Fast Convergence of Variable-Structure Congestion Control Protocol with Explicit Precise Feedback

Huixiang Zhang, Guanzhong Dai, Lei Yao, and Hairui Zhou

College of Automatic, Northwestern Polytechnical University, Xi'an, China
`hxiang.zhang@gmail.com`

Abstract. Traditional TCP has significant limitations such as unclear congestion implication, low utilization in high bandwidth delay product networks, unstable throughput and limited fairness. In order to overcome such limitations, research on design and development of more effective congestion control algorithms, especially in the high bandwidth delay product networks, is very active. Variable-structure congestion Control Protocol (VCP) uses two ECN bits to deliver the bottleneck link utilization region to end systems, and achieves high utilization, low persistent queue length, negligible packet loss rate and reasonable fairness. Owing to the utilization of large multiplicative decrease factor, VCP flows need very long time to finish fairness convergence. To address this problem, a new method called VCP-Fast Convergence (VCP-FC) is proposed in this paper. VCP-FC uses more bits to deliver precise network load factor back to end systems. The end system calculates the fairness bandwidth based on the variance rates of the load factor and throughput, and then quickly adjusts the congestion window to approach the fairness bandwidth. VCP-FC shortens the fairness convergence time effectively, and meanwhile improves the efficiency and fairness of VCP. At last, the performance of VCP-FC is evaluated using ns2 simulations.

1 Introduction

It is well-known that the Addictive Increase Multiplicative Decrease (AIMD)[1] congestion control algorithm employed by traditional TCP[2] doesn't perform well in high bandwidth delay networks. The research on design and development of more effective congestion control algorithms, especially in the high bandwidth delay product networks, is very active. One direction is pure end-to-end improvement, such as High-speed TCP[3], LTCP[4], Fast TCP[5] and Scalable TCP[6]. These algorithms increase congestion window aggressively and decrease conservatively to improve the network utilization. Another approach is to utilize explicit feedback from internet routers, such as VCP[7],RCP[8],ACP[9],XCP[10]. These algorithms redesign the internet to achieve high utilization, low persistent queue length, negligible packet loss rate and max-min fairness.

Among the algorithms utilizing explicit feedback, Variable-structure congestion Control Protocol (VCP) is a simple and low complexity protocol that modifies mainly in end systems and deploys easier than XCP. VCP is able to achieve

F.P. Preparata, X. Wu, and J. Yin (Eds.): FAW 2008, LNCS 5059, pp. 264–275, 2008.

comparable performance to XCP but converges significantly slower to the fair bandwidth than XCP. When a new flow joins the network of high utilization, the existing flows can't decrease occupied bandwidth fast owing to VCP large MD factor β (0.875), the new flow converges to the fair allocation very slowly.

There are several relative researches on improving convergence speed to fairness in the literature. [11] improves the convergence speed of High-Speed TCP to the fair bandwidth. The proposed mechanism checks the congestion window size just before a loss event. If the size continuously declines, the window is assessed to be on a downward trend. Once the difference between the maximum and minimum values of the checked size during the current downward trend exceeds a threshold, the congestion window decrease parameter is set larger than usual. Thus, flows with a larger window size than fair can decrease their window more aggressively to improve the convergence times. But the condition described above is an very special case in the fast changing network environment, so the mechanism isn't an general solution. [12] presented a congestion control algorithm based on the traditional TCP. The algorithm explicitly calculates the fair share and converges to it in two congestion cycles in a distributed fashion. But owing to the limitation of TCP, only synchronous flows is analyzed in [12].

To improve VCP convergence speed to fairness, a new method called VCP-Fast Convergence (VCP-FC) is proposed in this paper. VCP-FC use more bits to deliver precise network load factor back to end systems. End systems estimate the fair bandwidth based on the variance rates of the load factor and throughput. If current bandwidth deviates from the fair bandwidth, the flow rapidly adjusts its congestion window to the fair bandwidth, which improves the convergence speed to fairness. If current bandwidth approaches the fair bandwidth, the flow apply lager MD factor (0.9) than VCP so as to smooth the flow's throughput and meanwhile improves the network utilization. The additional benefit of using precise feedback is to improve VCP convergence speed to efficiency in MI stage.

The rest of the paper is organized as follows: Section 2 briefly reviews VCP and analysis its convergence behaviors. Section 3 elaborates VCP-FC. Section 4 uses ns2 simulations to evaluate the performance of VCP-FC. Finally, conclusions and future works are provided in Section 5.

2 Variable-Structure Congestion Control Protocol (VCP)

The VCP router calculates the load factor ρ periodically:

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

Here t_ρ is the calculation interval. Owing to 75% $\sim$ 90% of flows have RTTs less than 200 ms [13], VCP set t_ρ=200 ms. λ is the amount of input traffic during the last interval t_ρ. q is the persistent queue length during the last interval t_ρ. κ controls how fast the persistent queue drains and set $\kappa = 0.5$. γ is the target utilization and set $\gamma = 0.98$. C is the link bandwidth.

In every interval t_ρ, the link utilization is classified into three regions based on the load factor ρ. If $0 \leq \rho < 80\%$, the link utilization is classified as low-load region; if $80 \leq \rho < 100\%$, the link utilization is classified as high-load region, if $\rho \geq 100\%$, the link utilization is classified as overload region. VCP routers encode the utilization regions into two ECN bits in the IP header of each data packet. This information is then sent back by the receiver to the sender via ACK packets. Depending on the utilization regions, the sender applies different congestion response. If in low-load region, the sender increases its sending rate using MI to improve the link utilization quickly; if in high-load region, the sender increases its sending rate using AI to improve the link utilization slowly; if in overload region, the sender decreases its sending rate using MD immediately. The respective response functions are as follows:

$$MI : cwnd(t + rtt) = cwnd(t) \times (1 + \varepsilon) \tag{2}$$

$$AI : cwnd(t + rtt) = cwnd(t) + \alpha \tag{3}$$

$$MD : cwnd(t + rtt) = cwnd(t) \times \beta \tag{4}$$

Where $\varepsilon = 0.0625$, $\alpha = 1$, $\beta = 0.875$. To offset the impact of the RTT heterogeneity, VCP scales ε and α using equation (5)(6) respectively according to their RTTs. And further, in order to allocate the bandwidth in fairness, VCP uses equation (7) adding an additional scaling factor to the AI algorithm:

$$\varepsilon_s = (1 + \varepsilon)^{\frac{rtt}{t_\rho}} - 1 \tag{5}$$

$$\alpha_s = \alpha \frac{rtt}{t_\rho} \tag{6}$$

$$\alpha_{rate} = \alpha_s \frac{rtt}{t_\rho} = \alpha (\frac{rtt}{t_\rho})^2 \tag{7}$$

The behavior of VCP convergence could be divided into two stages. Stage one is convergence to efficiency. VCP flows quickly take the available bandwidth using MI. The link utilization ramps up to 80% quickly, which shows VCP has high efficiency. Stage two is convergence to fairness. VCP flows achieve to the fair bandwidth using AIMD. VCP doesn't guarantee fairness in stage one. Owing to the scaling of α_s, the same time started flows but of different RTTs will converge to the same congestion window in stage one, which also means that flows converge to unfair bandwidth. Only in stage two these flows of different sending rate converge to fairness. Owing to the scaling of α_s, flows of different RTTs increase their congestion window equally every t_ρ interval. Owing to the scaling of α_{rate}, flows of different RTTs increase their bandwidth equally every t_ρ interval. To prevent the system from oscillating between MI and MD, VCP set the MD factor $\beta = 0.875$. As only the MD function can affect the convergence time to fairness, the choice of 0.875 is the root cause of slow convergence to fairness.

The behavior that VCP flows converge to fairness slowly manifest in two aspects. Firstly, the same time started flows but of different RTTs need long time to converge to the fairness bandwidth. Figure 1 shows the convergence of congestion window of two VCP flows with different RTTs (20 ms and 100 ms respectively). Two flows start to send packets at $t = 0$. The bottleneck bandwidth is 100Mbps. In stage one, i.e. convergence to efficiency, two flows converge to the identical congestion window value. Then the flows converge to the fairness bandwidth using AIMD in stage two. The total convergence time is about 300 seconds. Secondly, when the link utilization is in high-load region, i.e. the load factor is between 80% and 100%, one new flow joins and starts sending packets; the new flow needs long time to converge to the fairness bandwidth. As shown in Fig.2, one flow starts firstly and achieves the stable state, then the other new flow starts to send packets at $t = 100s$. The two flows have identical RTT of 80 ms. The bottleneck bandwidth is 100Mbps. The new flow needs about 300 seconds to converge to the fairness bandwidth.

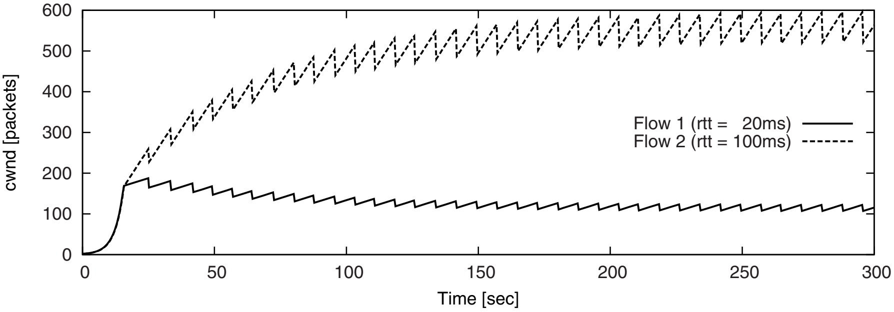


Fig. 1. Two VCP flows of different RTTs start simultaneously

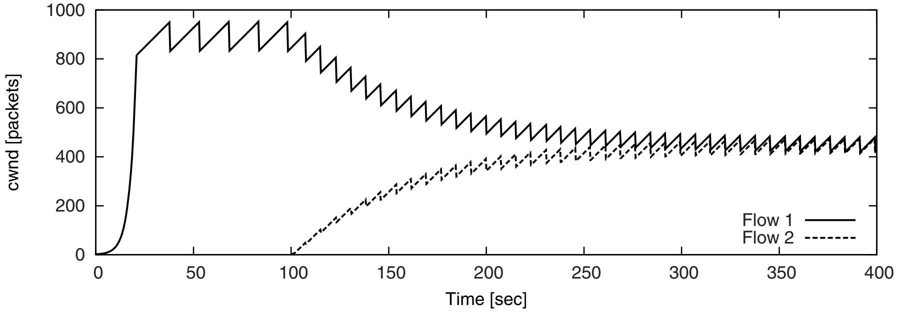


Fig. 2. Two VCP flows of identical RTTs start sequentially

3 VCP Fast Convergence (VCP-FC)

VCP-FC keeps the algorithms of VCP router unchanged and just uses ten bits to deliver the load factor back to end systems. The load factor ρ is represented using ten bits, that is to say the quantized load factor has the precision of 0.001. With the quantized load factor, end systems can improve the convergence speed to efficiency and fairness. The following subsections will describe them separately.

3.1 Convergence to Efficiency

In the stage of convergence to efficiency, $0 \leq \rho < 80\%$. Using the precise load factor, VCP-FC can adjust the congestion window more quickly and accurately. VCP-FC substitutes the VCP MI response function with equation (8):

$$cwnd(t + rtt) = cwnd(t) + \mu(1 - \rho(t))cwnd(t) \tag{8}$$

Where $\rho(t)$ is the load factor at time of t. μ controls how fast the convergence to efficiency; in order to guarantee the robustness, we choose $\mu = 0.5$. VCP-FC scales $\rho(t)$ as VCP does in equation (5) and just replaces ε with $\rho(t)$. At the end of convergence to efficiency, simultaneously started flows can reach the same value of congestion window. Using the new MI response function, the convergence speed to efficiency is improved effectively.

Suppose there is a single bottleneck with bandwidth of C shared by multiple flows. The flows start to send packets simultaneously. Assuming the flows have identical RTTs and start from the unit aggregate rates r(0)=1. The flows converge to efficiency using MI. For VCP flows, [7] proofs that the aggregate rate after n rounds of MI is $r(n) = r(0)(1+\varepsilon)^n$, where $\varepsilon = 0.0625$; then at the end of convergence to efficiency, VCP flows need $\frac{\log(0.8C)}{\log(1+\varepsilon)}$ rounds of MI. For VCP-FC flows, the aggregate rate after n rounds of MI is $r(n) = r(0)\prod_{i=1}^{n}(1 + \mu(1 - \rho_i))$, where $0 \leq \rho_i < 0.8$ and $\mu = 0.5$, so $\mu(1 - \rho_i) > 0.1 > \varepsilon$. Apparently the number of MI rounds VCP-FC flows needed is less than $\frac{\log(0.8C)}{\log(1+0.1)}$. Compared with VCP flows, VCP-FC flows converge to efficiency faster than VCP flows.

3.2 Convergence to Fairness

In the stage of convergence to fairness, $\rho \geq 80\%$. VCP flows converge to fairness using AIMD. Additive increase doesn't affect the fairness among flows. The convergence speed to fairness is determined by the MD factor, i.e. β. The smaller the MD factor, the faster approaching the fairness; but the oscillation is higher too. To prevent the system oscillation between MI and MD, VCP set $\beta = 0.875$. To decrease the MD factor to improve the convergence speed to fairness isn't a good solution.

VCP-FC delivers the load factor back to end systems. Thus, the fairness bandwidth is able to estimate using the variance rates of the load factor and throughput in end systems. Suppose there is a single bottleneck with bandwidth of C shared by multiple flows. Consider an situation which there are N flows existing and no flows join or leave during a period of ΔT. In the stage of convergence to fairness, all flows update their congestion window using AIMD. When the bottleneck link is in high-load region, the aggregate incremental amount of bandwidth during ΔT is : $\sum_{i}^{n}(r_i(t+\Delta T) - r_i(t))$. Owing to additive increase, all flows increase their bandwidth by the same amount, denote as Δr, so we have:

$$\sum_{i}^{n}(r_i(t+\varDelta T)-r_i(t))=N\varDelta r \tag{9}$$

Assuming $\varDelta T > t_\rho$, the incremental amount of the bottleneck utilization is $\varDelta u = u(t+\varDelta T) - u(t)$, we have:

$$\varDelta u = \frac{\sum_i^n (r_i(t+\varDelta T)-r_i(t))}{C} = \frac{N\varDelta r}{C} \tag{10}$$

Thus, We can calculate the fairness bandwidth F as follow:

$$F = \frac{C}{N} = \frac{\varDelta r}{\varDelta u} \tag{11}$$

And further we can obtain the fairness congestion window W as follow:

$$W = F \bullet rtt = \frac{\varDelta w}{\varDelta u} \tag{12}$$

The VCP is able to achieve very low persistent queue length, thus the change of utilization $\varDelta u$ is approximately substituted with the change of load factor $\varDelta\rho$ in end systems.

Suppose the network reaches congestion at some point, due to addictive increase. Then all flows will decrease their bandwidth multiplicative, and then resume addictive increase until the network congests again. We choose the interval between sequent congestion points as $\varDelta T$ to estimate the fairness bandwidth more accurately. Figure 3 elaborates this interval. There are $m > 1$ rounds of AI and one round of MD in every $\varDelta T$ interval. The bottleneck utilization is changing every t_ρ interval. Thus, in each $\varDelta T$ interval we can obtain n pairs of value denote as (u_i, w_i), where u_i represent the load factor every t_ρ interval; w_i represent the value of congestion window when u_i is feed back to the end system. At the time of $T + \varDelta T$, end systems calculate the fairness bandwidth using equation (13). End systems take (u_n, w_n) as the base value and calculate n-1 values of the fairness congestion window. Then end systems calculate the average of these values, so we obtain the fairness congestion window W_f in this $\varDelta T$ interval:

$$W_f = \frac{1}{n-1}\sum_{i=1}^{n-1}\frac{w_n - w_i}{u_n - u_i} \tag{13}$$

The end systems estimate one new W_f every $\varDelta T$ interval, then smooth the value as follow:

$$W_e^f = (1-\theta) last\_W_e^f + \theta W_f \tag{14}$$

In order to track the quickly changed network environment we choose larger value of θ and set $\theta = 0.4$.

End systems compare current congestion window with W_e^f, if $|cwnd - W_e^f| < \eta W_e^f$, where $\eta = 0.1$, that means the two variable is approximately equally, flows approach the fairness approximately. Then we can use larger MD factor to smooth the flow throughput and set $\beta = 0.9$; otherwise we use following equation to update the congestion window:

$$cwnd_{new} = (1 - \omega)cwnd + \omega W_e^f \quad (15)$$

where $\omega = 0.2$. In some situation, we can't obtain enough samples, if $n < 5$, VCP-FC set $\beta = 0.875$ as original VCP.

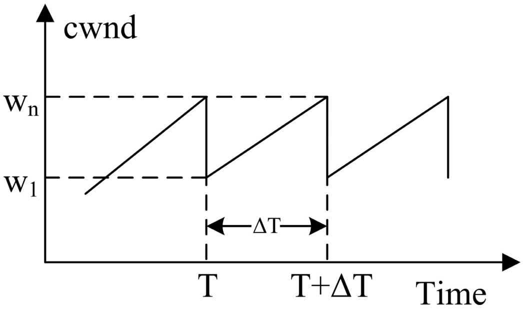


Fig. 3. Interval of ΔT

4 Simulations and Results

We incorporated our algorithm into VCP and validated its performance on NS-2[14]. The performance of VCP-FC is compare with VCP and XCP. We use the single congested link topology shown in Fig.4, where S_i is sending packets to D_i. We evaluate the convergence time to efficiency firstly, and then evaluate the convergence time to fairness when varied fairness bandwidth and round-trip times separately. At last, we study the performance of VCP-FC in an RTT heterogeneity environment and in an dynamic environment. We use FTP as the application layer data generator in all simulations. The data packet size is set to 1KBytes. The parameters of VCP and XCP is set according to the authors' recommendations in [7] and [10] separately.

Simulation results for convergence to efficiency: The bottleneck bandwidth varies from 2Mbps to 1Gbps. Only a single flow starts to send packets at $t = 0$, and its RTT=80 ms. The convergence time to efficiency is defined as how much time needed the bottleneck utilization reaches 80%. The result is show in Fig.5, and the x-axis is in logarithmic scale. From Fig.5, XCP converges to efficiency fastest. The convergence time to efficiency of VCP-FC is shorter than VCP regardless of the bandwidth. When the bandwidth varied from 2Mbps to 1Gbps, the convergence time to efficiency of VCP-FC increases slower than VCP. The result shows the VCP-FC flows converge faster to efficiency than VCP.

Simulation results for converge to fairness with varied fairness bandwidth: One flow starts to send packets at $t = 0$ and reaches the stable situation. Then the other flow starts to send packets. Two flows have identical RTT

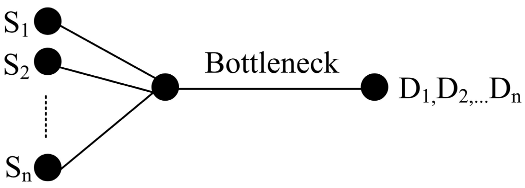


Fig. 4. A single bottleneck topology

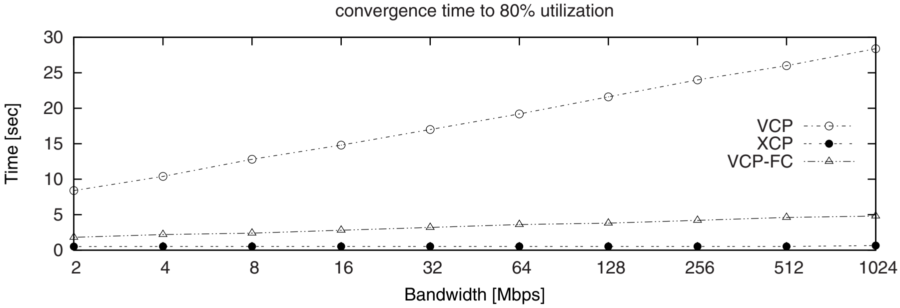


Fig. 5. Convergence time to efficiency versus bandwidth

of 80ms. The bottleneck bandwidth varies from 2Mbps to 1Gbps, which means the fairness bandwidth varies from 1Mbps to 512Mbps. The convergence time to fairness of the second flow is measured using the metric of $\delta - fair$ convergence time proposed in [15]. The metric is defined as the time taken for the second flow converges to $\frac{1-\delta}{2}$ of the link bandwidth. Here we set $\delta = 0.1$. As shown in Fig.6, XCP converges to the fairness bandwidth very fast and hardly affect by the fairness bandwidth. And the convergence time to fairness of VCP-FC is almost the same as VCP when the fairness bandwidth is less than 2Mbps. When the bandwidth increased, the increment of convergence time of VCP-FC is much less than VCP. The result shows VCP-FC significantly improve the convergence speed to fairness in high bandwidth environment, but still consume more time than XCP.

Simulation results for converge to fairness with varied RTT: The bottleneck bandwidth is fixed at 45Mbps. One flow starts to send packets at $t = 0$ and reaches the stable situation. Then the other flow starts to send packets. The RTT of the two flows is varied from 20ms to 200ms. The convergence time to fairness of the second flow is measured using $\delta - fair$ convergence time as the preceding simulation. As shown in Fig.7, the impact of RTT is rather slight to convergence time of VCP-FC, VCP and XCP. As the RTT grows, the convergence time increases very slowly. The convergence time of VCP-FC is smaller than VCP regardless of RTT, and XCP converges fastest to fairness.

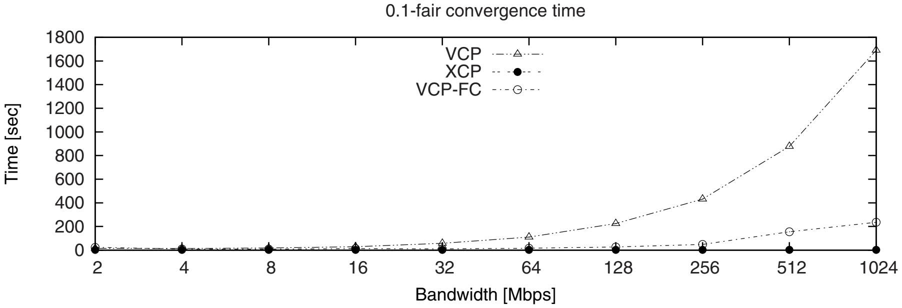


Fig. 6. Convergence time to fairness versus bandwidth

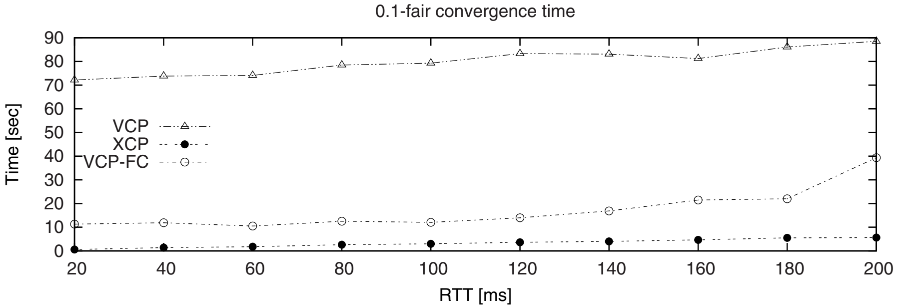


Fig. 7. Convergence time to fairness versus RTT

Simulation results for RTT heterogeneity environment: VCP flows can achieve max-min fairness in some extent. VCP-FC preserves the good property of VCP. We have 20 flows sharing the bottleneck link, and the bottleneck bandwidth is fixed at 200Mbps. We perform four sets of simulations: (a) the same RTT of 20ms; (b) small RTT difference from 20ms to 96ms; (c) large RTT difference from 20ms to 153ms; (d) huge RTT difference from 20ms to 229ms. We measured flows throughput in equilibrium. As shown in Fig.8, VCP-FC is able to allocate bandwidth fairly among competing flows, as long as their RTTs are not significantly different. With the RTT heterogeneity increases, the fairness of VCP-FC is degrade.

Simulation results for dynamic environment: In an dynamic environment, flows usually join or leave the network in an unpredictable manner. When flows leave, the available bandwidth is increased. VCP is able to take the available bandwidth by MI action quickly, and so does VCP-FC. when flows join, the contention of flows become intensive. The existing flows should decrease their bandwidth to make room for new flows. Here we focus on the effect of increased contention. We have 10 flows sharing a 400Mbps bottleneck. At t=60s, 110s, 160s, there are 10 flows join the network respectively. All flows have identical RTT of 80ms. The simulation last for 210s. Thus, the duration is divided into

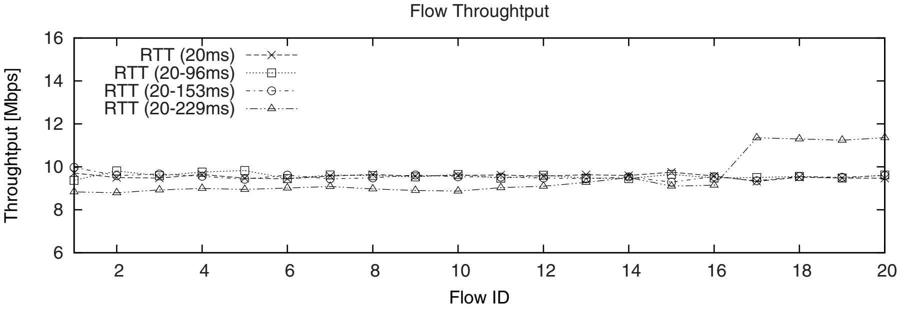
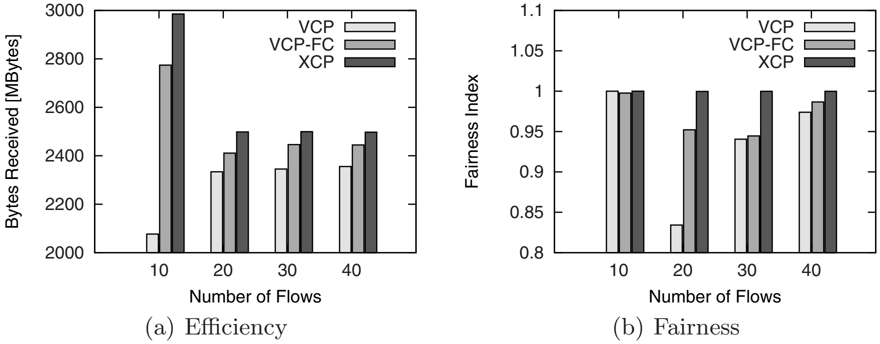


Fig. 8. Flow throughput in an RTT heterogeneity environment

four parts, which is 0-60s, 60-110s, 110-160s, 160-210s. We measured the efficiency and fairness in each part. The efficiency is measured using the goodput which is the bytes received in receivers. The fairness is measured using Fairness Index presented in [16]: $F(x) = (\sum x_i)^2/n(\sum {x_i}^2)$, where x_i is the goodput achieved by each flow. As show in Fig.9, XCP outperforms VCP and VCP-FC both in efficiency and fairness when graduated contention increased. In the first part of the duration, the efficiency of VCP-FC is much better than VCP, which also means VCP-FC converge to efficiency faster than VCP. In all parts of the duration, VCP-FC outperforms VCP in efficiency and fairness.



(a) Efficiency (b) Fairness

Fig. 9. Performance for graduated contention increase

5 Conclusion

This paper has proposed a new mechanism called VCP-FC for improving the convergence speed of VCP to efficiency and fairness. VCP-FC uses ten bits to deliver the load factor back to end systems. With the support from routers, VCP-FC can estimate the fairness bandwidth every congestion cycle. Simulation results show that VCP-FC is valid in stationary and dynamic environment, and also effective in the RTT heterogeneity environment.

VCP-FC improves the efficiency and fairness of VCP effectively, but still not as good as XCP. The weak point of VCP-FC manifest mainly in two aspects. Firstly, the convergence time to fairness is affected by the fairness bandwidth, which is determined by the AIMD algorithms employed by VCP-FC. Secondly, the router calculates the load factor every t_ρ interval ($t_\rho = 200ms$). If the RTT of flows is beyond 200ms, the fairness of VCP and VCP-FC becomes worse. In such large RTT variance environment, the validity of VCP-FC needs further research.

References

1. Chiu, D., Jain, R.: Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. Computer Networks and ISDN Systems 17(1), 1–14 (1989)
2. Jacobson, V.: Congestion Avoidance and Control. ACM SIGCOMM Computer Communication Review 25(1), 157–187 (1995)
3. Floyd, S.: RFC3649: HighSpeed TCP for Large Congestion Windows. Internet RFCs (2003)
4. Bhandarkar, S., Jain, S., Reddy, A.: Improving TCP Performance in High Bandwidth High RTT Links Using Layered Congestion Control. In: The 3rd International Workshop on Protocols for FAST Long-Distance Networks (2005)
5. Wei, D., Jin, C., Low, S., Hegde, S.: FAST TCP: Motivation, Architecture, Algorithms, Performance. IEEE/ACM Trans. Networking 16(6), 1246–1259 (2006)
6. Kelly, T.: Scalable TCP: Improving Performance in Highspeed Wide Area Networks. ACM SIGCOMM Computer Communication Review 32(2), 83–91 (2003)
7. Xia, Y., Subramanian, L., Stoica, I., Kalyanaraman, S.: One More Bit Is Enough. In: The 2005 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 37–48. ACM Press, New York (2005)
8. Dukkipati, N., Kobayashi, M., Rui, Z., McKeown, N.: Processor Sharing Flows in the Internet. In: The 13th International Workshop on Quality of service, pp. 286–297. Springer, Berlin (2005)
9. Lestas, M., Pitsillides, A., Ioannou, P., Hadjipollas, G.: Adaptive congestion protocol: A congestion control protocol with learning capability. Computer Networks 51(13), 3773–3798 (2007)
10. Katabi, D., Handley, M., Rohrs, C.: Congestion Control for High Bandwidth-Delay Product Networks. In: The 2002 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 89–102. ACM Press, New York (2002)
11. Nabeshima, M., Yata, K.: Improving the convergence time of highspeed TCP. In: The 12th IEEE International Conference on Networks, pp. 19–23. IEEE press, New York (2004)
12. Attie, P., Lahanas, A., Tsaoussidis, V.: Beyond AIMD: Explicit Fair-share Calculation. In: The 8th IEEE International Symposium on Computers and Communications, pp. 727–734. IEEE press, Washington (2003)
13. Jiang, H., Dovrolis, C.: Passive Estimation of TCP Round-Trip Times. ACM Computer Communications Review 32(3), 75–88 (2002)

14. The network simulator ns-2.30, `http://www.isi.edu/nsnam/ns`
15. Bansal, D., Balakrishnan, H., Floyd, S., Shenker, S.: Dynamic Behavior of Slowly-Responsive Congestion Control Algorithms. In: The 2001 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 263–274. ACM Press, New York (2001)
16. Jain, R., Chiu, D., Hawe, W.: A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Systems. Technical Report TR-301, Digital Equipment Corporation (1984)