

An Enhanced TCP Vegas Algorithm Based on Route Surveillance and Bandwidth Estimation over GEO Satellite Networks

Jingyao Qu

Beijing University of Aeronautics and Astronautics, Beijing 100191, China
qujingyao666@126.com

Abstract—TCP Vegas provides better performance than TCP Reno with respect to overall network utilization, stability, fairness, throughput and packet-loss. Thus TCP Vegas is usually used in space communication environment as well as satellite networks, which have large bandwidth and long delay. However, TCP Vegas is inefficient in the large GEO satellite networks. The burst during slow-start phase causes Vegas to change from slow-start phase to congestion-avoidance phase too early, and it also has disadvantage that can not distinguish link changing and congestion. In this paper, TCP Vegas-W is presented based on route surveillance and bandwidth estimation, and with surveillance of bandwidth and Round-Trip Time (RTT), congestion level is estimated much more correctly and precisely. Simulation results show that Vegas-W algorithm can get much better performance.

Keywords—TCP Vegas-W, Route Surveillance, Bandwidth Estimation, GEO Satellite Network

I. INTRODUCTION

The Transmission Control Protocol (TCP) is used as transport protocol by most of current Internet applications. The behavior of TCP is tightly coupled with the overall Internet performance. By now, TCP Reno[1] has become the most widely used version, but it usually causes network congestion and unstable bandwidth usage because it estimates the available bandwidth by detecting the packet loss. Studies have demonstrated that Vegas outperforms Reno in the aspects of overall network utilization [2],[3],[4], stability[5],[8], fairness[5],[8], throughput and packet loss [2],[3],[4],[5], and burstiness[5]. However, theory and experiments show that, TCP Vegas performs poorly in GEO Satellite network environment. It performs too sensitive when it detects the bandwidth of the link, which makes it end the slow-start phase too early. This means much longer time is needed to reach the available bandwidth. And in GEO Satellite, the path always changes, which can not be identified by Vegas. When RTT improves, we usually can not tell whether it is caused by link variety or congestion.

In our paper, a novel method called TCP Vegas-W is proposed. Route surveillance and bandwidth estimation is used to optimize TCP Vegas method. Simulation shows that our method performs much better than Vegas when used in GEO satellite network environment.

II. RELATED WORK

Several studies have been made to improve the TCP performance over high bandwidth and long-delay links.

TCP New Vegas[6] is proposed to improve the performance of TCP Vegas over high latency links. While α and β are invariable in Vegas algorithm, Vegas-A makes these two variables adjusted with the network status, and performs better than TCP Vegas in both wired and satellite networks. VenO is a mechanism which combines advantage of Vegas and Reno and differentiates the reason of losing packets, and at last improves the performance of Vegas. Over LEO Satellite network environment, Vegas-AB is proposed. It can change the value of α and β adaptively.

But mechanisms all above have not solved the problem that slow-start phase ends too early especially in high bandwidth and long delay networks. In GEO satellite network environment, if slow-start phase ends too early, it will takes much longer time to reach the bandwidth, which means network bandwidth has not been fully utilized. This will largely decreases transmission efficiency. Again, methods above have not solved the problem that BaseRTT can not be valued correctly which may causes congestion level can not be estimated precisely in satellite network environment. We can not judge the congestion level, and can not use the bandwidth powerfully.

III. TCP VEGAS AND BANDWIDTH ESTIMATION MECHANISM

In this section, we will review the mechanism of Vegas and bandwidth estimation method

A. TCP Vegas

Vegas adopts a more sophisticated bandwidth estimation scheme that tries to avoid rather than to react to congestion. It uses the measured RTT to accurately calculate the amount of data packets that a source can send, as shown in the following:

$$cwnd = \begin{cases} cwnd + 1 & diff < \alpha \\ cwnd & \alpha \leq diff \leq \beta \\ cwnd - 1 & diff > \beta \end{cases} \quad (1)$$

Where

$$diff = Expected_Rate - Actual_Rate$$

$diff$ is the amount of extra packets, which is estimated by TCP Vegas.

$Expected_Rate = \frac{cwnd(t)}{BaseRTT}$, where $cwnd(t)$ is the current congestion window size and $BaseRTT$ is the minimum RTT of that connection so far.

$Actual_Rate = \frac{cwnd(t)}{RTT}$, where RTT is the actual

round-trip time.

α and β are parameters whose values are usually set as 1kb/s and 3kb/s, respectively.

Several studies establish that TCP Vegas does achieve higher efficiency than Reno, causes much fewer packet retransmissions, and is not biased against the connections with longer round trip times (RTTs). But in GEO satellite network, it is more easily to mistakenly judge the congestion level, and then be too conservative to send packets. And in GEO satellite network environment, it is common that the routes change. Then it is not right if we still use the original BaseRTT. Besides, BaseRTT is used to calculate the value of diff, so it has a large bad effect on the transmission efficiency of the whole system if BaseRTT is not accurate.

B. Bandwidth Estimation

The method we use to estimate bandwidth is the same as the one which is used in TCP Westwood[7][8]. If an ACK is received at the source at time t_k , this implies that a corresponding amount of data d_k has been received by the TCP receiver. Therefore, we can measure the following sample of bandwidth used by that connection as:

$$b_k = \frac{d_k}{t_k - t_{k-1}} \quad (2)$$

Where t_{k-1} is the time the previous ACK was received.

Since congestion occurs whenever the low-frequency input traffic rate exceeds the link capacity[9], we employ a low-pass filter to average sampled measurements and to obtain the low frequency components of the available bandwidth. Notice that this averaging is also critical to filter out the noise due to delayed acknowledgments.

We propose the following discrete time filter which is obtained by discretizing a continuous low-pass filter using the Tustin approximation. We obtain:

$$\hat{b}_k = \alpha \hat{b}_{k-1} + \frac{(1-\alpha)}{2} [b_k + b_{k-1}] \quad (3)$$

$$\alpha = \frac{\frac{2\tau}{\Delta_k} - 1}{\frac{2\tau}{\Delta_k} + 1} \quad (4)$$

$$\Delta_k = t_k - t_{k-1} \quad (5)$$

Where \hat{b}_k is the filtered measurement of the available bandwidth at time $t = t_k$, and $\frac{1}{\tau}$ is the cut-off frequency of the filter.

IV. OUR PROPOSED MECHANISM: VEGAS-W

Compared with TCP Vegas, the largest improvement of our mechanism is that route surveillance and bandwidth estimation are used. We use the result of bandwidth estimation to monitor the slow-start process. With surveillance of RTT, we can distinguish link changing and congestion, and improve congestion avoidance mechanism of Vegas..

A. Slow-start Mechanism of Vegas-W

TCP Vegas performs poorly in GEO Satellite network environment. It performs too sensitive when it detects the bandwidth of the link, which makes it end the slow-start phase too early. When TCP connection is in slow-start phase, maybe a lot of packets are sent in a short time, then some packets will be accumulated in the network. This phenomenon will lead to queuing delay increasing temporarily. And with increasing of size of the congestion window, this phenomenon will be more and more obviously. In this case, it is improper if we mistakenly judge the congestion level, and end slow-start phase blindly. Congestion window will spend much longer time to reach a reasonable size.

In Vegas-W, we use estimated bandwidth to keep watch on the network. While TCP connection is established, TCP-W begins to use method above to estimate bandwidth. When the estimated amount of extra data is greater than the slow-start thresh, Vegas-W doesn't leave the slow-start phase at once, but compare the congestion window size with estimated bandwidth. If the congestion window is smaller than bandwidth, it means the congestion phenomenon is temporary, the connection should not end slow-start phase. On the contrary, if the congestion window is larger than bandwidth it means TCP connection needs to enter congestion avoidance phase.

B. Improvement of Vegas-W in Congestion Avoidance Phase

BaseRTT is used to calculate the expected packets-sent velocity in Vegas. So if BaseRTT is not accurate, we can not estimate the exact congestion level, and can not use the bandwidth fully. In wired network, link status will not change too much, so BaseRTT will not change too much adaptively. But in GEO satellite network, it is common that reroute happens or links change a lot. In this case, it is not appropriate to use a fixed BaseRTT. We can know from the congestion-avoidance mechanism of Vegas, BaseRTT takes part in the computation of diff, and influences the adjustment of the congestion window size. If we estimate the value of BaseRTT with a big error, the performance of the system will be largely influenced.

The optimized method in Vegas-W is monitoring RTT of each packet in sending end. When TCP connection enters congestion-avoidance phase, bandwidth size will not be estimated, but the changing of RTT value will be monitored. When network congestion takes place, RTT will change gradually. But if link status changes or reroute happens,

RTT will change sharply. This feature can be used to distinguish the two conditions.

When TCP connection enters congestion avoidance phase, RTT is estimated when an ACK is received. We record it as RTT_{now} , and compare this value with the average $RTT(RTT_{ave})$ of the m (m will be decided by link environment, in our simulation environment, we set it as 20) packets whose sequence numbers are just before the sequence number of this packet. If RTT_{now} is greater than $nRTT_{ave}$ (n will also be decided by link environment, in our simulation environment, we set it as 2), BaseRTT refreshes.

C. Improvement of Vegas-W in Retransmission and Recovery Phase

The major difference between Vegas-W and Vegas in retransmission and recovery mechanism are whether the method differentiates the congestion reason. In Vegas-W, when packets sending end receives a duplicated ACK, it will not reduce the congestion window at once, but judge the reason why the data packet is lost (congestion or link error). When sending end receives a duplicated ACK, it will look back to the last two packets whose sequence numbers are just before the abnormal packet. We must make sure two things: First, the sequence of these two packets must be next to the abnormal packet's sequence number; second, these two packets are translated normally. Then we record the sending time and receiving time of the two packets. With these time values and our computation, we can get the difference of the sending time of the two packets, and get the difference of ACK arrival time of the two packets. We compare these two values. If the former is less than the latter, it means the network is close to congestion. We consider that packets lost are caused by congestion. Other while, we consider that packets lost are caused by link error. If congestion leads to packets lost, we set $ssthresh$ (slow-start thresh) equal to the product of the estimation value of bandwidth and the minimum of $RTT(RTT_{min})$, and set the congestion window equal to $ssthresh$; If packets lost are caused by link error, the value of $ssthresh$ and congestion window will not be changed.

If packets lost are caused by time-out, the system enters slow-start phase, the mechanism of slow-start phase is then started.

V. SIMULATION OVER GEO NETWORK

The network simulator (ns-2) is used as the tool to conduct our simulation. Figure.1 shows the network simulation topology, where S_i , R_i and G_i represent sender hosts, destinations and gateways respectively. $G1-G2$ is bottleneck link. The links are labeled with their bandwidth and propagation delay. $S-G$, $G-R$ represents satellite-land networks, and $G1-G2$ represents a satellite-satellite network. We assume that all window sizes are measured in number of fixed-size packets, which are 1000 bytes and the loss ratio is 10^{-6} . The buffer size at each gateway is set to 200 packets.

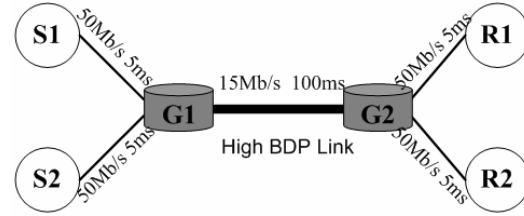


Figure 1. Network Simulation topology

A. Performance of slow-start mechanism of Vegas-W

In this subsection, we investigate the throughput and $cwnd$ (congestion window size) of TCP Vegas, TCP Vegas-W and NewReno to explore the performance of slow-start mechanism of Vegas-W. A TCP connection from $S1$ to $R1$ is Vegas, Vegas-W or NewReno. And a CBR traffic flow from $S2$ to $R2$ with 6 Mb/s rate starts at 80 second and stops at 150 second to simulate the variety of network bandwidth. So the real bandwidth of bottleneck link is 9Mb/s between 80s and 150s, and the rest of the time is 15Mb/s. Fig.2 and Fig.3 show the congestion window size and throughput, respectively.

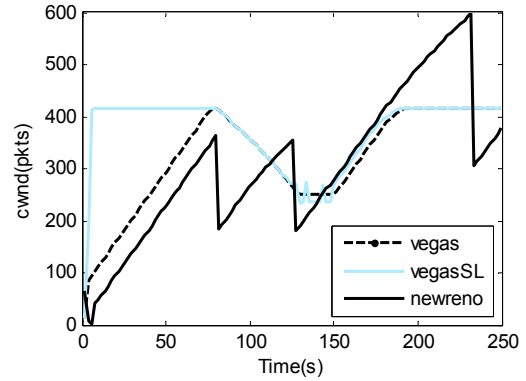


Figure 2. Cwnd comparison of Vegas, Vegas-W and NewReno

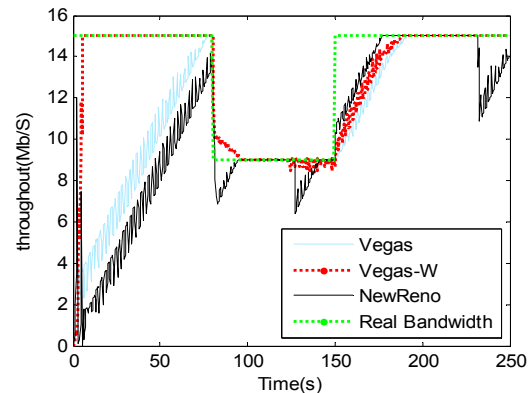


Figure 3. Throughput comparison of Vegas, Vegas-W and NewReno

By observing the congestion window size evolution shown in Figure 2 and throughput evolution shown in Figure 3, we can find that mechanism of Vegas-W extend the time

of the slow-start phase and raise the rate of getting to the available bandwidth. We can find Vegas-W needs only 6.4s to get the full network utilization (15Mb/s), The start-up performance is highly improved. And when the bandwidth changes sharply (80-150s), Vegas-W responds swiftly.

B. Experiment of route changing

In this subsection, we explore the performance of TCP Vegas-W when its route changes caused by reroute. S1-R1 is a Vegas-W connection. The route begins to change at 50 second, and changes to the original route at 100 second. The new route has larger bandwidth and longer propagation delay. The objective of the simulation scenario is to explore performance of congestion avoidance mechanism of Vegas-W.

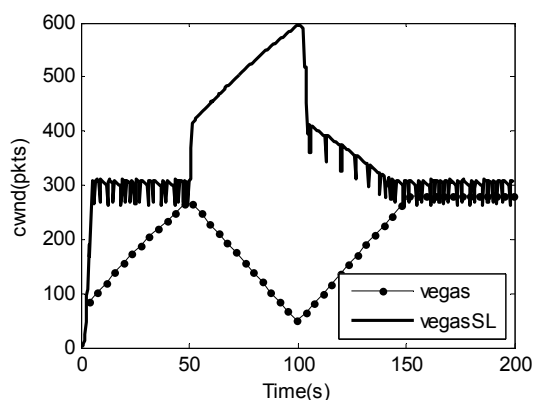


Figure 4. CWND comparison between Vegas and Vegas-W

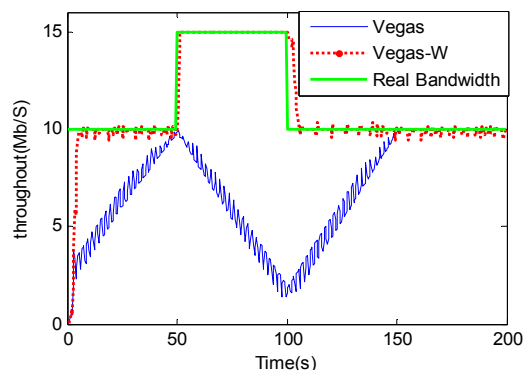


Figure 5. Throughput comparison between Vegas and Vegas-W

By observing the congestion window size evolution shown in Figure 4 and throughput evolution shown in Figure 5, we can find that when route changes, RTT increases with propagation delay increases. However, BaseRTT can not change adaptively accordingly. RTT increasing is considered as congestion arrival. But in fact, the bandwidth of the new path is larger than the old one. The congestion window should increase, while the congestion window of Vegas decreases ridiculously. In this scenario, Vegas-W performs much better than Vegas.

VI. CONCLUSION

In this research, we propose an improved version of TCP Vegas named Vegas-W for GEO satellite networks. Vegas-W improves Vegas by route surveillance and bandwidth estimation. In slow-start phase, TCP Vegas raises the rate of getting to the available bandwidth, and improves the start-up performance. In congestion avoidance phase, link status is monitored by monitoring RTT. A more accurate BaseRTT can be achieved and congestion level can be estimated more precisely. In retransmission and recovery phase, we distinguish the packet-lost reason and solve the problem of misjudgment of congestion level caused by link error. Simulation results show that Vegas-W reacts better in GEO satellite network environment and therefore improves the overall performance. Using route surveillance and bandwidth estimation in Vegas is a novel method. It performs well, but also has time and space overhead. So how to combine these two aspects would be our future work.

REFERENCES

- [1] V. Jacobson. Modified TCP Congestion Avoidance Algorithm. Technical report, Technical report, April 1990.
- [2] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP Vegas: New techniques for congestion detection and avoidance," in Proc. ACM SIGCOMM'94, no. 4, pp. 24-35, Aug. 1994.
- [3] L. S. Brakmo and L. L. Peterson, "TCP Vegas: End to end congestion avoidance on a global Internet," IEEE J. Select. Areas Commun., vol. 13, pp. 1465-1480, Oct. 1995.
- [4] J. S. Ahn, P. B. Danzig, Z. Liu, and L. Yan, "Evaluation of TCP Vegas: Emulation and experiment," in Proc. ACM SIGCOMM'95, vol. 25, pp. 185-195, Aug. 1995.
- [5] G. Hasegawa, M. Murata, and H. Miyahara, "Fairness and stability of congestion control mechanism of TCP," Telecommunication Systems Journal, pp. 167-184, Nov. 2000.
- [6] Harchol Balter M , Crovella M E , Murta C D . On Choosing a Task Assignment Policy for a Distributed Server System. Journal of Parallel and Distributed Computing , 1999 ,59 (2) :2042228
- [7] S. Mascolo, M.Y. Sanadidi, C. Casetti, M. Gerla, and R. Wang, "TCP Westwood: End-to-End Congestion Control for Wired/Wireless Networks," Wireless Networks J., vol. 8, pp. 467-479, 2002.
- [8] Ferorelli, R.; Grieco, L.A.; Mascolo, S.; Piscitelli, G.; Camarda, P.; "Live Internet measurements using Westwood+ TCP congestion control ", 2002 GLOBECOM '02. IEEE, Volume: 3, 17-21 Nov.2002 Pages:2583-2587 vol.3.
- [9] Li, S. Q. and C. Hwang, "Link Capacity Allocation and Network Control by Filtered Input Rate in High speed Networks", IEEE/ACM Transactions on Networking, vol.3, no. 1, Feb. 1995, pp. 10-25