International Journal of
Advanced Technology and Innovative Research
www.ijatir.org

ISSN 2348–2370
Vol.08,Issue.06,
June-2016,
Pages:1192-1196

Fuzzy Logic Based Congestion Controlling Mechanism for High Speed Networks

G.KRANTHI KUMAR[1], **R.SUBHAN TILAK BASHA**[2]
[1]Programmer, Dept of CSE, JNTUACEK, Kalikiri, AP, India. Email:kranthig89@gmail.com.
[2]Assistant Professor, Dept of ECE, Prayoglabs Pvt Ltd, Hyderabad, Telangana, India, Email:subhan.ece2008@gmail.com.

Abstract: An intuitive based congestion management framework provides routers in taking quick decisions to overcome from massive traffic. Although having many sophisticated protocols to control traffic in networks requires information about the network capabilities which is an overhead task. Hence proposing Fuzzy Logic based controller implementing in router which takes an intelligent decisions to control congestion and managing network traffic by directly evaluating the magnitude of the queue directly of an router without estimating any network parameters, thereby giving max-min fairness little queuing detain. Thus simulation analysis and correlation have showed that our new traffic management scheme can accomplish more than the existing protocols.

Keywords: Intuitive, Protocol, Capabilities, Fuzzy Logic.

I. INTRODUCTION

WSNs are typically used for information gathering in applications like habitat monitoring, military surveillance, agriculture and environmental sensing, and health monitoring. The primary functionality of a WSN is to sense and monitor the state of the physical world. In most cases, they are unable to affect the physical environment. However, in many applications, observing the state of the physical system is not sufficient, it is also expected to respond to the sensed events/data by performing corresponding actions on the system. This stimulates the emergence of wireless sensor/actuator networks (WSANs). Featuring coexistence of sensors and actuators, WSANs enable the application systems to sense, interact, and change the physical world. They can be deployed in lots of applications such as disaster relief, planet exploration, intelligent building, home automation, industrial control, smart spaces, pervasive computing systems, and cyber-physical systems. Real-world WSAN applications have their requirements on the quality of service (QoS). For instance, in a fire handling system built upon a WSAN, sensors need to report the occurrence of a fire to actuators in a timely and reliable fashion; then, the actuators equipped with water sprinklers will react by a certain deadline so that the situation will not become uncontrollable. Both delay in transmitting data from sensors to actuators and packet loss occurring during the course of transmission may potentially deteriorate control performance of the system, and may not be allowed in some situations where the systems are safety-critical. In a smart home, although there is no hard real-time constraint, actuators should turn on the lights in a timely fashion once receiving a report from sensors when someone enters or will enter a room where all lights are off; people would get unsatisfied if kept staying in dark for a long time waiting for lighting.

In practice, QoS requirements differ from one application to another; however, they can be specified in terms of reliability, timeliness, robustness, trustworthiness, and adaptability, among others. Some QoS metrics may be used to measure the degree of satisfaction of these services. Technically, QoS can usually be characterized by, e.g., delay and jitter, packet loss, deadline miss ratio, and/or network utilization (or throughput) in the context of WSANs. As an alternative, a class of explicit congestion control protocols has been proposed to signal network traffic level more precisely by using multiple bits. Examples are the XCP, RCP, JetMax and MaxNet. These protocols have their controllers reside in routers and directly feed link information back to sources so that the link bandwidth could be efficiently utilized with good scalability and stability in high BDP networks. Specifically, JetMax and MaxNet signal network congestion by providing the required fair rate or the maximum link price, and then the final sending rate is decided by sources according to some demand functions or utility functions. XCP feeds back the required increment or decrement of the sending rate, while RCP directly signals sources with the admissible sending rate according to which sources pace their throughput. The advantages of these router-assisted protocols are that 1) they can explicitly signal link traffic levels without maintaining per-flow state, and 2) the sources can converge their sending rates to some social optimum and achieve a certain optimization objective. However, most of these explicit congestion control protocols have to estimate the bottleneck bandwidth in order to compute the allowed source sending rate or link price.

Recent studies show that misestimating of link bandwidth (e.g., in link sharing networks or wireless networks) may easily occur and can cause significant fairness and stability problems. There are some latest protocols on wireless applications such as QFCP (Quick Flow Control Protocol) and the three protocols called Blind, Errors and MAC. They have improved on the estimation error while having high link utilization and fair throughput. However, they still have the fundamental problem of inaccurate estimation resulting in performance degradation. In addition, their bandwidth probing speed may be too slow when the bandwidth jumps a lot. Also, they cannot keep the queue size stable due to oscillations, which in turn affect the stability of their sending rates. There are some explicit protocols that appear to compute the sending rates based solely on the queue size, but in fact they still need to estimate the number of active flows in a router, and this consumes CPU and memory resources. Examples are the rate-based controllers for packet switching networks and the ER (Explicit Rate) allocation algorithmsfor ATM (Asynchronous Transfer Mode) networks. For the API-RCP controller, both the original method (a truncated network model) and the improved method face a memory problem when dealing with many flows (that numbers in millions) arriving to a core router every hour. In some other controllers, the TBO (Target Buffer Occupancy) is designed to be as high as 3 times of the BDP, which can cause large queuing delay and thus degrading network performance, and this becomes even worse in the high-speed networks. Historically, the ER allocation algorithms in ATM networks also share the same problems because they need to evaluate the link bandwidth and/or the numbers of active VCs (Virtual Circuits). Some others adjust the source sending rates in binary-feedback switches or explicit feedback switches according to a few queue thresholds, which may cause unfairness as well as high cell loss rate.

From the perspective of network and service management, the aforementioned congestion control approaches have QoS (Quality of Service) problems in that they cannot guarantee a certain level of performance under some situations due to design drawbacks. There are many different approaches to improve QoS. For example, admission control, as a network traffic management approach, can guarantee QoS by checking the availability of network bandwidth before establishing a connection, Service priority as another approach can be used to improve QoS by providing different service priorities to different users. Pricing or routing policies are also found to address QoS problems. However, they are outside the scope of this paper that focuses on congestion control as an approach to address the QoS management problem. FLC (Fuzzy Logic Control) has been considered for IC (Intelligence Control). It is a methodology used to design robust systems that can contend with the common adverse synthesizing factors such as system nonlinearity, parameter uncertainty, measurement and modeling imprecision. In addition, fuzzy logic theory provides a convenient controller design approach based on expert knowledge which is close to human decision making, and readily helps engineers to model a complicated non-linear system. In fact, fuzzy logic control has been widely applied in industrial process control and showed extraordinary and mature control performance in accuracy, transient response, robustness and stability. FLC has found its applications to network congestion control since 1990. In early stage, it was used to do rate control in ATM network, to guarantee the QoS. These control algorithms are explicit in nature, and they depend on absolute queue length (the maximum buffer size) instead of the TBO to adjust the allowed sending rate. Nevertheless, these early designs have various shortcomings including cell loss (even though cell loss is used as a congestion signal to compute the rate factor, queue size fluctuations, poor network latency, stability and low utilization.

Later, FLC was used in RED (Random Early Detection) algorithm in TCP/IP networks, to reduce packet loss rate and improve utilization. However, they are still providing implicit or imprecise congestion signaling, and therefore cannot overcome the throughput fluctuations and conservative behavior of TCP sources. In light of the above review of different protocols and their shortcomings, we would like to design a distributed traffic management scheme for the current IP (Internet Protocol) networks (and the next generation networks where applicable), in which routers are deployed with explicit ratebased congestion controllers. We would like to integrate the merits of the existing protocols to improve the current explicit traffic congestion control protocols (like XCP, RCP, APIRCP and their enhancements) and form a proactive scheme based on some prudent design ideas such that the performance problems and excessive resource consumption in routers due to estimating the network parameters could beovercome. In this respect, a fuzzy logic controller is quite attractive because of its capability and designing convenience as discussed above Specifically, the objectives of this paper are: 1) to design a new rate-based explicit congestion controller based on FLC to avoid estimating link parameters such as link bandwidth, the number of flows, packet loss and network latency, while remaining stable and robust to network dynamics (Hence, we make this controller “intelligent”). 2) To provide maximum fairness to achieve an effective bandwidth allocation and utilization; 3) to generate relatively smooth source throughput, maintain a reasonable network delay and achieve stable jitter performance by controlling the queue size; 4) to demonstrate our controller has a better QoS performance through case study. To achieve the above objectives, our new scheme pays attention to the following methodologies as well as the merits of the existing protocols.

Firstly, in order to keep the implementation simple, like TCP, the new controller treats the network as a black box in the sense that queue size is the only parameter it relies on to adjust the source sending rate. The adoption of queue size as the unique congestion signal is inspired by the design experience of some previous AQM controllers (e.g., RED and API-RCP) in that queue size can be accurately measured and is able to effectively signal the onset of network congestion as shown in Fig.1. Secondly, the controller retains the merits of the existing rate controllers such as XCP and RCP by providing explicit multi-bit congestion information without having to keep per-flow state information. Thirdly, we rely on the fuzzy logic theory to design our controller to form a traffic management procedure. Finally, we will employ OPNET modeler to verify the effectiveness and superiority of our scheme. The contributions of our work lie in: using fuzzy logic theory to design an explicit rate-based traffic management scheme (called the IntelRate controller) for the high-speed IP networks as shown in Fig.1.

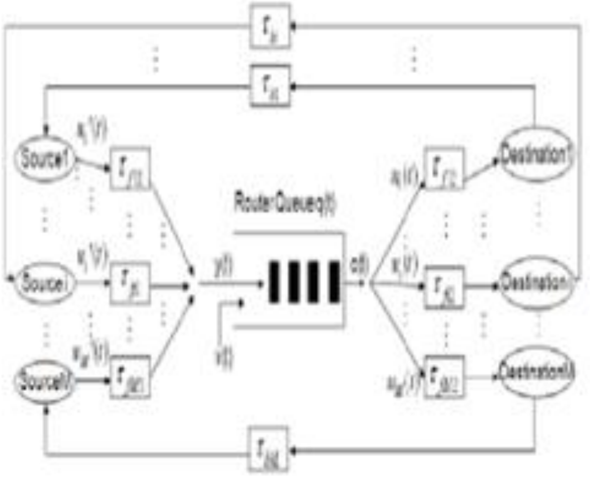

Fig.1. System Model of an AQM Router

The application of such a fuzzy logic controller using less performance parameters while providing better performances than the existing explicit traffic control protocols; 3) The design of a Fuzzy Smoother mechanism that can generate relatively smooth flow throughput; 4) the capability of our algorithm to provide max-min fairness even under large network dynamics that usually render many existing controllers unstable. The rest of the paper is organized as follows. After a description of network model and assumptions in Section II, Section III introduces the design rationale and the controller implementation procedure.

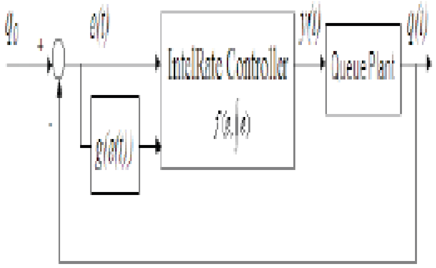

Fig.2. The Intelrate Closed-Loop Control System.

II. DISCUSSION

The good performances above demonstrated by the IntelRate controller also justify the rationale and verify the feasibility of the choices of the design parameters. On one hand, the experiments under different link bandwidths above show that a choice of the TBO value (e.g., which causes a queueing delay less than 10 ms) works well in terms of the throughput performance and queuing delay. On the other hand, the max-min fairness has shown that the IntelRate controller can guarantee the maximum output according to the biggest rate recorded in *req_rate* among all passing flows. This verifies that the controller meets our design objective of choosing the outmost edge value *D* of the output. Besides, our experiments show that the controller can maintain the IQSize at a level much lower than the designed buffer size *B*, even when the network undergoes big dynamics (such as the traffic change or bandwidth variations). This means the Intel Rate controller can significantly save the buffer resources, and thus a reduction in router design cost and size. Perhaps the most notable feature in our design, as mentioned in the beginning, is the utilization of the queue size as the only parameter that the controller needs, which avoids mis-estimation and saves a lot of computational and memory resources that are required in other controllers.

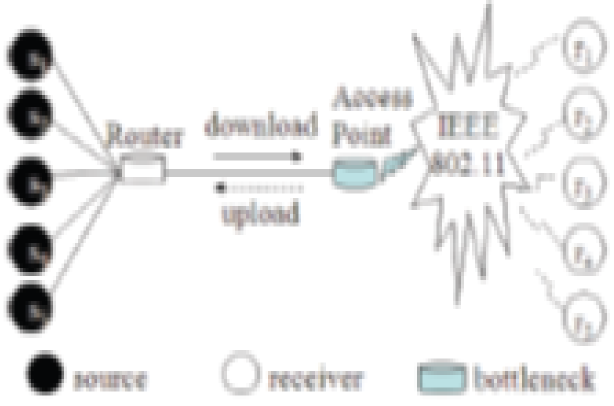

Fig.3. Wireless LAN

We have conducted other in-depth studies including the analysis of the stability, complexity and the characteristics of the IntelRate controller. Space limits would not allow us to produce details here.

III. COMPARISON

Our preliminary results demonstrate that the IntelRate controller is superior to other explicit congestion controllers. For examples, the IntelRate controller has better robustness and link utilization than the XCP upon bandwidthvariations; it has a lower requirement on computational and memory resources than API-RCP while having equivalent and even better performances (the comparisons of the computational intensity and memory requirement with other controllers will be presented in our other papers). Here we pick the QFCP and Blind (we didn't choose ErrorS or MAC from the same paper to do the comparison becauseErrorS is an interchangeable algorithm of Blind and faces the same problem while MAC is too complicated to be put into practice so far) for further comparison as they are enhancements of XCP in reducing the bandwidth estimation errors as discussed before.We use the same IEEE 802.11 wireless LAN (Local AreaNetwork) as done in QFCP and Blind to do the comparison.As shown in Fig. 14, the wireless LAN consists of 5 source destination flows(i.e., $si−ri$, i = 1, 2, ..,5). The band width between *si* and the router is 100Mbps.The backhaul (i.e., the network between the router and the AP) has 1Gbps bandwidth with 100ms propagation delay. The nominal bandwidth ofthe wireless is 11Mbps which needs to be probed by the controllers. In such a network, the wireless interface of the AP (Access Point) is the bottleneck when traffic flows fromthe wired network to the wireless network. The congestion controller resides in the AP to prevent the congestion. We use the Application and Profile module of the OPNET to generate traffic in the sources *si*. In the Application module,we choose video (this module also has other types of traffic available such as ftp, http, audio or email) as the network traffic because, like ftp, it is another type of giant traffic generator nowadays. The starting time, duration and the number of repetitions of the video can be set in the Profile module. The average packet/frame size of the video is 1300 bytes and distributes between [360, 1500] bytes. The TBO of the IntelRate controller is set to 60 packets. To make the experiment more stringent, we make the sourceof each flow greedy by setting its desired sending rate to infinity. The parameters of QFCP and Blind controllers in the simulation are the same as those used in and respectively. The buffer size *B* is set to 600 packets in all the controllers.

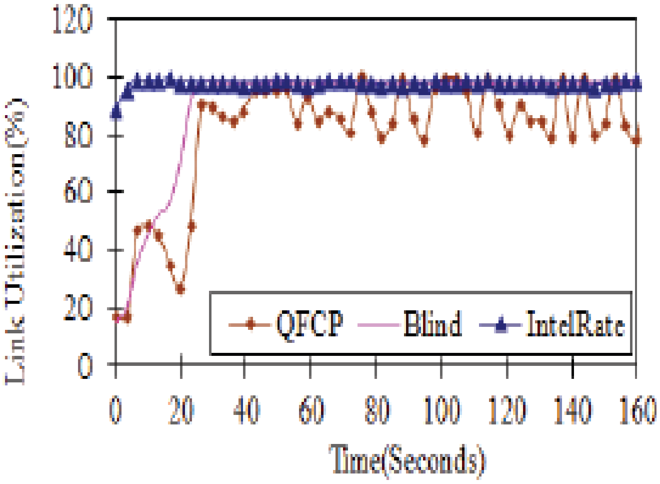

Fig.4. shows the bottleneck utilization which is the ratio between the actual bottleneck throughput and 11Mbps in this case. Since the controllers do not know the bottleneck bandwidth at the beginning, in order to fully utilize the available bandwidth, they need to probe (or say, to estimate)it. As seen, QFCP spends about 20s approaching the 100%utilization (i.e., fully utilizing 11Mbps). There are oscillationsin both its probing stage (when *t* <20s) and steady state (when *t* <20s). The fact that there are still oscillations in itssteady state between 80% and 100% shows that the QFCP has a bandwidth under-estimation problem even though the overestimation issue is addressed by using the router output as the link bandwidth. Blind spends a similar amount of timeon the probing but shows smoother steady-state utilization (overlapping with the IntelRate) than QFCP. In contrast, theIntelRate controller only spends about 3s reaching 100% link bandwidth utilization. Furthermore, it shows the same stable performance as Blind in the steady state.

The reason that QFCP and Blind take a longer probing time is due to the probing process (see Equation (6) in and Equation (18) or (19) in they use to explore the available bandwidth. In comparison, the IntelRate controller aims at building the queue up to the TBO of 60 packets as soon as possible. Once the queue size is built up, the bandwidth is fully utilized. The ability of the Intel Rate controller to stably control the queue size to TBO of 60 packets indicates that the incoming traffic and the link bandwidth of 11Mbps have struck a balance. This is why the Intel Rate controller shows much shorter time reaching the 100% utilization. The throughput of one of the sources in the three controllers is shown in Fig5. In the first 20s, both the QFCP and theBlind gradually increase their sending rates in order to probe the available bandwidth. In their steady state (i.e., *t* >20s), QFCP has more oscillations than Blind due to two reasons: 1) QFCP calculates the sending rate based on the estimated link bandwidth. If the bandwidth is under-estimated from time totime, the sending rate of the source will be decreased accordingly; 2) QFCP has unstable queue size to be illustrated next.In the steady state, their queue sizes are unstable and oscillating wildly all the time. This in turn results in an oscillating RTT which affects the smoothness of their source sending rates. Unlike QFCP and Blind, the IntelRate controller has a closed-loop system dedicated to controlling the queue size. Therefore, the IntelRate controller presents a much more stable queue size after it quickly reaches.

IV. CONCLUSION

A novel traffic management scheme, called the IntelRate controller, has been proposed to manage the Internet congestion in order to assure the quality of service for different service applications. The controller is designed by paying attention to the disadvantages as well as the advantages of the existingcongestion control protocols. As a distributed operation in networks, the IntelRate controller uses the instantaneous queue size alone to effectively throttle the source sending rate with max-min fairness. Unlike the existing explicit traffic control protocols that potentially suffer from performance problems or high router resource consumption due to the estimation ofthe network parameters, the IntelRate controller can overcome those fundamental deficiencies. To verify the effectiveness and superiority of the IntelRate controller, extensive experiments have been conducted in OPNET modeler. In addition to the feature of the FLC being able to intelligently tackle the nonlinearity of the traffic control systems, the success of the IntelRate controller is also attributed to the careful design of the fuzzy logic elements.

V. REFERENCES

[1]M. Welzl, Network Congestion Control: Managing Internet Traffic. JohnWiley& Sons Ltd., 2005.

[2]R. Jain, “Congestion control and traffic management in ATM networks: recent 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.
[6]D. Katabi, M. Handley, and C. Rohrs, “Congestion control for high bandwidth-delay product networks,” in Proc. 2002 SIGCOMM, pp. 89–102.
[7]S. H. Low, F. Paganini, J.Wang, et al., “Dynamics of TCP/AQM and a scalable control,” in Proc. 2002 IEEE INFOCOM, vol. 1, pp. 239–248.
[8]S. Floyd, “High-speed TCP for large congestion windows,” RFC 3649, Dec. 2003.
[9]W. Feng and S. Vanichpun, “Enabling compatibility between TCP Reno and TCP Vegas,” in Proc. 2003 Symp. Applications Internet, pp. 301–308.
[10]M. M. Hassani and R. Berangi, “An analytical model for evaluating utilization of TCP Reno,” in Proc. 2007 Int. Conf. Computer Syst.Technologies, p. 14-1-7.
[11]N. Dukkipati, N. McKeown, and A. G. Fraser, “RCP-AC congestion control to make flows complete quickly in any environment,” in Proc.2006 IEEE INFOCOM, pp. 1–5.
[12]Y. Zhang, D. Leonard, and D. Loguinov, “JetMax: scalable max-min congestion control for high-speedheterogeneous networks,” in Proc.2006 IEEE INFOCOM, pp. 1–13.

Author’s Profile:

G.Kranthi Kumar received M.C.A (Post Graduation) Degree from Rayalaseema Institute of Information and Management Sciences, Tirupati. He is currently pursuing M.Tech Degree in Software Engineering specialization in Karnataka State Open University, Mysore, India. His interests include Software Engineering, Computer Programming, Telecommunication Engineering, Information Science, Robotics, Computer Networks & Wireless Sensor Networks and Database Management Systems. He is a member of IEEE – Hyderabad Section. He is a member of IEEE Communication society.

R. Subhan Tilak Basha received B.Tech Degree from Intell Engineering College, Anantapuram. He Received M.Tech Degree in Embedded Systems specialization in Annamacharya Institute of Technology & Sciences, Rajampet, Andhra Pradesh, India. His interests include Embedded Systems, Communication Systems, Bio-medical Engineering, Computer Networks & Wireless Sensor Networks. He presented many papers in National Level Technical Symposium and Presented Papers in National Conferences. He Presented Papers in International Conferences and Published Research Papers in International Reputed Journals. He is a Senior member of IEEE – Hyderabad Section. He is a member of Professional societies like IAEST, IAENG, IACSIT and SDIWC.