Research Journal of Information Technology

ISSN 1815-7432

Academic Journals Inc.

www.academicjournals.com

Research Journal of Information Technology 5 (1): 45-52, 2013
ISSN 1815-7432 / DOI: 10.3923/rjit.2013.45.52
© 2013 Academic Journals Inc.

An Improved Vegas Algorithm for Enhancing Compatibility with TCP Reno Based on Game Theory

[1]Hua Zhang, [1,2]Guang Sun and [1]Minghua Liao

[1]Department of Information Management, Hunan College of Finance and Economics, No.139. Fenglin 2nd Road, Changsha, 410205, China

[2]School of Computer and Communication, Hunan University, No. 252. Lushan South Road, Changsha, 410082, China

Corresponding Author: Hua Zhang, Department of Information Management, Hunan College of Finance and Economics, No.139. Fenglin 2nd Road, Changsha, 410205, China

ABSTRACT

TCP (Transmission Control Protocol) Vegas algorithm can provide a better performance compared to the traditional TCP Reno scheme. However, there exists a serious incompatibility problem of unfair bandwidth sharing in favor of TCP Reno when TCP Vegas and TCP Reno connections coexist and share the same link in wired network. To solve this problem, we analyze the compatibility of TCP Vegas with TCP Reno and propose an enhanced Vegas algorithm G-Vegas by using the method of Game Theory. Through, the simulation experiment, the results show that G-Vegas algorithm improves the compatibility with TCP Reno and it has a better capacity in competition with the Reno for bandwidth than other improved Vegas algorithms.

Key words: Congestion control, TCP Vegas algorithm, game theory, compatibility

INTRODUCTION

Since, the beginning of the computer network, especially in the last ten years, it is happening explosive growth. Meanwhile, it has become an important and indispensable part to the government departments, enterprises and social organizations. Because the network has been getting a high degree of attention, it has more and more fast development. However, with the rapid development of network, the network congestion problem becomes more and more serious (Shenker, 1995).

It is the fundamental cause of network congestion that the load of the network of users providing is greater than the network resources capacity and its processing power. Network congestion has many forms, including packet delay increasing, packet loss rate increasing and the upper application system performance declining and so on (Jasem *et al*., 2010).

The direct reason of network congestion can be attributed to the following three aspects (Luo and Lin, 2001).

One is the network bandwidth capacity lack. According to the Shannon's information theory (Shannon and Weaver, 1949) the sending rate of all source must be less than or equal to channel capacity, otherwise, it is unable to realize error-free transmission. For network, the network bandwidth bottlenecks will form in the low speed link. When the network can't satisfy the

requirements of its source bandwidth, the network congestion will happen. Two is that the network port storage capacity is limited. When multiple data streams share a port, a data message queue will form in this port, when the message queue length is more than the length of the port cache, lost package will appear. This is congestion. If a message queuing time is too long, overtime will appear even if the port of caching ability is unlimited. This is a form of congestion too. Three is that the processor computing power is not enough.

The profound reason of network congestion is the internet design mechanism. The model of internet can be abstract described using the following points (Peterson and Davie, 2000):

- **Message switching network:** Message switching improves the efficiency of resource utilization through the sharing and it is better than circuit switching. But, the service quality under the way of sharing is inferior to circuit switching (Bennett *et al*., 1999). Message disorderly sequence phenomenon of the message switching network influences the quality of service and the complexity of the source system will be increased to process disorderly sequence messages
- **Connectionless network:** It doesn't need to establish a connection link between the communication nodes in internet. This kind of no connection model has simplified the design of network and made the network nodes that it need not save and link relative information. However, it is difficult to introduce the "admission control" algorithm in the no connection model. If the users' needs are more than the loads of the network resources, the service quality can't guarantee. In addition, the no connection model is hard to track the data source leading to network security hidden dangers; at the same time, the no connection network is one of the main reasons of disordered packets
- **Best-effect service model:** The best-effort network doesn't guarantee service quality of data transmission and just do it best. As for the traditional applications, namely FTP, SMTP, etc., it isn't sensitive to changes of network performance. But the best-effect service model can satisfy. However, for the applications of multimedia in later, because they are sensitive to changes of network performance, the best-effort service model has difficulties in these applications

In order to solve the network congestion problem, many of the congestion control algorithms have appeared. These algorithms can be divided into end-to-end congestion control algorithms and congestion control algorithms based on the network.

In end-to-end congestion control algorithms, Vegas algorithm has good performance with active avoid congestion (Meng, 2006). But, Reno algorithm is still the current mainstream algorithm. When Reno algorithm and Vegas algorithm coexist in network, there will appear bandwidth loss. In order to solve this problem, in this study, we propose an improved algorithm, namely G-Vegas.

INTRODUCTION TO THE ALGORITHMS

The Reno algorithm: The Reno algorithm was put forward in 1960 by Jacobson and it network is the most widely used TCP congestion control algorithm at present (Zheng *et al*., 2002). Four-fold mechanisms are included in this algorithm: Slow start, Congestion avoidance, Fast retransmit and Fast recovery. The algorithm can be expressed as follows:

```
Initial ();//Initialize send window win; the source end send window awin
Win=min(cwnd,awin),cwnd=1;
If (cwnd<ssthresh)
   Cwnd=cwnd+1;//slow start
Else
   Cwnd=cwnd+1/cwnd;//congestion avoidance
Timeout occurred:
  Ssthresh=max(2,min(cwnd/2,awin));//update ssthresh
Cwnd=1;
```

Cwnd says the window size at present, namely the number of data packets. Ssthresh is defined as the threshold value of the congestion avoidance algorithm. The Reno algorithm predicts and processes the network state after network congestion has occurred. So, it does not effectively prevent the occurrence of network congestion, this is its shortcomings. In addition, in the coexistence network of the multiple internet protocols, the Reno algorithm will as more as possible increase congestion window to possess the bandwidth; so, it is unfavorable to the Vegas algorithm.

The Vegas algorithm (Kim and Choi, 2003): The Vegas algorithm adopts a new retransmission mechanism in comparison to the Reno algorithm. It uses a repetitive ACK packet but not three repeat ACK packets in the Reno algorithm to judge overtime and improves the timeliness of the detection congestion. The Vegas algorithm also cautiously increases the congestion window in Slow Start to make the exponential growth every other RTT (round-trip time).

It is the core idea of the Vegas algorithm that it observes the value of RTT and compares the expected rate with the actual rate to adjust the congestion window size. The Vegas algorithm is as follows:

```
Expected=cwnd/BaseRTT;//calculating
                          the Expected rate
Actual=cwnd/RTT;// calculating the Actual rate
Diff=Expected-Actual;
      Cwnd(t)+1 Diff<α/BaseRTT
Cwnd(t+1)={Cwnd(t)α/BaseRTT<Diff<β/BaseRTT
Cwnd(t)-1 Diff>β/BaseRTT
```

Base RTT says the minimum numerical loop response time of observation, generally takes the RTT value of the first packet sent after the connection is established. α and β are defined two threshold, general setting $\alpha = 1$ and $\beta = 3$. Vegas algorithm hopes to keep a certain number of messages in network queue on order to improve the system performance.

IMPROVED VEGAS ALGORITHM G-VEGAS

The Vegas algorithm uses the active congestion avoidance mechanism, reduces congestion window before actual losing packets happening. And the Reno algorithm adopts reactive congestion avoidance mechanism, finds lost packages after increasing congestion window (Wang *et al*., 2009). Consequently, the Vegas algorithm has more superior congestion control performance than the Reno algorithm. But, in the co-existence, the Reno will steal the Vegas's bandwidth. This compatibility limits the Vegas algorithm application. In order to overcome the limitations, this study

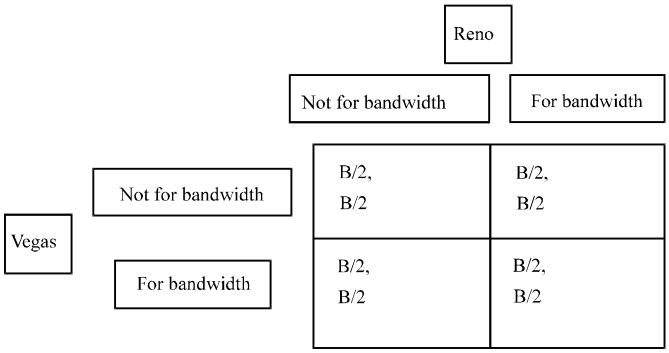

Fig. 1: Prisoner's game model

proposes an improved algorithm G-Vegas. It attempts from congestion avoidance algorithm to solve the problems caused by the unfair competition, improves fairness of the bandwidth competition.

This kind of unfair competition applies the classic prisoner's dilemma the prisoner and it is shown in Fig. 1. It can be seen from the graph that this is a zero-sum game, B for the network total capacity. The Reno algorithm always wants to rob the bandwidth so the system stability point will occur in the second case, the Vegas for B/2-A, the Reno for B/2+A. If the Vegas algorithm does not be improved, it is always at a disadvantage in the game.

The Reno algorithm always pursues the maximization of bandwidth of its own nodes. In order to maintain the bandwidth of the nodes in the Vegas algorithm, these nodes must lead the game to the prisoner's dilemma stable state in a certain bandwidth competition way.

In order to solve the bandwidth competition, we introduce the bandwidth competition perception algorithm and competitive situation change algorithm. They are used to balance the advantages of the Vegas algorithm and bandwidth competition ability of the Reno algorithm and enhance compatibility between the Vegas and Reno algorithm. In the new algorithm, in normal circumstances, we use the traditional Vegas algorithm to adjust the window and the bandwidth competition perception algorithm to assess whether there is the unfair competition. In this way, the bandwidth of in the Vegas algorithm is long-term below the nodes in the Reno algorithm. If the unfair competition appears, we use the competitive situation change algorithm to participate in the competition. So, it makes the Reno nodes perceive the network congestion and adjust its bandwidth state at the same time.

The bandwidth competition perception algorithm assesses whether the unfair distribution of bandwidth resources is happened by surveillance of TCP window size. The thought is shown in Fig. 2. Unfair competition mainly refers to the Reno algorithm taking up larger bandwidth and the Vegas algorithm maintaining lesser bandwidth. If found unfair competition, we will use the competitive situation change algorithm to break this situation. It makes the nodes in the Reno algorithm perceive the unfair distribution of bandwidth and turn down related parameters, such as the threshold value of window and so on. So, it realizes fair distribution of resources. We use the Reno algorithm to achieve the competitive situation change algorithm, promote the nodes in the Reno algorithm perception network status and adjust the related parameters by the same way competition.

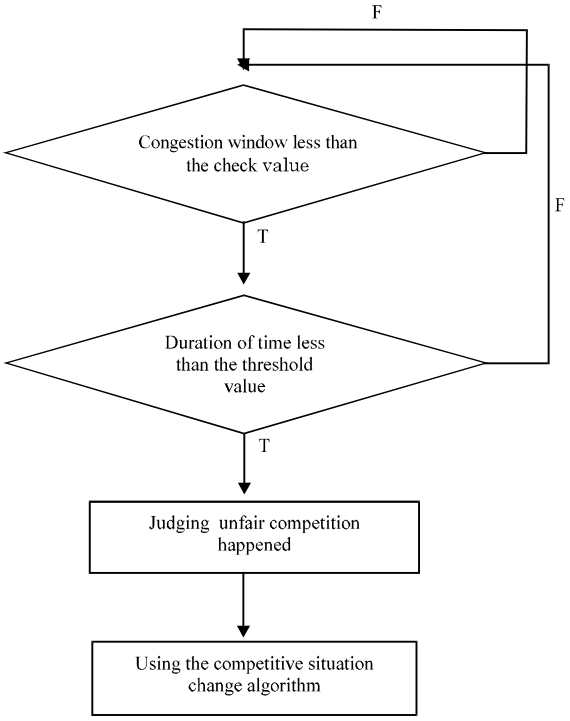

Fig. 2: The bandwidth competition perception algorithm

The improved congestion control algorithm-G-Vegas can be expressed as follows:

```
Initial();
Win=min(cwnd,awin),cwnd=1,ssthresh=64k,
tcwnd<ssthresh=0;
If (tcwnd<ssthresh<T)
  Update the value of tcwnd<ssthresh;
  Expected=cwnd/BaseRTT;
  Actual=cwnd/RTT;
  Diff=expected-actual;
  If (Diff<α/BaseRTT)
Cwnd(t+1)=cwnd(t)+1;
Else if (α/BaseRTT<Diff<β/BaseRTT)
  Cwnd(t+1)=cwnd(t);
Else if (Diff>β/BaseRTT)
  Cwnd(t+1)=cwnd(t)-1;
Else
 If (cwnd<ssthresh)
  Cwnd=cwnd+1;
Else
  Cwnd=cwnd+1/cwnd;
Timeout occurred:
   Ssthresh=max(2,min(cwnd/2,awin));
   Cwnd=1;
   tcwnd<ssthresh=0;
```

The updated algorithm of tcwnd<ssthresh is as follows:

```
T0 for the initial time, t for the current time;
If (cwnd<ssthresh)
  tcwnd<ssthresh=t-t0;
else
  if (tcwnd<ssthresh<T)
    tcwnd<ssthresh=0;
    t0=t;
```

SIMULATION EXPERIMENT

Algorithm simulation uses NS2 software (Xu *et al.*, 2003) and network topology structure is shown in Fig. 3. Among them, the S1 uses TCP Reno algorithm and the S2 uses TCP Vegas algorithm. Network designs for two transmitting nodes S1 and S2, two router nodes R1 and R2 and two receiving nodes D1 and D2. The bandwidth is 8 Mb sec^{-1} from S1, S2-R1, from R2-D1, D2 and the propagation delay is 3 msec; the bandwidth is 1.5 Mb sec^{-1} from R0-R1, propagation delay is 3 msec. Bottleneck bandwidth uses Drop Tail strategy, the simulation continues 80 sec.

In Fig. 3, the S1 uses TCP Reno algorithm and S2 uses TCP Vegas algorithm. The Fig. 4 shows average throughput of the TCP Vegas and TCP Reno sharing bottleneck bandwidth. From the graph shows, Reno throughput is far larger than the Vegas and two algorithms exist in obvious inequality on bandwidth competition.

Figure 5 shows the congestion window of the Vegas and Reno sharing bottleneck bandwidth. Because the Vegas algorithm uses the active congestion avoidance mechanism, the congestion window is relatively stable compared with the Reno. But, because of the Vegas actively avoiding congestion, it leads excess bandwidth to the Reno, thus, the Vegas algorithm is at a disadvantage in the competition bandwidth.

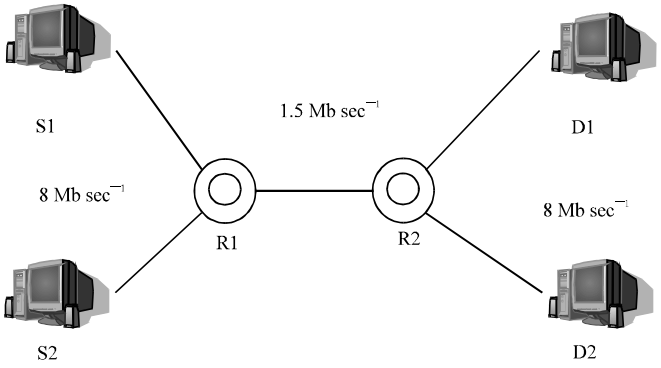

Fig. 3: Network simulation topology

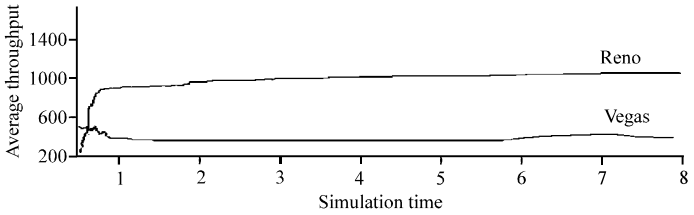

Fig. 4: The simulation results of average throughout

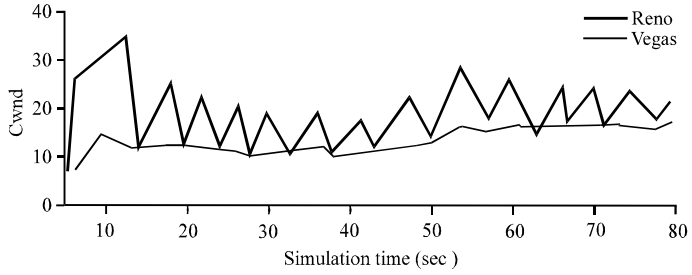

Fig. 5: The congestion window of the Vegas and Reno in bandwidth sharing

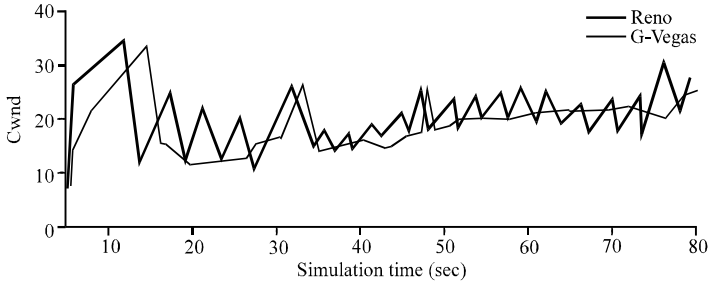

Fig. 6: The congestion window of the G-Vegas and Reno in bandwidth sharing

Figure 6 shows the congestion window of the Reno and the improved congestion control algorithm G-Vegas sharing bottleneck bandwidth. It can be seen from the comparison of Fig. 6 and Fig. 5 that the bandwidth competition ability of G-Vegas algorithm is obviously improved. At the same time, in the G-Vegas algorithm, the congestion window maintains the stability at most of the time; only when in the Reno preemptive too much bandwidth, it will promote the Reno to make room for more bandwidth through the competitive situation and realize the compatible with the Reno. From the simulation results, it can be found that the overall performance of the network is distinctly improved compared with Fig. 5.

CONCLUSION

In this study, we proposed the end-to-end TCP congestion control algorithm, focusing attention on the Vegas algorithm. Firstly, we introduced the compatibility of TCP Vegas with TCP Reno. Then we analyzed the existing problems and the solutions of Vegas algorithm from the perspective of game theory. We proposed an enhanced Vegas algorithm-G-Vegas by using the method of Game Theory. Finally, we achieved the simulation experiment on the NS2 platform to verify the effectiveness. Simulation results showed that G-Vegas algorithm improves the compatibility with TCP Reno and had a better capacity in competition with the Reno for bandwidth than other improved Vegas algorithms.

ACKNOWLEDGMENT

This study is supported by the higher school science research project of Hunan Province of China No. 11C0216.

REFERENCES

Bennett, J.C.R., C. Partridge and N. Shectman, 1999. Packet reordering is not pathological network behavior. IEEE/ACM Trans. Networking, 7: 789-798.

Jasem, H.N., Z.A. Zukarnain, M. Othman and S. Subramaniam, 2010. The delay with new-additive increase multiplicative decrease congestion avoidance and control algorithm. Inform. Technol. J., 4: 1327-1335.

Kim, K. and C.H. Choi, 2003. Queue delay estimation and its application to TCP Vegas. Comput. Networks, 43: 619-631.

Luo, W. and C. Lin, 2001. TCP/IP congestion control study. Comput. J., 24: 1-18.

Meng, C., 2006. The Research of TCP Vegas Algorithm. Sichuan University Press, Chengdu, China.

Peterson, L.L. and B.S. Davie, 2000. Computer Networks: A System Approach. 2nd Edn., Morgan Kaufmann Publishers, New York, USA., ISBN-13: 9781558605145, Pages: 748.

Shannon, C.E. and W. Weaver, 1949. The Mathematical Theory of Communication. University of Illinois Press, Urbana.

Shenker, S., 1995. Fundamental design issues for the future internet. IEEE J. Sel. Areas Commun., 13: 1176-1188.

Wang, Y.T. and J.A. Fang, 2009. Enhanced TCP congestion control algorithm based on TCP Vegas. J. Tsinghua Univ. (Natl. Sci. Edn.), Vol. 26.

Xu, L., B. Pang and Y. Zhao, 2003. NS and Network Simulation. Posts and Telecom Press, Beijing, China.

Zheng, Y., M. Lu and Z. Feng, 2002. Evolutionary game theoretic model of TCP Reno/Vegas algorithms. J. Tsinghua Univ. (Sci. Technol.), 7: 970-973.