ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

Using Fuzzy Logic Avoid Congestion and Buffering To Provide Intellectual Data Traffic Administration for Internet

Mr.Chougule Meghraj Balasaheb

Mtech (C.S.E.), St.Mary's Group of Institutions, JNTU ,Hyderabad, India

Abstract: FLC (Fuzzy Logic Control) [11] has been considered for IC (Intelligence Control). It is a methodology used to design robust systems that provide factors such as system nonlinearity, parameter uncertainty, measurement and modelling imprecision. In fact, fuzzy logic control showed extraordinary and mature control performance in accuracy, transient response, robustness and stability [12], [13]. Fuzzy logic control use to avoid buffering using by avoiding variation of queuing delay (Queuing jitter) due to the dynamics queue length. Queuing flows can be buffered on the receiving side before being delivered. Buffering them does not affect the reliability or bandwidth, and increases the delay, but it smooth out the jitter.

Index Terms: Congestion control, fuzzy logic control, quality Of service, max-min fairness, robustness, traffic management, buffering, jitter.

I. INTRODUCTION

Network traffic management is a core area of research that is of great importance in the field of communication. Although many models have been proposed since time being, all those techniques have their own drawbacks. This paper proposes a new scheme for traffic management by exploiting the possibilities of fuzzy logic.

To provide intelligent traffic management service, routers are deployed with intelligent data rate controllers to control the data traffic. In order to control data traffic mass, Traffic control protocols are necessary to estimate network parameters e.g., link latency, bottleneck bandwidth, packet loss rate, or the number of flows to compute the allowed source sending rate. Our fuzzy-logic-based controller can measure the router queue size directly; hence it avoids various potential performance problems arising from parameter estimations while reducing much consumption of computation and memory resources in routers. As a network parameter, the queue size can be accurately monitored and used to proactively decide if action should be taken to regulate the source sending rate, thus increasing the resilience of the network to traffic congestion and buffering. The communication QoS (Quality of Service) is assured by the good performances of our scheme such as max-min fairness, low queuing delay and good robustness to network dynamics. Queuing flows can be buffered on the receiving side. Simulation results and comparisons have verified the effectiveness and showed that our new traffic administration scheme can achieve better performances.

II. RELATED WORK

Any information that we have searched may be analysed by any web search and follows books and refers authors.
Jungang Liu from Wuhan University of Technology, China. He researched Internet traffic control, modelling and performance evaluation of computer networks, industrial process control, and automation.
Oliver W.W. Yang, Ontario, Canada. He researched the modelling, analysis, and performance evaluation of computer communication networks, their protocols, services, and interconnection architectures. The CCNR Lab under his leadership has been working on various projects in switch architecture, traffic control, traffic characterization, and other traffic engineering issues in both wireless and photonic networks, while great efforts have been made to find the information of Fuzzy Logic to avoid congestion and buffering to provide intellectual data traffic administration for Internet.

III. METHODOLOGY

In buffering we see a stream of packets being delivered with substantial jitter. Packet 1 is sent from the server at t = 0 sec and arrives at the client at t = 1 sec. Packet 2 undergoes more delay and takes 2 sec to arrive. As the packets arrive, they are buffered on the client machine. So if we want to avoid buffering then it is necessary to avoid variation of queuing delay due to the dynamics queue length.

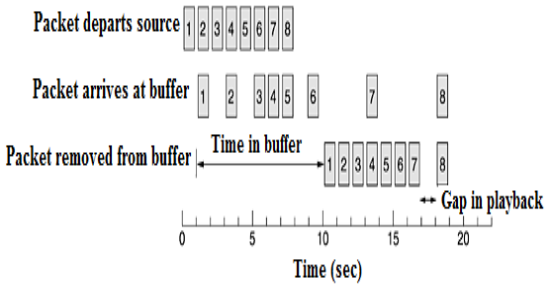

Fig.1.Smoothing the output stream by buffering packets

At t = 10 sec, playback begins. At this time, packets 1 through 6 have been buffered so that they can be removed from the buffer at uniform intervals for smooth play. Unfortunately, packet 8 has been delayed so much that it is

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

not available when its play slot comes up, so playback must stop until it arrives, creating an annoying gap in the music or movie. This problem can be alleviated by delaying the starting time even more, although doing so also requires a larger buffer. Commercial Web sites that contain streaming audio or video all use players that buffer for about 10 seconds before starting to play.

Congestion notification is slightly more complex than buffering and it is typically used in conjunction with buffering to eliminate its major problems. With congestion notification, when a device's buffers begin to fill (or it notices excessive congestion through some other method), it sends a message to the originating station basically saying "Slow down!" When the buffers are in better shape, it then relays another message stating that transmission can begin again. The obvious problem with this situation is that in a string of intermediate devices (such as routers), congestion notification just prolongs the agony by filling the buffers on every router along the path.

For example, imagine Router A is sending packets to Router C through Router B (as In Figure 2)

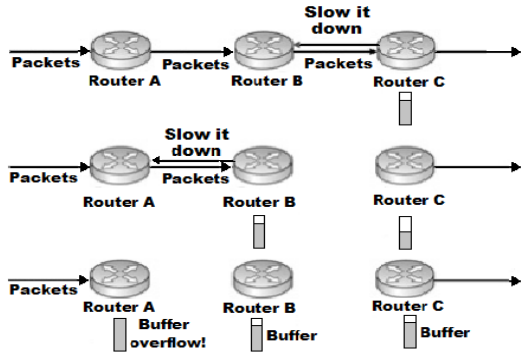

Fig.2.The problems with bffering and congestion notification.

As Router C's buffer begins to fill, it sends a congestion notification to Router B. This causes Router B's buffer to fill up. Router B then sends congestion notification to Router A. This causes Router A's buffer to fill, eventually leading to a "spill" (unless, of course, the originating client understands congestion notifications and stops the flow entirely). Eventually, Router C sends a restart message to Router B, but by that time, packets will have already been lost. Figure 2 is the problems with buffering and congestion notification windowing. The most complex and flexible form of flow control, windowing, is perhaps the most commonly used form of flow control today. In windowing, an agreed- upon number of packets are allowed to be transferred before an acknowledgment from the receiver is required. This means that one station should not be able to easily overload another station: it must wait on the remote station to respond before sending more data. In addition to flow control, windowing is also used for error recovery.

Objectives for design to avoid congestion and buffering system:-

1) To design a new rate-based explicit congestion controller (called the IntelRate controller) based on FLC to avoid link bandwidth, the number of flows, packet loss and network latency i.e. "intelligent" controller. 2) To provide max-min fairness to achieve an effective bandwidth allocation and utilization. 3) To generate relatively smooth source throughput, maintain delay and achieve stable jitter performance by controlling the queue size. 4) To demonstrate our controller has a better QoS performance 5) Avoid buffering using by avoiding variation of queuing delay due to the dynamics queue length.

To achieve the above objectives:- 1) the new controller treats the network as a black box in the sense that queue size is the only parameter to adjust the source sending rate e.g. queue size in RED(Random Early Detection) algorithm and API-RCP [5] 2) The controller retains the merits of the existing rate controllers such as XCP and RCP 3) we will employ OPNET modeller to verify the effectiveness and superiority 4) router acts as a data rate regulator by measuring and monitoring the IQSize so we avoid in variation of queuing delay (Queuing jitter) for dynamic queue length which is responsible for buffering.

Inside each router, our distributed traffic controller acts as a data rate regulator by measuring and monitoring the IQSize. As per its application, every host (source**)** requests a sending rate (Req_rate). So value deposited into a dedicated field inside the packet header. Field can be updated by any router. Each router along the data path will compute an allowed source transmission rate according to the IQSize and then compare it with the rate already recorded in Req_rate field. If the former is smaller than the latter, the Req_rate field in the packet header will be updated; otherwise it remains unchanged. The value of the Req_rate field reflects the allowed data rate from congested router. The receiver then sends this value back to the source via an ACK (Acknowledgment) packet, and the source update its current sending rate accordingly. If no router modifies Req_rate field, it means that all routers allow the source to send its data with the requested desired rate. The streaming of encoded video clips is taking an increasing share of bandwidth on the Internet. Video streams are brittle flows in the sense that they are sensitive to packet loss and packet queuing delay.So by maintaining queue size of data rate we can avoid buffering.

1) Every source requests a desired sending rate from then network according to its application.
2) A destination always has enough buffer space to receive data from its source.
3) The propagation delay and the queuing delay along the data path are the two dominant components
4) The queuing discipline of routers is FIFO (First-In-First-Out).

A. Design Parameters:-

a) TBO (Target Buffer Occupancy):- TBO value should be as small as possible. This is especially true under the

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

heavy traffic conditions when the queue is to be stabilized at TBO. Therefore, a bigger TBO will result in longer steady state queuing delay, which is not desirable to some Internet applications such as the real-time video. In short, the TBO chose the network that has a reasonable queuing delay while maintaining good throughput and link utilization. [1]

b) The number N of LVs: - The choice of N has to consider the trade-off between the throughput performance and the computation complexity of the controller. Big N complicates the controller to do more logic computations on choosing the allowed sending rate according to the rules. A small N may lead the controller output to oscillate due to the big partitions of the LVs.

c) The Output Edge Value: - the outermost edge value in the output MFs corresponds to the maximum sending rate that the controller can output. This parameter is chosen to be the maximum value of the Req_rate field among the active incoming flows. [1]

d) The Width Limit : - The parameter m defines the base width of each membership function in the FS. it also affects the extent of overlapping between the adjacent MFs, the parameter m is to have a smaller TBO while remaining the controller output smooth

e) Buffer Size:-The determination of buffer size is closely related to the chosen value of TBO.

B. The Control Procedure:
Additional noise does not significantly deviate. However, beyond 30% of additional noise, the FLC showed significant improvement over the FLC in terms of reduced fluctuation (s.d.) in the sending rate and a reduced packet loss rate, both of which will reflect them in better average video quality. In fact, confirms that delivered average video quality is improved, though, for very high levels of measurement noise, the encoded video stream is so corrupt it matters little which FLCC is in control, the quality is very poor .Note that this procedure actually allows the router to perform the max-min fairness. Sending rate of small flow is smaller than along their data path have no restriction. When the packet arrives at the destination, the receiver extracts Req_rate from the header and records it into the ACK packet before sending it back to the source. Traffic controller acts as a data rate regulator by measuring and monitoring the IQSize .So avoid of buffering using by avoiding variation of queuing delay due to the dynamics queue length .As the video frame is taken from the delivered video stream after decoding, when the video stream was under the control FLCC and the IT2 FLCC respectively. The improvement from employing the IT2 FLCC is self evident. The blocky facts displayed are typically the result of macro block errors.Macroblocks is the units of motion estimation to remove temporal redundancy in compression.

Problem Definition &Linguistic Variable Description Here, we define the problem and will give the description of Linguistic Variables (LV) of the fuzzy system .The Problem Definition Produce a crisp value for the rate of packet flow (r (t)), taking router queue deviation (e (t)) and processing capacity of the router (p (t)) as the two inputs.

C. Linguistic Variables:
1. LV for Queue Deviation (e (t)):
a. Very Small (VS)
b. Small (SS)
c. Medium (MM)
d. Large (LL)
e. Very Large (VL)

2. LV for Processing Speed (p (t)):
a. Low (LL)
b. Average (AA)
c. High (HH)

3. LV for Rate of Packet Flow (r (t)):
a. Minimum (MI)
b. Optimal (OO)
c. Maximum (MX)

D. Fuzzy Set Description:
Here, we define the fuzzy sets used to construct fuzzy values.

Here, for making the design simple, we make following assumptions: Maximum queue size is 3MB (3072 KB), Processing speed of router vary between 150MHz to 800MHz, Maximum Rate of packet flow is 1Mbps.

1. Fuzzy set for Queue Deviation
a. VS: for 0<= e(t) <= 500
b. SS: for 500<= e(t) <= 1024
c. MM: for 1024<= e(t) <= 2048
d. LL: for 2048 <= e(t) <= 2560
e. VL: for 2560 <= e(t) <= 3072

2. Fuzzy set for Processing Speed
a. LL: for 150 <= p(t) <=450
b. AA: for 450 <= p(t) <=650
c. HH: for 600 <= p(t) <= 800

3. Fuzzy set for Rate of packet flow
a. MI: for 0<= r(t) <= 400
b. OO: for 400 <= r(t) <= 900
c. MX: for 800<= r(t) <= 1024

E. Fuzzy Rules:

Following rules are applied on fuzzy input sets:
a. If e(t) is 'VS' OR p(t) is 'LL', then r(t) is MI
b. If e(t) is 'VS' AND p(t) is 'AA', then r(t) is MI
c. If e(t) is 'VS' AND p(t) is 'LL', then r(t) is MI
d. If e(t) is 'SS' AND p(t) is 'HH', then r(t) is MI
e. If e(t) is 'SS' AND p(t) is 'AA', then r(t) is OO
f. If e(t) is 'SS' AND p(t) is 'LL', then r(t) is OO
g. If e(t) is 'MM' AND p(t) is 'HH', then r(t) is OO
h. If e(t) is 'MM' AND p(t) is 'AA', then r(t) is OO
i. If e(t) is 'MM' AND p(t) is 'LL', then r(t) is OO

DOI 10.17148/IJARCCE.2015.44107

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

j. If e(t) is 'LL' AND p(t) is 'HH', then r(t) is OO
k. If e(t) is 'LL' AND p(t) is 'AA', then r(t) is OO
l. If e(t) is 'LL' AND p(t) is 'LL', then r(t) is MX
m. If e(t) is 'VL' AND p(t) is 'HH', then r(t) is MX
n. If e(t) is 'VL' AND p(t) is 'AA', then r(t) is MX
o. If e(t) is 'VL' AND p(t) is 'LL', then r(t) is OO
p. If e(t) is 'VL' OR p(t) is 'HH', then r(t) is MX

V. TOOLS USED

NS2:
The network to be used for evaluation can be simulated using NS2 (Network Simulator 2). NS2 is an open-source event-driven simulator designed specifically for research in computer communication networks. NS2 contains modules for numerous network components such as routing, transport layer protocol, application, etc. To investigate network performance, researchers can simply use an easy-to-use scripting language to configure a network, and observe results generated by NS2.

Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done using NS2. In general, NS2 provides users with a way of specifying such network protocols and simulating their corresponding behaviours. NS2 provides users with executable command ns which take on input argument, the name of a Tcl simulation scripting file.

Users are feeding the name of a Tcl simulation script (which sets up a simulation) as an input argument of an NS2 executable command ns. In most cases, a simulation trace file is created, and is used to plot graph and/or to create animation. NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl).

While the C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events (i.e., a frontend). The C++ and the OTcl are linked together using TclCL.

VI. PERFORMANCE EVALUATION

The single bottleneck network is used to investigate the controller behaviour of the most congested router.

We choose Router 1 as the only bottleneck in the network, whereas Router 2 is configured to have sufficiently high service rate and big buffer B so that congestion never happens there.

The controller is evaluated by the following performance measures.

1) Source throughput: (or source sending rate) bits/second [15]
2) Here, a bit must part of a packet
3) IQSize: length of the bottleneck buffer queue (measured in packets) seen by a departing packet [16]
4) Queuing delay: the waiting time of packet in the router
5) Queuing jitter: Variation of queuing delay due to the dynamics queue length

$$6)\ \text{Link bottleneck utilization} = \frac{\text{Current actual throughput In the bottleneck}}{\text{The maximum data rate of The bottleneck}}$$

$$7)\ \text{Packet loss rate} = \frac{\text{The number of packet dropped}}{\text{The number of total packets received per second by the bottleneck.}}$$

8) Max-min fairness: A feasible allocation of rates is 'max-min fair' if and only if an increase of any rate within the domain of feasible allocations must be at the cost of a decrease of some already smaller or equal rates [14].

VII. CONCLUSIONS

Existing TCP/IP congestion control algorithms cannot efficiently bear new and emerging services needed by the internet community. Intelligent techniques based on Fuzzy Logic for congestion control and avoid buffering are discussed in this paper. As the requirement for the queue management scheme for congestion avoidance is congestion detection. It is hoped that in the continuing paper, an improved networking congestion control approach for TCP using ECN feedback mechanism based on Fuzzy Logic will be implemented.Traffic controller acts as a data rate regulator by measuring and monitoring the IQSize. So avoid of buffering using by avoiding variation of queuing delay due to the dynamics queue length. Intelligent control of network traffic flows has been little explored, though policing of networks that have an access control mechanism has received some attention. However, the streaming of encoded video clips is taking an increasing share of bandwidth on the Internet. Video streams are brittle flows in the sense that they are sensitive to packet loss and packet queuing delay. TCP transport is unsuitable as a means of controlling these flows because it's very reliability results in delay variation unless large buffers are deployed at the receiver. Unfortunately, such buffers are unsuitable for mobile devices because of the energy drain, even if the click-and-stream culture would permit the start-up delay. Therefore, UDP transport with an application layer congestion controller is the normal solution. Though mathematical modelling of TCP at the application layer as a way of preserving its average behaviour has gained ground, this still results in fluctuations in the sending rate and larger packet losses than necessary. Fuzzy logic has been applied to congestion control with satisfactory results.

REFERENCES

[1]. Jungang Liu, Oliver W. W. Yang, "Using Fuzzy Logic Control to Provide Intelligent Traffic Management Service for High-Speed Networks", IEEE transactions on network and service management, vol. 10, no. 2, June 2013
[2]. R. Jain, "Congestion control and traffic management in ATM networks: ecent advances and a survey," Computer Networks ISDN Syst, vol. 28,No. 13, pp. 1723–1738, Oct. 1996.
[3]. V. Jacobson, "Congestion avoidance and control," in Proc. 1988 SIGCOMM, pp. 314–329.
[4]. V. Jacobson, "Modified TCP congestion avoidance algorithm," Apr. 1990.
[5]. K. K. Ramakrishnan and S. Floyd, "Proposals to add explicit congestion Notification (ECN) to IP," RFC 2481, Jan. 1999.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

[6]. D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high Bandwidth-delay product networks," in Proc. 2002 SIGCOMM, pp. 89– 102.
[7]. J. Liu and O. Yang, "Stability analysis and evaluation of the IntelRate Controller for high-speed heterogeneous networks," in Proc. 2011 IEEE ICC, pp. 1–5.
[8]. J. Liu and O. Yang, "Characterization of the IntelRate controller for High-speed networks," in Proc. 2011 Common. Netw. Services Research Conf., pp. 63–68.
[9]. C. Chang and R. Cheng, "Traffic control in an ATM network using Fuzzy set theory," in Proc. 1994 IEEE INFOCOM, vol. 3. pp. 1200–1207.
[10]. A. Pitsillides and A. Sekercioglu. Congestion control. In W. Pedrycz and A. Vasiliakos, editors, Computational Intelligence in Telecommunications Networks, pages 109–158. CRC Press, Boca Raton, FL, September 2000
[11]. K. M. Passino and S. Yurkovich, Fuzzy Control. Addison Wesley Longman Inc., 1998.
[12]. T. W. Vaneck, "Fuzzy guidance controller for an autonomous boat," IEEE Control Syst. Mag., vol. 17, no. 2, pp. 43–51, Apr. 1997.
[13]. T. Kiryu, I. Sasaki, K. Shibai, et al., "Providing appropriate exercise Levels for the elderly," IEEE Eng. Med. Biol. Mag., vol. 20, no. 6, pp. 116–124, 2001.
[14]. M. Welzl, Network Congestion Control: Managing Internet Traffic. John Wiley & Sons Ltd., 2005.
[15]. "OPNET modeler manuals," OPNET version 11.5, OPNET Technologies Inc., 2005.
[16]. D. Gross, J. Shortle, J. Thompson, et al., Fundamentals of Queuing Theory, 4th edition. John Wiley & Sons Inc., 2008.