IPHV7I3014X
International Journal Of Advanced Research and Innovation -Vol.7, Issue .III
ISSN Online: 2319 – 9253
Print: 2319 – 9245

CONGESTION CONTROL AND TRAFFIC MANAGEMENT IN HIGH SPEED NETWORKS

[#1] G.Gnana Prasuna -M.Tech Pursuing,
[#2]Shaikh Karimullah Basha -Associate Professor,
[#3]B. Venkata Krishna- Assistant Professor
Department of Computer Science & Engineering,
CMR COLLEGE OF ENGINEERING AND TECHNOLOGY, Hyd, TS, India.

Abstract: As the increase in use of computing devices such as computers, tablets and smart phones there is a huge demand for the fast-growing internet traffic. Distributed traffic management frame work has been proposed, in which routers are deployed with intelligent data rate controllers to tackle the high traffic level. The traffic control protocol is unique as other traffic control protocols have to estimate network parameters which involves link latency, bottleneck bandwidth, packet loss rate, or the number of flows in order to compute the allowed source sending rate. The fuzzy-logic based controller can measure queue size directly; it neglects various potential performance issues arising due to parameter estimations we reduce much consumption of computation and memory resource in routers. As a network parameter, the queue size can be monitored accurately and used for making proactive decision if action should be taken to regulate the source sending rate, and thus increases the resilience of the network to traffic congestion. By the fuzzy logic technique, QoS (Quality of Service) in communication is assured by good performances of our scheme such as max-min fairness, low queuing delay and good robustness to network dynamics. The conclusion is that the results and comparisons have verified the effectiveness and made a created a new benchmark that our traffic management scheme using fuzzy-logic can achieve better performance than the existing protocols that depend entirely on the estimation of network parameter.

***Keywords:* Congestion control, fuzzy logic control, quality of service, max-min fairness, robustness, traffic management.**

I.INTRODUCTION

Traffic congestion control is one of the effective approaches to manage the network traffic [1], [2]. Network traffic management can prevent a network from severe congestion and degradation in throughput delay performance. Unlike other explicit congestion controllers that depend on the estimation of network parameters (such as link latency, bottleneck bandwidth, packet loss, or the number of flows) to compute the allowed source sending rate, the Intel Rate controller can avoid this while maintaining good stability and robustness. Simulation results have verified these performances. Our study shows that the Intel Rate controller can be approximated by a PI (Proportional-Integral) controller but with time-varying gains, which allows the controller to outperform its counterparts. Finally, by comparing with the API-RCP (Adaptive PI Rate Control Protocol) using Opnet simulation, we experimentally illustrate our conclusion [3].The back-pressure algorithm is a well-known throughput-optimal algorithm. However, its delay performance may be quite poor even when the traffic load is not close to network capacity due to the following two reasons. First, each node has to maintain a separate queue for each commodity in the network, and only one queue is served at a time. Second, the back-pressure routing algorithm may route some packets along very long routes. The user presents solutions to address both of the above issues, and hence, improve the delay performance of the back-pressure algorithm. One of the suggested solutions also decreases the complexity of the queuing data structures to be maintained at each node [4]. Recently, explicit Control Protocol (XCP), Rate Control Protocol (RCP), and Adaptive Proportional-Integral Rate Control Protocol (API-RCP) have been proposed for congestion control, with the main objective of achieving fair and maximum bandwidth utilization. However, studies reveal that both RCP and XCP may suffer continuous oscillations due to misestimating the bottleneck link capacity, and API-RCP may experience oscillations because of its PI adaptively scheme which involves switching nonlinearity. To avoid these problems, in the paper a way of designing congestion control based on the instantaneous queue sizes in the routers is proposed. The new scheme attains high link utilization and smooth dynamics by clamping the bottleneck queue at a desired size. And it maintains good fairness by allocating the bottleneck bandwidth equally to the competing flows. Simulations are performed to verify the effectiveness of the theoretical design [5].Next generation IP-based networks will offer quality of service (QoS) guarantees by deploying technologies such as differentiated services (DiffServ) and multi-protocol label switching (MPLS) for traffic engineering and network-wide resource management. Despite the progress already made, a number of issues still exist regarding edge-to-edge intra-domain and inter-domain QoS provisioning and management. This tutorial will start by providing background on technologies such as DiffServ, MPLS and their potential combination for QoS support. It will subsequently introduce trends in service level agreements (SLAs) and service level specifications (SLSs) for the subscription to QoS-based services It will then move to examine architectures and frameworks for the management and control of QoS-enabled networks, including the following aspects: approaches and algorithms for off-line traffic engineering and provisioning through explicit MPLS paths or through hop-by-hop IP routing; approaches for dynamic resource management to deal with traffic fluctuations outside the predicted envelope; a service management framework supporting a "resource provisioning cycle"; the derivation of expected traffic demand from subscribed SLSs and approaches for SLS invocation admission control; a monitoring architecture for scalable information collection supporting traffic engineering and service management; and realization issues given the current state-of-the-art of management protocols and monitoring support. The tutorial will also include coverage of emerging work towards inter-domain QoS provisioning, including aspects such as: an inter-domain business model; customer and peer provider SLSs; architecture for the management and control of inter-domain services; inter-domain off-line traffic engineering; and QoS extensions to BGP for dynamic traffic engineering. Relevant industrial activities such as IP sphere will be also covered. In all these areas, recent research- - work will be presented, with pointers to bibliography and a specially tailored Web page with additional resources [6].A new active queue management scheme, fuzzy explicit marking (FEM), implemented within the differentiated services (Diffserv) framework to provide the congestion control using a fuzzy logic control approach. Network congestion control remains a critical and high priority issue. The rapid growth of the Internet and increased demand to use the Internet for time-sensitive voice and video applications necessitate the design and utilization of effective congestion control algorithms, especially for new architectures, such as Diffserv. As a result, a number of researchers are now looking at alternatively schemes to TCP congestion control. RED (random early detection) and its variants are one of these alternatives to provide QoS in TCP/IP Diffserv networks. The proposed fuzzy logic approach for congestion control allows the use of linguistic knowledge to capture the dynamics of nonlinear probability marking functions and offer effective implementation, use of multiple inputs to capture the state of the network more accurately, enable finer tuning for packet marking behaviors for aggravated flows, and thus provide better QoS to different types of data streams, such as TCP/FTP traffic, whilst maintaining high utilization [7].

II. RELATED WORKS

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, e.g., [8]. Service priority as another approach can be used to improve QoS by providing different service priorities to different users, e.g., [9]. Pricing or routing policies are also found to address QoS problems e.g., [10]. 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) [11] 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 [12]. 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.

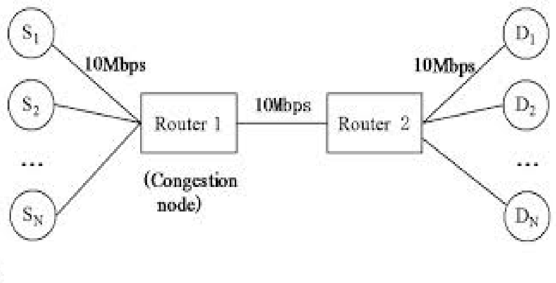


Fig.1 Basic Configuration of a fuzzy logic system

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 [13], [14]. 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, e.g., [15]), queue size fluctuations, poor network latency, stability and low utilization. Later, FLC was used in RED (Random Early Detection) algorithm in TCP/IP networks, e.g., [16], 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. 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 be overcome. 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 ; 2) to provide maxmin 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. 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.

III. PRINCIPLE FOR TRAFFIC MANAGEMENT

We consider a backbone network interconnected by a number of geographically distributed routers, in which hosts are attached to the access routers which cooperate with the core routers to enable end-to-end communications. Congestion occurs when many flows traverse a router and because it's IQSize (Instantaneous Queue Size) to exceed the buffer capacity, thus making it a bottleneck in the Internet. Since any router may become bottleneck along an end-to-end data path, we would like each router to be able to manage its traffic. Below is the general operation principle of our new traffic management/control algorithm. 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 it desires by depositing a value into a dedicated field *Req_rate* inside the packet header. This field can be updated by any router en route. Specifically, 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. After the packet arrives at the destination, the value of the *Req_rate* field reflects the allowed data rate from the most congested router along the path if the value is not more than the desired rate of the source. The receiver then sends this value back to the source via an ACK (ACKnowledgment) packet, and the source would update its current sending rate accordingly. If no router modifies *Req_rate* field, it means that all routers en route allow the source to send its data with the requested desired rate. The following assumptions for the remainder of this paper pertain.1) Every source requests a desired sending rate from the network according to its application. 2) A destination always has enough buffer space to receive data from its source. This is because we do not want the destination to impose any constraint on the source sending rate when we verify the effect of our new control scheme in a bottlenecked router.3) The propagation delay and the queuing delay along the data path are the two dominant components of the RTT while other components like processing delay of a packet in routers or hosts are negligible in comparison. 4) The queuing discipline of routers is FIFO (First-In-First-Out). 5) Long-lived flows with infinitely long files are used to approximate the greedy behavior of a source when active. This would generate the severest traffic in order for us to verify the robustness of the new scheme.

IV. THE INTELRATE CONTROLLER DESIGN

Figure 2 depicts the components of our fuzzy logic traffic controller for controlling traffic in the network system defined in Fig. 1. Called the IntelRate, it is a TISO (Two-Input Single- Output) controller. The TBO (Target Buffer Occupancy) $q0 > 0$ is the queue size level we aim to achieve upon congestion. The queue deviation $e(t) = q0 - q(t)$ is one of the two inputs of the controller. In order to remove the steady state error, we choose the integration of $e(t)$ as the other input of the controller, i.e. $g(e(t)) = \_\_\ e(t)\,dt$. The aggregate output is $y(t) = ui(t - \tau i)$. Under heavy traffic situations, the IntelRate controller would compute an allowed sending rate $ui(t)$ for flow i according to the current IQSize so that $q(t)$ can be stabilized around $q0$. In our design, IQSize $q(t)$ is the only parameter each router needs to measure in order to complete the closed-loop control.

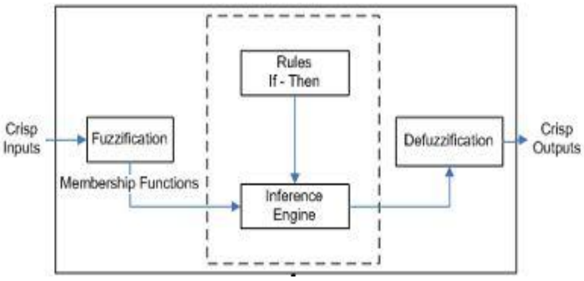
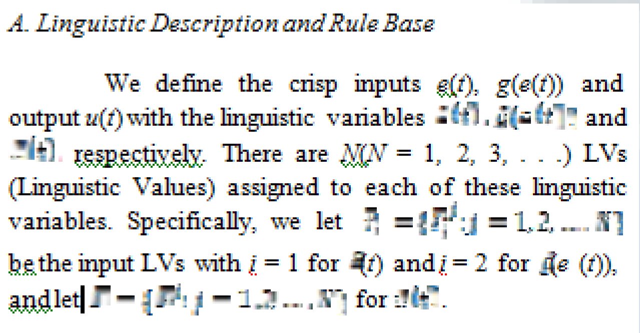


Fig.2 Intelrate Controller System

FLC is a non-linear mapping of inputs into outputs, which consists of four steps, i.e., rule base building, fuzzification, inference and defuzzification. The concepts of fuzzy set and logic of FLC were introduced in 1965 by Zadeh, and it was basically extended from two-valued logic to the continuous interval by adding the intermediate values between absolute TRUE and FALSE. Interested readers are referred to some standard tutorials/texts like [11], [17] for the details of the fuzzy logic theory. In the sequel, we formulate our new controller by following those four steps along with designing the fuzzy linguistic descriptions and the membership functions. The parameter design issues and the traffic control procedure are also discussed at the end of the section.



A. Linguistic Description and Rule Base

We define the crisp inputs $e(t)$, $g(e(t))$ and output $u(t)$ with the linguistic variables [illegible], [illegible] and [illegible] respectively. There are $N(N = 1, 2, 3, . . .)$ LVs (Linguistic Values) assigned to each of these linguistic variables. Specifically, we let [illegible] $= \{$[illegible]$: j = 1, 2, \ldots,$ [illegible]$\}$ be the input LVs with $i = 1$ for [illegible](t) and $i = 2$ for [illegible]$(e(t))$, and let [illegible] $= \{$[illegible]$: j = 1, 2, \ldots,$ [illegible]$\}$ for [illegible].

B. Membership Function, Fuzzification and Reference

Our IntelRate controller employs the isosceles triangular and trapezoid-like functions as its MFs (Membership Functions).Figure 3 describes the MFs used to determine the certainty of a crisp input or output.

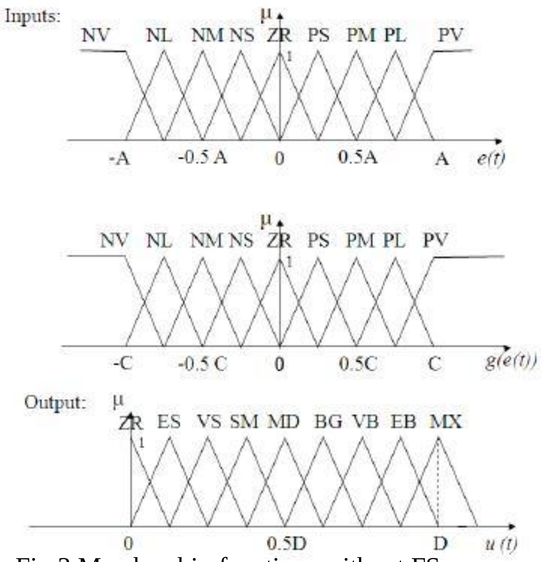


Fig.3 Membership functions without FS.

Note that Fig. 3 describes the usual way of designing MFs with FLC to determine the certainty of a crisp input or output, where $e(t)$ and $g(e(t))$ have the same number of MFs and there are no boundaries for the input $g(e(t))$. While Fig. 4 is used to illustrate the design of a general FS, the designated values actually come from an example of $N = 9$ LVs with the absolute values of both the upper and lower limits of $g(e(t))$ set to $mq0$. Since $e(t)$ is bounded by the physical size of a queue, we have the boundaries according to the limits $q0-B \leq e(t) \leq q0$. The vertical dashed lines in Fig. 4 denote those boundaries of inputs or output.

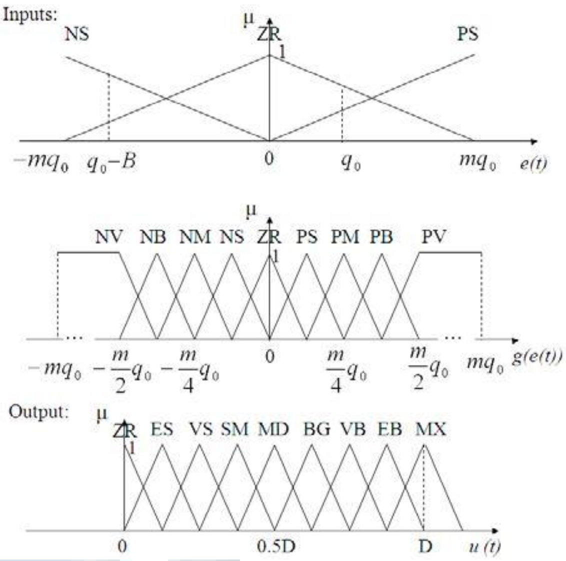


Fig.4 Membership functions with FS.

C. Defuzzification

For the defuzzification algorithm, the IntelRate controller applies the COG (Center of Gravity) method to obtain the crisp output.

D. Design Parameters

From our design above, one can see there are different parameters which ultimately will affect the performance of our traffic controller.

E. The Control Procedure

The traffic-handling procedure of the IntelRate controller in a router is , (1) Upon the arrival of a packet, the router extracts *Req_rate* from the congestion header of the packet. (2) Sample IQSize $q(t)$ and update $e(t)$ and $g(e(t))$. (3) Compute the output $u(t)$ and compare it with *Req_rate*. (4) If an operation cycle d is over, update the crisp output $u(t)$ and the output edge value of D. Note that this procedure actually allows the router to perform the max-min fairness in that the greedy flows are always restricted to $u(t)$ by a router under heavy traffic conditions while those small flows whose desiredsending rate are smaller than u(t) along their data path have no such a restriction.

V. PERFORMANCE EVALUATION

The capability of the Intel Rate controller is demonstrated by performance evaluations through a series of experiments.

A. Simulated Network

The controller is evaluated by the following performance measures.

1) Source throughput (or source sending rate) is

defined to be the average number of bits successfully sent out by a source per second, i.e. bits/second. Here, a bit is considered to be successfully sent out if it is part of a packet that has been successfully sent. 2) IQ Size is the length of the bottleneck buffer queue seen by a departing packet. 3) Queuing delay is the waiting time of a packet in the router queue before its service.4) Queuing jitter is the variation of queuing delay due to the queue length dynamics, and is defined as the variance of the queuing delay.5) Link (or bottleneck) utilization is the ratio between the current actual throughput in the bottleneck and the maximum data rate of the bottleneck. It is expressed as a fraction less than one or as a percentage.6) Packet loss rate is the ratio between the number of packet dropped and the number of total packets received per second by the bottleneck.7) A feasible allocation of rates is ‘maxminfair’ 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

B. Robustness to Large Network Charges

The real world Internet traffic is always dynamic. The performance of our controllers when faced with drastic network changes as in Fig.5 the variations of the number of flows or the available bandwidth is been investigated.

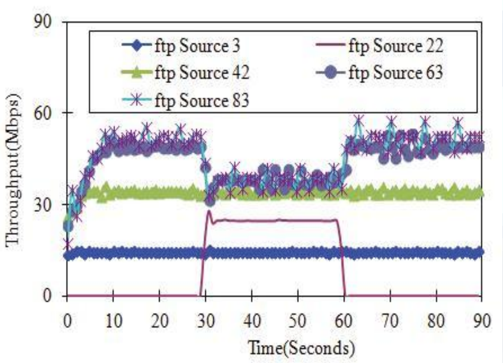


Fig. 5 Source and IQ Size dynamics under traffic change.

C. Queuing Jitter Control

One main source of the network latency oscillations comes from the dynamics of queuing delay in the routers. In Fig.6, we want to check under heavy traffic condition show the queuing delay fluctuates under the different TBO sin the Intel Rate controller, and how big the queuing jitters can be.

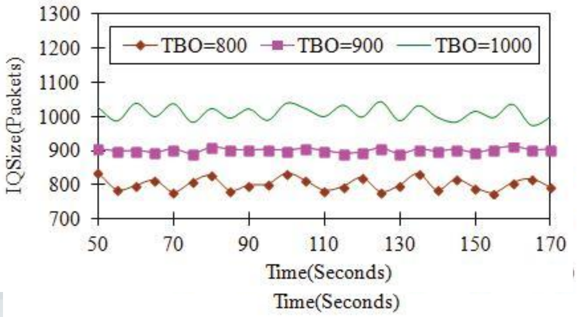


Fig.6 IQSize and queuing delay under different TBOs.

E. Effect of Short-Lived Traffic

The experiment shows that the long-lived flow scans accommodate the large short-lived http flows upon their arrivals. They simply regard the http flows as the long-lived ftp flows

F. Utilization and Packet Loss Rate

The utilization and packet loss rate performance of the Intel Rate controller with respect to bottleneck bandwidth or the different settings of TBOs. First we check the system utilization and packet loss rate under the different bottleneck bandwidth from 45Mbpsto 10Gbps. The simulation results show that the Intel Rate controller is able to maintain the ideal zero packet loss rate with 100% link utilization despite the different bottlenecks(here we omit the plots due to space limit). Therefore, the buffer never overflows and packets are never lost upon heavy traffic. In the meanwhile, the stable feature in IQ Size and throughput guarantees the full bandwidth utilization.

VI. CONCLUSIONS

A novel traffic management scheme, called the Intel Rate 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 existing congestion control protocols. As a distributed operation in networks, the Intel Rate controller uses the instantaneous queue size alone to effectively throttle the source sending rate with max-min fairness. The back-pressure algorithm, while being throughput-optimal, is not useful in practice for adaptive routing since the delay performance can be really bad. In the paper, we have presented an algorithm that routes packets on shortest hops when possible and decouples routing and scheduling using a probabilistic splitting algorithm built on the concept of shadow. Probabilistic routing table that hinges slowly over time, real packets do not have to explore long paths to improve throughput; this functionality is performed by the shadow “packet. A packet-by-packet wavelength-routing interconnect technique for switching system with three-stage architecture has been demonstrated. The technique uses an optical wavelength division multiplexing link and dynamic bandwidth-sharing among wavelengths.

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] J. Liu and O. Yang, "Characterization of the IntelRate controller for high-speed networks," in Proc. 2011 Commun. Netw. Services Research Conf., pp. 63–68.

[4] Bui, L. ; Srikant, R. ; Stolyar, A.," Novel Architectures and Algorithms for Delay Reduction in Back-Pressure Scheduling and Routing ," INFOCOM 2009 *Commun. Netw. Services Research Conf.*, pp. 33

[5] W. Hu and G. Xiao, "Design of congestion control based on instantaneous queue size in the routers," in *Proc. 2009 IEEE GLOBECOM*, pp. 1–6.

[6] G. Pavlou, "Traffic engineering and quality of service management for IP-based NGNs," in *Proc. 2006 IEEE/IFIP Netw. Operations Manage. Symp.*, p. 589..

[7] C. Chrysostomou, A. Pitsillides, G. Hadjipollas, *et al.*, "Fuzzy explicit marking for congestion control in differentiated services networks," in *Proc. 2003 IEEE Int. Symp. Computers Commun.*, vol. 1, pp. 312–319.

[8] A. Vashist, M. Siun-Chuon, A. Poylisher, *et al.*, "Leveraging social network for predicting demand and estimating available resources for communication network management," in *Proc. 2011 IEEE/IFIP Int.Symp. Integrated Netw. Manage.*, pp. 547–554.

[9] D. Toelle and R. Knorr, "Congestion control for carrier ethernet using network potential," *Proc. 2006 IEEE/IFIP Netw. Operations Manage. Symp.*, pp.1–4.

[10] M. Charalambides, P. Flegkas, G. Pavlou, *et al.*, "Policy conflict analysis for diffserv quality of service management," *IEEE Trans. Netw. Service Manage.*, vol. 6, no. 1, pp. 15–30, Mar. 2009.

[11] G. Kesidis, "Congestion control alternatives for residential broadband access," *Proc. 2010 IEEE Netw. Operations Manage. Symp.*, pp. 874– 877.

[12] K. M. Passino and S. Yurkovich, *Fuzzy Control*. Addison Wesley Longman Inc., 1998.

[13] E. Jammeh, M. Fleury, C. Wagner, *et al.*, "Interval type-2 fuzzy logic congestion control for video streaming across IP networks," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 5, pp. 1123–1142, 2009.

[14] T. W. Vaneck, "Fuzzy guidance controller for an autonomous boat," *IEEE Control Syst. Mag.*, vol. 17, no. 2, pp. 43–51, Apr. 1997. [15] 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.

[16] R. Chang and C. Cheng, "Design of fuzzy traffic controller for ATM networks," *IEEE/ACM Trans. Netw.*, vol. 4, no. 3, pp. 460–469, June 1996.

[17] H. Aoul, A. Nafaa, D. Negru, and A. Mehaoua, "FAFC: fast adaptive fuzzy AQM controller for TCP/IP networks," in *Proc. 2004 IEEE GLOBECOM*, vol. 3, pp. 1319–1323.

[18]S.Kaehler.Available: http://www.seattlerobotics.org/encoder/mar98/fuz/flindex .html