FUZZY OPTIMIZED CONGESTION CONTROL IN TCP/IP

Prof.Deepa Jose[1]

Abstract—**Congestion occurs in the network when arrival rate to a router is greater than its departure rate. In this paper, using fuzzy logic approach, we have proposed a modified TCP delay-based congestion avoidance mechanism which is based on traditional TCP-Africa algorithm. Here we present our fuzzy controller which is expected to act as a congestion controller in the routers. The proposed fuzzy controller, uses queue delay and link capacity as input linguistic variables. The output of fuzzy controller is the window size to which the sending window must be adjusted . Both the current standard and most of the experimental congestion control methods are fundamentally loss-based, that is they rely on packet loss to detect that the network is above full capacity**

Keywords-fuzzy,TCP/IP,bandwidth,congestion ,queue delay,window size

---

[1]**Lecturer,**MCA Department,Y.M.T College of Management,Kharghar,Navi Mumbai ,Maharashtra,deepa_jos@rediffmail.com

1.Introduction

In the intervening years, TCP's standard congestion control (Tahoe, Reno) has evolved in small ways, but no major redesign has taken place. Various problems have been shown, particularly in its achieved throughput on large bandwidth-delay product (BDP) networks and its propensity to cause high latency on network links with large available queues. In response to this, various experimental schemes have been proposed and implemented. These algorithms have not as yet been ratified by the standards body.

Figure 1 shows the growth of data traffic in the Internet for the past five years and the expected traffic for 2010. As it is

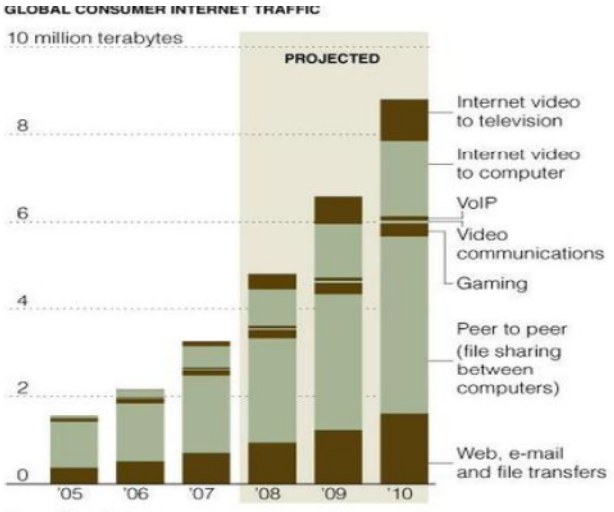

Figure 1: Exponential growth in the Internet traffic (source: CISCO systems survey)

observed in the picture the peer to peer file sharing, Internet video to television/ computer and web, email applications constitute to a major part of the rapid increase in the global Internet traffic. Consideration of queuing delay inevitably leads to consideration of delay-based congestion control algorithms. Potentially, allocation of network bandwidth between competing

sources can be achieved while maintaining low queuing delay [1](even when network buffers are large, unlike with loss-based algorithms), and with almost full utilization of network links. High utilization with low queuing delay is termed "operating at the knee of the curve" and is evidently a desirable property. Delay-based congestion control algorithms have of course been widely studied, with Vegas and FAST, receiving particular attention.

It is presumed that the loss is caused by a full network queue. Both the current standard and most of the experimental congestion control methods are fundamentally loss-based, that is they rely on packet loss to detect that the network is above full capacity. It is presumed that the loss is caused by a full network queue. However, a second means of detecting incipient congestion proposed in the late eighties is to observe that the round trip time of a packet increases as network queues build . This has the advantage of warning early rather than merely waiting until the network is over-utilized and packets are lost.

Using queuing delay as the congestion measure has two advantages. First, queuing delay can be more accurately estimated than loss probability both because packet losses in networks with large bandwidth-delay product are rare events (probability on the order 10−8 or smaller), and because loss samples provide coarser information than queuing delay samples. Indeed, measurements of delay are noisy, just as those of loss probability. Each measurement of packet loss (whether a packet is lost) provides one bit of information for the filtering of noise, whereas each measurement of queuing delay provides multi-bit information. This makes it easier for an equation-based implementation to stabilize a network into a steady state with a target fairness and high utilization. Second, the dynamics of queuing delay seems to have the right scaling with respect to network capacity. This helps maintain stability as a network scales up in capacity [2].

TCP's congestion management is composed of two important algorithms. The slow-start and congestion avoidance algorithms allow TCP to increase the data transmission rate without overwhelming the network. They use a variable called CWND (Congestion Window)[3]. TCP's congestion window is the size of the sliding window used by the sender. TCP cannot inject more than CWND segments of unacknowledged data into the network[3].

TCP ensures that arrival of all packets at their destination is confirmed, retransmitting any which are lost; ensures that any reordering of the packets is corrected, keeps different communication sessions from interfering and attempts to send at the highest rate it can without causing excessive packet loss due to congestion in the network[4,5]. This last responsibility of TCP is called "congestion control" and was added in the late eighties in response to several instances of congestion collapse on the early internet.

2.Related works

The Internet is increasingly facing packet loss and queue delays due to its rapid growth. This may lead to a congestion collapse that will reduce the quality of Internet applications. Active queue management (AQM) techniques try to detect and react to the congestion before its consequences such as packet drops or queuing delays. In reaction to a suspected congestion, AQM algorithms proposed in either drop packets early or mark them as ECN to inform the congestion to the traffic sources. The most important differences among AQM schemes are the guess congestion time determination and the marked/dropped packets selection.

. The DT Algorithm

In this algorithm, only when a packet loss occurs due to the queue overflow, sources are informed to reduce their transmission rates. The time between detecting a packet loss in the router and informing the source to reduce its rate causes a large number of packets to be dropped since the router cannot provide space for incoming packets which have been sent with the previous rate. This algorithm keeps the queues always full. It is unfair with bursty traffics and can affect in lockouts[6] .

. The RED Algorithm

This algorithm is one of the first proposed solutions to the active queue management. RED simply sets minimum and maximum dropping thresholds, which are presented by minth and maxth , respectively. If the average queue size (avg) exceeds the minimum threshold, RED starts dropping packets based on a probability depending on the average queue size[7]. If average queue size exceeds the maximum threshold, then every packet is dropped. A pseudocode of the RED algorithm is given below:

For every packet arrival {

Calculate avg

If( )

{

Drop the packet

}

else if (avg > minth) {

Calculate the dropping probability pa

Drop the packet with probability pa, otherwise forward

it

}

else {

Forward the packet

}

}

As expressed RED contains severe problems. The fundamental one is that it uses queue length as a congestion indicator. This indicator cannot completely show the severity of congestion. On the other hand, average queue length varies with the level of congestion as well as with the parameter settings [8].

As a result, the queuing delay of RED is too sensitive to a traffic load and parameter settings .Different variants of RED such as Stabilized RED (SRED) and Adaptive RED (ARED) have been proposed to fix some of its shortcomings.

. The BLUE Algorithm

The main idea of this algorithm is to use different parameters such as packet loss and link utilization to detect a congestion and adjust the rate of packet dropping/marking [6]. BLUE uses single probability, Pm,which is used to mark or drop packets when they are enqueued. If the queue is continually dropping packets due to overflow, Pm is increased. If the link is underutilized,the probability is decreased. The amount of increase and decrease are d1 and d2, respectively. The rate of updating Pm is 1/freeze_time. This allows Pm to take effect before the next update. The probability is also updated when the queue length exceeds a

threshold L. In other words, it does not let the queue become full. This allows space to be reserved in the queue for occasional bursts. The BLUE algorithm is given below. Notice that "now" corresponds to the current time while "last_update" corresponds to the last time Pm was tuned.

Upon packet loss (or queue length > L) event

If ((now-last_update) > freeze_time) then

Pm = Pm + d1

last_update = now

Upon link idle event

If ((now-last_update) > freeze_time) then

Pm = Pm - d2

last_update = now

The important features of BLUE are the minimal amount of buffer space, the reduction of end to end delay, the high link utilization and a very low rate of packet loss .

.TCP-Africa[7]

TCP-Africa is a delay sensitive congestion avoidance mode algorithm that allows for scalable, aggressive behavior in large underutilized links , yet falls back to the more conservative TCP-Reno algorithm once links become well utilized and congestion is imminent. TCP-Africa promises excellent utilization, efficiency, and acquisition of available bandwidth, with significantly improved safety, fairness, and RTT bias properties .

When one goes about designing a high speed transfer protocol, the most straightforward approach is to modify TCP's increase and decrease parameters to adjust its response function to provide a more desirable performance. However, there are several considerations that need to be taken into account when designing such a protocol.TCP-Africa, in its scalable "fast" mode, quickly grabs available bandwidth. Once the delay begins to increase, and the protocol senses that the amount of available bandwidth is becoming small, the protocol switches to a conservative, linear increase mode of one additional packet per round trip time. The motivation behind such a slow growth mode for a high speed protocol is that once a flow is within a reasonable percent of its maximum bandwidth, it no longer has any reason to be growing at an increased rate.

3.Proposed fuzzy based approach

Choosing a large value for the α parameter of these delay based protocols can lead to an improved ability to compete with Reno-like flows, but the correct choice of α will depend on the number of competing flows and the buffer capacity of the link[9] .

The main part of designing the Fuzzy TCP-AFRICA is to design the Fuzzy TCP-AFRICA Controller (FTAC) in a way to act as a congestion controller in the routers. There exist two different methodologies for design of fuzzy logic controller: trial-and-error approach and the theoretical approach[10]. To design Fuzzy TCP-AFRICA, we used the trial-and-error approach. In this approach, a set of IF-THEN rules are collected from an introspective verbalization of experience-based knowledge; then fuzzy controllers are constructed from these fuzzy IF-THEN rules; finally, the fuzzy controllers are tested and if the performance is not satisfactory, the rules are fine-tuned or designed in a number of trial-and-error cycles until the performance is satisfactory.

/* Linguistic rules of FTAC */
Rule 1: if link utilization is small and queue delay is low then α is high;
Rule 2: if link utilization is small and queue delay is medium then α is high;
Rule 3: if link utilization is small and queue delay is high then α is moderate;
Rule 4: if link utilization is medium and queue delay is low then α is moderate;
Rule 5: if link utilization is medium and queue delay is

medium then α is moderate;
Rule 6: if link utilization is med and queue delay is high then α is low;
Rule 7: if link utilization is big and queue delay is low then α is moderate;
Rule 8: if link utilization is big and queue delay is medium then α is low;
Rule 9: if link utilization is big and queue delay is high then α is low;

CONCLUSION

Fuzzy logic control, in turn, is a well-known technique for designing feedback control systems. The application of fuzzy control technique to the problem of congestion control is appropriate due to the problems we will face in obtaining a precise model using conventional methods. In this paper, we introduced the Fuzzy TCP-AFIRCA which acts more effective and robust in comparison with other approaches including the traditional TCP-AFRICA. key towards succeeding at both challenges is to use delay information as an indicator for the appropriate level of aggressiveness. we note that there may be more than one way to use delay information to achieve safe, fast utilization of network resources, while having acceptable performance against traditional greedy flows.

REFERENCES

[1] V. Jacobson, "Congestion avoidance and control," *ACM SIGCOMM*, pp. 314–329, 1988.

[2] S. Floyd, "Highspeed tcp for large congestion windows," *RFC 3649, Experimental*, Dec. 2003.

[3] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," *ACM SIGCOMM*, 1998.

[4] C. Jin, D. X. Wei, and S. H. Low, "FAST TCP: motivation, architecture, algorithms, performance," *IEEE INFOCOM*, 2004.

[5] M. Kwon and S. Fahmy, "A comparison of load based and queue-based active queue management algorithms", in Proc. SPIE ITCom, vol. 4866, Aug. 2002.

[6] M. H. Yaghmaee, M. Menhaj, H. Amintoosi, "A fuzzy extension to the blue active queue management algorithm", IAEEE, Journal of Iranian Association of Electrical and Electronics Engineers, vol. 1, No. 3,Winter 2005, pp. 3-14.

[7] R. King, R.Riedi, and R. Baraniuk , "TCP-Africa: An Adaptive and Fair Rapid Increase Rule for scalable TCP," in Proc. of INFOCOM 2005, Miami,FL, USA, March 2005.

[8] W. Feng, et.al., "Stochastic fair blue: a queue management algorithm forenforcing fairness", In Proc. of IEEE INFOCOM, Apr 2001.

[9] D. Chiu, R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks", Computer Networks and ISDN systems, 17:1-14, 1989

[10] Raj Jain, "A delay-based approach for congestion avoidance in interconnected heterogeneous computer networks," ACM ComputerCommunication Review, vol. 19, no. 5, pp. 56–71, Oct. 1989