI parameter needs to be dynamically adjusted according to the link utilization, i.e., it should be set to a large value for low utilization and a small one for high utilization. However, it is difficult for VCP sources to adjust the MI parameter according to link utilization because they cannot get detailed information on link utilization except whether the link is low-loaded, high-loaded, or overloaded. On the other hand, while VCP routers are able to easily estimate accurate link utilization, it is impossible to represent any value by only two bits of ECN. So, dynamically adjusting the MI parameter is impractical.

One solution to achieve various rates of increasing the congestion window by using fixed parameters is to frequently switch the increase algorithm, AI or MI, in the millisecond time scale, which is the key concept of HI. Fig. 1 depicts the format of a frame of HI. As shown in Fig. 1, the window increase rate in HI can be changed between the rate of AI and MI by adjusting the mixing ratio between MI and AI. Since the adjustment of the mixing ratio is done at the router side, no modification is required for VCP sources which just invoke AI or MI accordingly based on the ECN bits in the ACK packet. Here, we consider deriving congestion window control in HI. We first define the mixing ratio, r $(0 \le r \le 1)$, as follows:

$$r = \frac{n_{01}}{n_{01} + n_{10}}, \qquad (7)$$

where n_{01} and n_{10} are the numbers of arrival packets marked with 01 and 10 during one RTT, respectively. The congestion window control equation in HI can be expressed as follows:

$$\begin{aligned} cwnd(t+rtt) &= r \times (1+\varepsilon_s) \times cwnd(t) \\ &\quad + (1-r) \times \{cwnd(t) + \alpha_s\} \\ &= (1+\varepsilon_s \times r) \times cwnd(t) \\ &\quad + (1-r) \times \alpha_s. \end{aligned} \quad (8)$$

Since the second term on the right side in Eq. (8) is negligible for a large congestion window size, the equation can be approximated as follows:

$$cwnd(t+rtt) \approx (1+\varepsilon_s \times r) \times cwnd(t). \quad (9)$$

In the above equations, $(1+\varepsilon_s \times r)$ corresponds to the HI parameter. Note that since the value of the MI parameter, ε_s, determines the maximum rate of increase in HI, it needs to be set to a larger value than that of the original VCP in order to achieve fast start-up.

2) Traffic Controller: The role of the traffic controller at each router is to control aggregate traffic traversing it so that the network capacity is promptly fully utilized without causing network congestion, by adjusting the mixing ratio between MI and AI. In the router, the traffic controller resets the target value of the link utilization when the estimated value of link utilization is updated. At the discrete time n, the target link utilization, ρ_{n+1}, is determined based on the current link utilization, ρ_n, as follows:

$$\rho_{n+1} = \rho_n + k \times (\rho_\infty - \rho_n), \quad (10)$$

where ρ_∞ indicates the ideal link utilization, i.e., full utilization. k is a control parameter dominating how fast the link utilization reaches the ideal value. Assuming that RTTs of all traversing flows are less than the monitoring time interval, t_ρ, at the router, to boost up the link utilization from ρ_n to ρ_{n+1} during the time interval, the congestion window sizes of all flows need to be increased by ρ_{n+1}/ρ_n times. In other words, the desired value of the HI parameter is equal to ρ_{n+1}/ρ_n. From this reasoning and Eq. (10), the proper mixing ratio at time n, r_n, is derived as follows:

$$r_n = \frac{k}{\varepsilon_s \cdot \rho_n} \times (\rho_\infty - \rho_n). \quad (11)$$

According to the obtained mixing ratio, the router performs probabilistic packet marking. By doing so, the router can control the growth of network traffic and achieve fast convergence of the link utilization to the ideal value.

Here, we analyze the convergence of link utilization. By solving Eq. (10), ρ_n can be expressed by using k as below;

$$\rho_n = \rho_\infty \times \{1 - (1-k)^{n-1}\}. \quad (12)$$

Eq. (12) implies that the time variation of link utilization does not depend on bandwidth capacity. While most end-to-end congestion control schemes spend a long time proportional to the bandwidth capacity in order to fill the pipe, BISE is able to fill link capacity within a certain time regardless of the amount of network capacity. This advantage is significant especially in high BDP networks. Meanwhile, k is a key parameter which determines the convergence speed of link utilization. It is clear from Eq. (12) that the link utilization never exceeds and monotonically increases to the ideal value if $0 < k < 1.0$. Fig. 2 shows the effect of the parameter k on the traffic control. Note that a large value of k contributes to fast convergence of the link utilization.

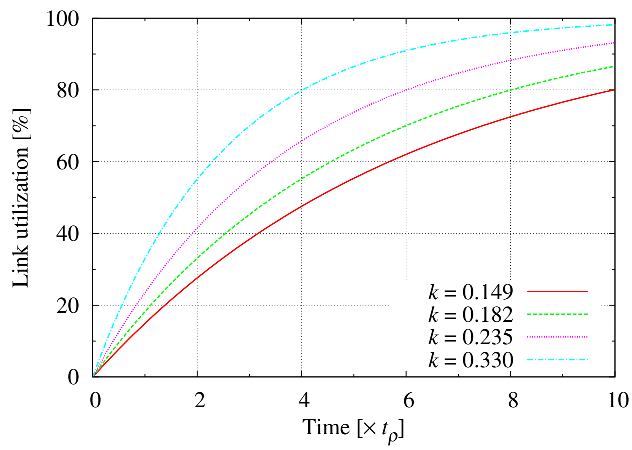


Fig. 2. Analytical results of increase of link utilization in BISE

B. Wireless Loss Tolerant Extension (WLTE)

Since VCP is originally designed for wired high BDP networks, the transmission rate can be drastically decreased in wireless networks. The unnecessary cutdown in the transmission rate is attributed to the fact that VCP adopts the same fast recovery algorithm as TCP; congestion window is halved upon detecting a packet loss regardless of the cause of the loss. However, there is no need to reduce congestion window if the loss is due to wireless errors. On the other hand, almost all losses in VCP are due to wireless errors. Indeed, congestive losses seldom occur in VCP which attempts to maintain empty queue length. Thus, the VCP throughput can be improved by modifying the fast recovery algorithm, i.e., by making no change in the congestion window size. Meanwhile, the congestion window is reset to one at the invocation of the timeout algorithm similar to that in TCP.

V. Performance Evaluation

In this section, we evaluate the performance of the proposed extensions of VCP by using the Network Simulator version 2 (NS2) [13]. A simple dumbbell topology as shown in Fig. 3 is used. The bandwidth of the bottleneck link, B, is set to half of that of other links. The bottleneck link is configured as a lossy link only when simulating wireless environments. The monitoring time interval is set to 1.25 s because of the long delay environment. Each queue has a capacity equal to BDP. The packet size is set to 1040 bytes. As listed in Table III, the values of AI and MD parameters in the proposed scheme are similar to those in the original VCP. However, the value of

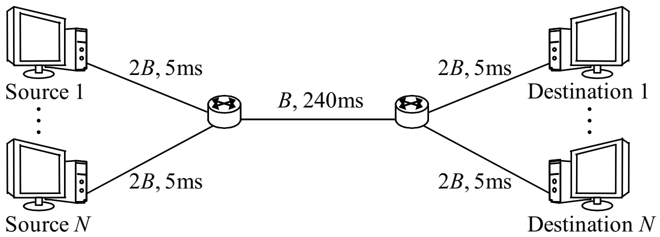


Fig. 3. Network topology

TABLE III
PARAMETER SETTINGS IN THE PROPOSED SCHEME

Time interval t_ρ	1.25 s
MI parameter $(1+\varepsilon)$	10.0
AI parameter α	1.0
MD parameter β	0.875
Queue size	BDP
Packet size	1040 bytes

the MI parameter is changed to 10.0 so that the HI parameter may obtain any value in the range of 1.0 to 10.0. The system parameter, k, is set to 0.330 unless otherwise specified. The original VCP and TCP NewReno are used for comparisons. Simulations were repeated one hundred times and the average values were used as results.

To clarify the performance improvement by BISE, we first evaluate the change of the congestion window size in the start-up period. Fig. 4 shows an example of the congestion window change in case that only a single flow traverses the bottleneck link having 100 Mbps corresponding to BDP=6009 packets. The proposed scheme succeeds in dramatically improving the growth of the congestion window as compared to the original VCP. While TCP experiences timeout caused by heavy packet losses due to its greedy algorithm, "slow start", the proposed scheme can achieve smooth migration from HI to AI without any packet loss. It is worth noting that such performance advantages of the proposed scheme are owing to the proposed BISE extension.

The aim of the second experiment is to both confirm the bandwidth-independent feature of the proposed scheme and clarify the impact of k on the performance. Fig. 5 represents the necessary time to fill 80% of the link capacity for different values of k. The bottleneck link bandwidth, B, is varied from 10 Mbps to 1 Gbps. The number of the flow is set to one. Fig. 5 represents that a larger value of k contributes to the faster increase of throughput as expected from the analytical results depicted in Fig. 2. However, there are certain differences between the results of analysis and simulation. In particular, the values obtained from the simulation exceed those derived from analysis. In addition, for each value of k, the value increases a little bit as the link capacity becomes larger. These gaps occur because of requiring one RTT to establish a connection, and the restriction of the maximum growth rate of the congestion window as the value of the MI parameter is not considered in Eq. (10). To make the proposed scheme strictly bandwidth-independent, the MI parameter needs to be

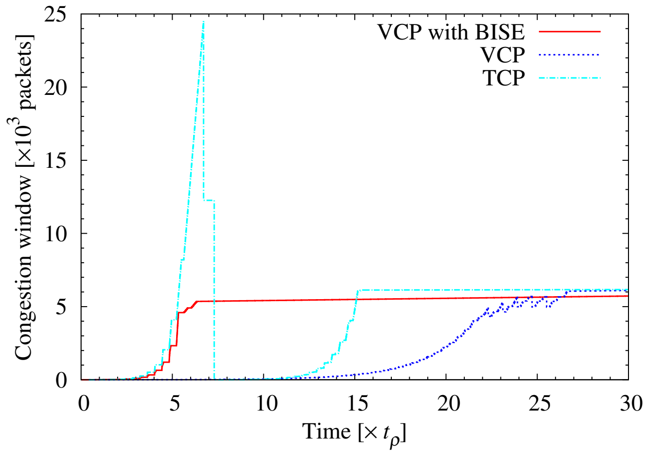


Fig. 4. Comparison of window change in start-up for different protocols

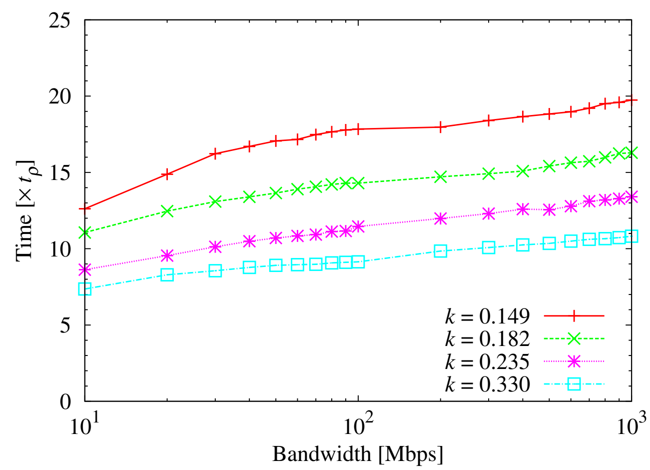
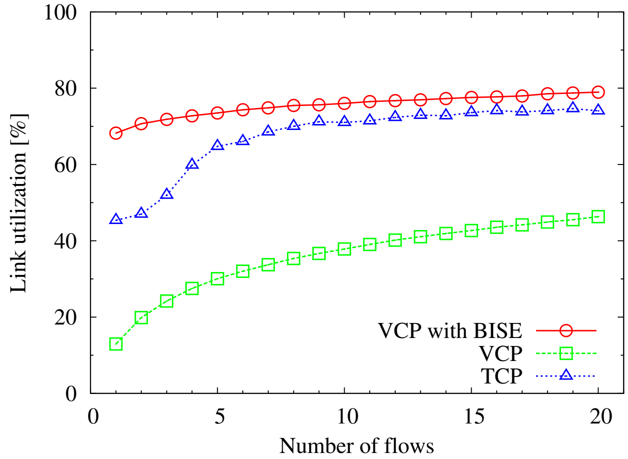


Fig. 5. Elapsed time until 80% of link utilization at the bottleneck link

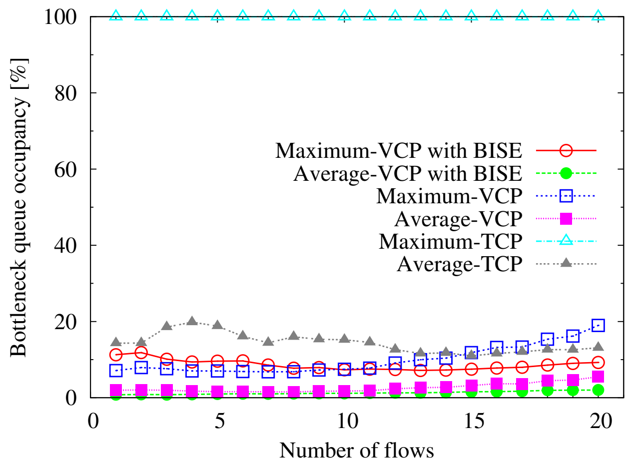
set to a large value while it may lead to undesirable packet losses due to network congestion.

To evaluate the performance under network congestion, the link utilization and the queue occupancy are measured for thirty seconds at the beginning of each connection in different number of flows, varied from one to twenty. The average queue occupancy indicates the value averaged over only when the queue is not empty, and the maximum queue occupancy denotes the maximum value observed during the monitoring period. B is set to 100 Mbps. Fig. 6(a) shows that the proposed scheme achieves high link utilization regardless of the network congestion level. At the same time, Fig. 6(b) demonstrates that the proposed scheme maintains a small queue occupancy even in congested situations. Thus, the proposed scheme inherits the advantages of the original VCP without deteriorating them.

Finally, we evaluate the throughput improvement by WLTE in lossy wireless environments. Fig. 7 demonstrates the goodput of each scheme under different Packet Error Rates (PERs) from 10^{-7} to 10^{-3}. Here, the goodput is calculated from the amount of data, except retransmitted packets, successfully



(a) Link utilization



(b) Queue occupancy

Fig. 6. Comparison of the performance for different number of flows

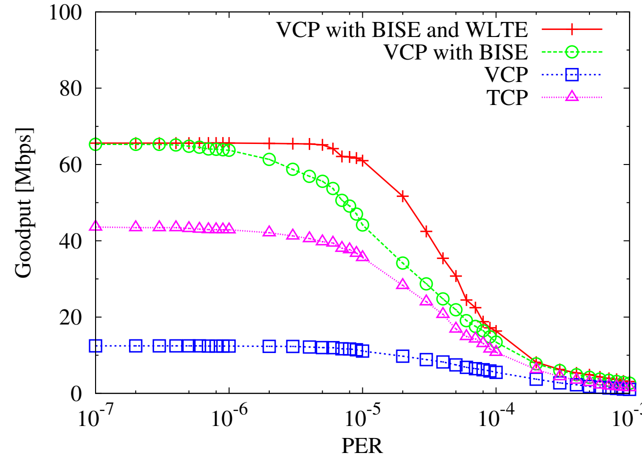


Fig. 7. Goodput for different value of PER

transmitted for thirty seconds at the beginning of each connection. To remove the influence of network congestion, the number of flows is set to one. B is set to 100 Mbps. Both of the proposed schemes are able to accomplish high throughput even in high PER environments. Moreover, it is clearly evident from the comparison between the proposed methods with and without WLTE that WLTE may greatly enhance the throughput especially when PER is 10^{-5}. With too high PERs, throughput is decreased to almost zero because timeouts are consistently invoked by overwhelming packet losses.

VI. Conclusion

In this paper, we have proposed two kinds of extensions for VCP, designed for high BDP networks, to enhance its performance in start-up phase and also in wireless environments. BISE adopting the HI algorithm enables the fast increase in the transmission rate to the 80% of the link bandwidth within a certain period of time regardless of the network link capacity. On the other hand, WLTE enables high throughput even in wireless environments with high error rates. Our simulation results have demonstrated that the proposed extensions significantly enhance the performance of VCP in high BDP and/or wireless environments. They have also revealed that the proposed approaches lead to rapid increase of transmission rate in start-up and also achieve high throughput in wireless environments.

References

[1] Y. Xia, L. Subramanian, I. Stoica, and S. Kalyanaraman, "One more bit is enough," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 37–48, Oct. 2005.
[2] S. Floyd, "Highspeed TCP for large congestion windows," RFC 3649, 2003.
[3] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control (BIC) for fast long-distance networks," in *Proc. of IEEE INFOCOM*, Mar. 2004, pp. 2514–2524.
[4] C. Caini and R. Firrincieli, "TCP hybla: a TCP enhancement for heterogeneous networks," *International Journal of Satellite Communications and Networking*, vol. 22, no. 5, pp. 547–566, Sep. 2004.
[5] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, "TCP westwood: Bandwidth estimation for enhanced transport over wireless links," in *Proc. of ACM MOBICOM*, Italy, Jul. 2001, pp. 287–297.
[6] K. Xu, Y. Tian, and N. Ansari, "TCP-jersey for wireless IP communications," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 4, pp. 747–756, May 2004.
[7] K. Xu, Y. Tian, and N. Ansari, "Improving TCP performance in integrated wireless communications networks," *Computer Networks*, vol. 47, no. 2, pp. 219–237, Feb. 2005.
[8] D. Katabi, M. Handley, and C. Rohrs, "Congestion control high bandwidth-delay product networks," *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 4, pp. 89–102, Oct. 2002.
[9] K. Zhou, K. L. Yeung, and V. O. K. Li, "P-XCP: a transport layer protocol for satellite IP networks," in *Proc. of IEEE GLOBECOM*, Nov./Dec. 2004, pp. 2707–2711.
[10] K. K. Ramakrishnan and S. Floyd, "A proposal to add explicit congestion notification (ECN) to IP," RFC 2481, 1999.
[11] L. Guang, W. Yong-chao, and Z. Miao-liang, Eds., *EVCP: a convergence time improved high-speed transport congestion control protocol*. China: Zhejiang University Press, co-published with Springer-Verlag, 2007.
[12] H. Zhang, G. Dai, L. Yao, and H. Zhou, Eds., *Fast Convergence of Variable-Structure Congestion Control Protocol with Explicit Precise Feedback*. China: Springer-Verlag, 2008.
[13] Network simulator–ns (version 2). [Online]. Available: http://www.isi.edu/nsnam/ns/