Dynamic Traffic Management Services To Provide High Performance In IntelRate Controller Using Fuzzy Logic

M.Vasuki, N.Balkis, V.Jayalakshmi
Department of MCA, Sri Manakula Vinayagar Engineering College. Puducherry, India

ABSTRACT
Traffic is the chief puzzle problem in which every country faces to elaborate sending a number of packets throughout the world. This paper proposes a new speculation for distributed traffic management by availing the presumption of fuzzy logic. The routers are established by using an IntelRate Controllers to manage the traffic congestion in the networks dynamically. Fuzzy logic is used to previse the maximum allowable sending rate by observing the queue size of router. The network traffic control protocol is unique to estimate the network parameter which involves link latency, bottleneck bandwidth or packet loss rate 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 estimation as reduce consumption of computation and memory resource in router. A network parameter, the queue size can be viewed accurately and if action should be taken to regulate the source sending rate and it increases the resilience of the network to traffic congestion. Using the fuzzy logic technique, QoS (Quality of Service) can achieve better performance than the existing protocol that depends on the estimation of network parameter, to make the network more adaptive for current traffic conditions.

Keywords
congestion control, fuzzy logic control, QoS (Quality of Service), max-min fairness, robustness, traffic management.

1. Introduction

Traffic congestion control is one of the effective approaches to manage the network traffic [1], [2]. Network traffic management can be used to prevent and control a network traffic from congestion and degradation in throughput delay performance. An explicit congestion controller depend on estimation of network parameters (such as link latency, bottleneck bandwidth, packet loss, or the number of flows) which is used to compute the allowed source sending rate and it will measured accurately. The Intel Rate controller can be used to avoid the good stability and robustness in the network traffic. 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 and finally by comparing with API-RCP (Adaptive PI Rate Control Protocol) by using OPNET simulation[3].

The back-pressure algorithm is a well-known throughput-optimal algorithm and its delay performance may be quite poor. When the traffic load is not close to network capacity with the following two reasons. First, each node has been maintained as a separate queue in the network, and only one queue is distributed at a time. Second, the back-pressure routing algorithm may send some packets along very long routes. The user introduce some solving solutions to address both the above issues, and hence, make better improvement in the delay performance of the back-pressure algorithm. One of the recommended solutions also reduce 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 target to achieving the fair and maximum bandwidth utilization. However, it shows 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, one way of designing the congestion control based on theIQSize (Instantaneous Queue Sizes) in the routers. The new scheme achieves high link utilization and smooth dynamics by fixing the bottleneck queue at a desired size. It maintains good fairness by allocating the bottleneck bandwidth equally to the competing flows. these are all performed to verify the effectiveness of the theoretical design of the network traffic [5].Next generation IP-based networks will has guaranteed of quality of service (QoS) by deploying technologies such as differentiated services (DiffServ) and multi-protocol label switching (MPLS) for traffic engineering and network-wide resource management.

A number of issues still exist regarding edge-to-edge intra-domain and inter-domain QoS provisioning and management for the subscription to QoS-based services. It will then move to examine the architectures and the 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 big deal with traffic fluctuations outside an

Manuscript received November 5, 2015
Manuscript revised November 20, 2015

envelope; a service management framework supporting a "resource provisioning cycle"; the derivation of expected traffic request from subscribed SLSs and approaches for SLS invocation admission control; observing 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. [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. 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 it also gives the effective implementation. By using the multiple inputs to capture the network traffic more accurately, by enabling the elegant tuning for packet marking manners for aggravated flows, and thus provide better QoS to different types of data streams, such as TCP/FTP traffic, it maintains high utilization [7].These has been widely examined with various proposed solutions such as AQM(Active Queue Management) schemes [8]-[10] whose control protocols are also implicit in nature.

A class of explicit congestion protocols has been proposed to signal network traffic level by using the multiple bits. Examples are the XCP [6], RCP [11], JetMax [12], MaxNet [13].The explicit protocol is used to compute the sending rates based on the queue size, to estimate the number of active flows in a router and consumes memory resources and others. for examples are the rate-based controllers [14]-[16] for packet switching networks and also ER (Explicit Rate) allocation algorithm [17] for ATM (Asynchronous Transfer Mode) networks. The API-RCP controller [15], both the original and the improved method [18] faces some memory problem when dealing with many flows arriving to a router each and every hour [19]. In some other controllers (eg., [17]), the TBO (Target Buffer Occupancy) is designed as high as 3 times of the BDP, which can cause a large queuing delay and degrading network performance, and it becomes even in the high-speed networks.

The ER allocation algorithms in ATM networks will also shares the same problems (e.g., [17], [20]) because they just need to compute the link bandwidth and the numbers of active VCs (Virtual Circuits). Some others (e.g., [21]) adjust the source sending rates in binary-feedback switches according to a queue thresholds, which may cause un fairness as well as high cell loss rate [2], [22].

A. Existing system:

In existing 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 mentioned 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., [23]- [25] Service priority as another approach can be used to improve QoS by providing different service priorities to different users. This paper focuses on congestion control as an approach to address the QoS (Quality of Service) problems in that they cannot guarantee a certain level of performance under some situations due to design drawbacks large queuing delay and thus degrading network performance.

B. Proposed system:

FLC (Fuzzy Logic Control) [26] 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 [27]. 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 [28], [29] .Fuzzy Smoother mechanism that can generate relatively smooth flow throughput. Finally, we will employ OPNET modeler to verify the effectiveness and superiority of our scheme. FLC has found its applications to network congestion control since 1990. In early stage, it was used to do rate control in ATM network, e.g., [30], [31], to guarantee the QoS. 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. Section IV and Section V illustrate the simulation performance of the IntelRate controller. Finally, we conclude this paper with Section VI. For the remainder of the paper, the following notations and symbols pertain.

A	Edge value of MFs (Membership Functions) of $e\ (t)$, beyond which the MFs of $e\ (t)$ saturate
B	Buffer capacity
$c\ (t)$	Service rate (output link capacity) of a router
C	Edge value of MFs of $g(e(t))$, beyond which the MFs of $g(e(t))$ saturate
D	Outermost edge value of MFs of $u\ (t)$
$e(t)$	Queue error which is one input of the IntelRate controller
$g(e(t))$	Integration of $e\ (t)$ which is the other input of the IntelRate controller
m	Multiple of TBO to design the width limit for the MFs of input $e(t)$ and $g(e(t))$
N	Number of LVs (Linguistic Values)
q_0	TBO of a router
$q(t)$	IQSize (Instantaneous Queue Size) of a router
$u(t)$	The controller crisp output for each flow
$u'(t)$	Current source sending rate
$v(t)$	Aggregate uncontrolled incoming traffic rate to a router
$y(t)$	Aggregate controlled incoming traffic rate to a router (also aggregate controller output)
$\mu_{Pi}{}^{j}$	Input fuzzy set of the IntelRate controller
$\mu_{U}{}^{j}$	Output fuzzy set of the IntelRate controller
T_{fi1}	Time delay of a packet from source I to a router
T_{fi2}	Time delay of a packet from a router to its destination i
T_{bi}	Feedback delay of a packet from destination I back to source i
T_{pi}	RTPD (Round Trip Propagation Delay)
T_i	RTT (Round Trip Time)

2. Literature Survey

Literature survey is the documentation of a comprehensive review of the published and unpublished work from secondary sources data in the areas of specific interest to the researcher. The library is a rich storage base for secondary data and researchers used to spend several weeks and sometimes months going through books, journals, newspapers, magazines, conference proceedings, doctoral dissertations, master's theses, government publications and financial reports to find information on their research topic. With computerized databases now readily available and accessible the literature search is much speedier and easier and can be done without entering the portals of a library building.

The researcher could start the literature survey even as the information from the unstructured and structured interviews is being gathered. Reviewing the literature on the topic area at this time helps the researcher to focus further interviews more meaningfully on certain aspects found to be important is the published studies even if these had not surfaced during the earlier questioning.

So the literature survey is important for gathering the secondary data for the research which might be proved very helpful in the research. The literature survey can be conducted for several reasons. The literature review can be in any area of the business.

In this paper, we referred the “Congestion avoidance and control” was developed in the year 1988 by the author V. Jacobson to avoid the congestion in the network and followed by “Modified TCP congestion avoidance algorithm” was developed in the year 1990 by the author V. Jacobson.“Feedback control of congestion in packet switching networks: the case of a single congested node” was developed in the year 1993 by the author L. Benmohamed and S. M. Meerkov.” Enhanced PRCA (proportional rate control algorithm)” was developed in the year 1994 by the author L. Roberts.The fuzzy logic traffic controller is proposed to perform traffic and functions by using, “Design of fuzzy traffic controller for ATM Networks” was developed in the year 1996 by the author R. Chang and C. Cheng.“Self-similarity in world wide web traffic: evidence and possible causes” was developed in the year 1997 by the author M. E. Crovella and A. Bestavros, this will shows evidence that WWW traffic exhibits is consistent with self-similar traffic model. “Fuzzy Control” was developed in the year 1998 by the author K. M. Passino and S. Yurkovich.“Proposals to add explicit congestion notification (ECN) to IP” was developed in the year 1999 by the author K. K. Ramakrishnan and S. Floyd.“A neural-fuzzy system for congestion control in ATM networks”, was developed in the year 2000 by the author S. J. Lee and C. L. Hou, the neural-fuzzy scheme for rate-based feedback congestion control in ATM.“Providing appropriate exercise levels for the elderly” was developed in the year 2001 by the author T. Kiryu, I. Sasaki, K. Shibai, et al.“Passive estimation of round-trip times” was developed in the year 2002 by the author H. Jiang and C. Dovrolis.“Dynamics of TCP/AQM

and a scalable control" was developed in the year 2003 by the author S. H. Low, F. Paganini, J.Wang,et al."Network Congestion Control: Managing Internet Traffic" was developed in the year 2004 by the author M. Welzl."Design of adaptive PI rate controller for best effort traffic in the Internet based on phase margin" was developed in the year 2007 by the author Y. Hong and O. Yang."A game-theoretic model for capacity-constrained fair bandwidth allocation" was developed in the year 2008 by the author Y. Yan, A. El-Atawy, and E. Al-Shaer.
The policy management will provides the ability to (re-)configure the differentiated services networks so that will desired QoS goals can be achieved with the help of "Policy conflict analysis for diffuser quality of service management" was developed in the year 2009 by the author M. Charalambides, P. Flegkas."Explicit congestion control (XCC) algorithms for time varying capacity media" was developed in the year 2011 by the author F. Abrantes, J. Araujo, and M. Ricardo.

3. BACKGROUND DETAILS

A. Traffic management principles and modelling

Consider a backbone network interconnected by a number of geographically distributed routers; the hosts are attached to the access routers which cooperate with the core routers to enable end-to-end communications. Congestion control protocol occurs when many flows traverse a router and because it's IQSize (Instantaneous Queue Size) to exceed the buffer capacity and make a bottleneck in the Internet. Any router may become bottleneck along an end-to-end data path. Each router can be able to manage its traffic. Below is the general operation principle of our new traffic 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 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. Each router has the data path and it allows you to compute the 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.

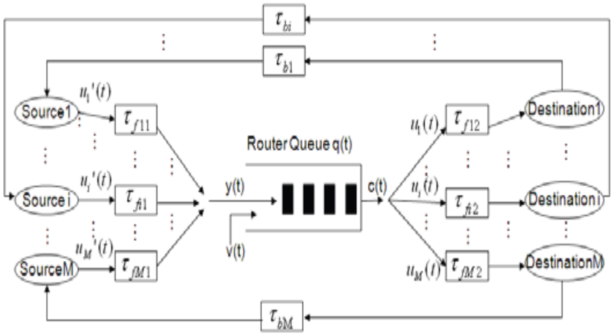

Fig. 1. System model of an AQM router

Some of the assumptions are used in this paper to pertain:
• Every source requests a desired sending rate from the network according to its application.
• 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.
• 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.
• The queuing discipline of routers is FIFO (First-In-First-Out).
• 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.

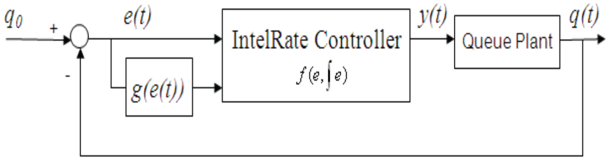

Fig. 2. The IntelRate closed-loop control system.

B. Traffic attributes

To set a connection on ATM network, user can specify the following parameters. It relates the input traffic characteristics and desired Quality of Service.
Peak Cell Rate (PCR): It can be used to find the maximum instantaneous rate at which the user will transmit.
Sustain Cell Rate (SCR): The average rate can be measured over a long interval.
Cell Loss Ratio (CLR): The percentages of cells are lost in the network due the error and congestion. So it cannot be delivered to the destination.

$$\text{Cell Loss Ratio}=\frac{\text{Lost Cells}}{\text{Transmitted Cells}}$$

Each ATM cells has a “Cell Loss Priority (CLP)” bit in the header. During congestion, first drop the network cells that have CLP bit set. Since the loss of CLP=0 cell is more harmful to the operation of the application. CLR can be specified separately for cells with CLP=1 and for those with CLP=0.

Cell Transfer Delay (CTD): the experienced delay cells are used between the network entry and exit points is called cell transfer delay. It includes queuing delay propagation delay in various intermediate switches; queuing points are used at service time.

Cell Delay Variation (CDV): It can be used to measure the variance of CTD. High variation implies lager buffering for delay sensitive traffic such voice and video. Multiple ways can be used to measure CDV. “peak-to-peak” are used, in this CDV consist of computing the difference between the (1-α)-percentile and the minimum of the cell transfer delay for some small value of α.

Cell Delay Variation Tolerance (CDVT) and Burst Tolerance (BT): source transmitting the Intel cell time.

Maximum Burst Size (MBS): Maximum number of back-to- back cell are used to send at the peak cell rate without violating the sustained cell rate. It related to the PCR, SCR and BT as follows:

$$\text{Burst Tolerance} = (\text{MBS}-1)\left(\frac{1}{\text{SCR}} - \frac{1}{\text{PCR}}\right)$$

MBS is more intuitive than BT, MBS used signaling messages. BT can be easily calculated from MBR, SCR, and PCR.

Minimum Cell Rate (MCR): The minimum rate desired by a user.

The first six of the above network parameters were specified in UNI version3.0. MCR are recently added and it will appear in the next version of the traffic management document.

C. Service categories

There are five categories of services. The QoS parameters for these categories are summarized in Table 1 and are explained below:

Constant Bit Rate (CBR): It can be used for emulating circuit switching. The cell rate is constant. Cell loss ratio is specified for CLP=0 cells and may or may not be specified for CLP=1 cells. Examples: CBR are telephone, video conferencing, and television applications are used.

Variable Bit Rate (VBR): It allows users to send the variable rate. Statistical multiplexing may be used as small nonzero random loss. Depending upon whether or not the application is sensitive to cell delay variation and it can be subdivided into two categories: Real time VBR and Non-real time VBR. In non-real time VBR, only mean delay is specified. In real time VBR, maximum delay and peak-to-peak CDV are specified; an example of real time VBR is interactive compressed video while that of non-real time VBR is multimedia email.

Available Bit Rate (ABR): It can be used design the normal data traffic such as file transfer and email. It does not require the standard cell transfer delay and cell loss ratio to be minimized; it’s desirable for switches to minimize the delay and loss as much as possible. It depend upon the congestion state of the network, the source is required to control its rate. The users are allowed to declare a minimum cell rate, which is guaranteed to the VC by the network. Most VCs will ask for an MCR of zero with higher MCR may be denied connection if sufficient bandwidth is not available.

Unspecified Bit Rate (UBR): It can be used to design the data applications. We can use any left-over capacity and it should not be sensitive to cell loss or delay. Connections are not rejected on the basis of bandwidth shortage and not policed for their usage behavior. During congestion, the cells are lost but the sources are not expected to reduce their cell rate.

D. Fuzzy logic

It’s a form of multi-valued logic; it deals with reasoning that is approximate rather than fixed and exact. Compared to traditional binary sets (where variables may take on true), fuzzy logic variables may have a truth value that ranges in degree between 0 and 1.Fuzzy logic has been extended to handle the concept of partial truth, where the truth value may range between completely true and completely false. When linguistic variables are used, these degrees may be managed by specific functions. Irrationality can be described in terms of what is known as the fuzzy objective. The term “fuzzy Logic" was introduced with the 1965 proposal of fuzzy set theory by Lotfi A. Zadeh. Fuzzy logic has been applied to many fields, from control theory to artificial intelligence. Fuzzy logic had, however, been studied since the 1920s, as infinite-valued logic - notably by Łukasiewicz and Tarski.

Some of the 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
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

IV. The Intelrate Controller Design

Figure 2 contains the components of our fuzzy logic traffic controller for controlling traffic in the network system defined in Fig. 1. Called the Intel Rate, it is a TISO (Two-Input Single-Output) controller. The TBO (Target Buffer Occupancy) q0. 0 is the queue size level and it can be used to achieve the congestion control protocol. The queue deviation e (t) = q0 −q(t) is one of the two inputs of the controller. To remove the steady state error, we can 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 − τi). In heavy traffic situations, the IntelRate controller would compute the sending rate ui(t) for flow i according to the current IQSize. q(t) can be stabilized around q0. IQSize q(t) is the only parameter that can be used in the each router.

It measures the order to complete the closed –loop control.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. 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 based

We define the crisp inputs e(t), g(e(t)) and output u(t) with the linguistic variables $\vec{e}(t)$, $\vec{g}$ (e (t)) and $\vec{u}$ (t), respectively. There aren N(N = 1, 2, 3, . . .) LVs (Linguistic Values) assigned to each of these linguistic variables. Specifically, we let Pi ={$\overline{\text{p}}$i j : j= 1, 2,…,N} be the input LVs with i = 1 for $\vec{e}(t)$ and i = 2 for $\vec{g}$(e (t)), and let $\overline{U}$ ={$\overline{U}$ j: j = 1, 2, … ,N} for $\vec{u}$ (t). For example, when N = 9, we can assign the following values for both the inputs e(t) and g(e(t)). $\overline{P}$i1 =“Negative Very Large (NV),” $\overline{P}$i2i =“Negative Large (NL),” $\overline{P}$3i =“Negative Medium (NM),” $\overline{P}$4ii =“Negative Small (NS),” $\overline{P}$5i =“Zero (ZR),” $\overline{P}$6i =“Positive Small (PS),” $\overline{P}$7i =“Positive Medium (PM),” $\overline{P}$8i =“Positive Large (PL),”and $\overline{P}$9i =“Positive Very Large (PV),” i = 1, 2. Similarly, we can designate the output when N = 9 with the following linguistic values. U1=“Zero (ZR),” U2=“Extremely Small (ES),” U3=“Very Small (VS),” U4=“Small (SM),”U5=“Medium (MD),” U6=“Big (BG),” U7=“Very Big (VB),” U8=“Extremely Big (EB),” and U9=“Maximum (MX).”

Table I illustrates the controller rule base using N = 9. The rule base is the set of linguistic rules used to map the inputs to the output using the “If. . . Then. . . ” format, e.g. “If e(t) is ZR (Zero) and g(e(t)) is PS (Positive Small), Then u(t) is BG (Big).” In the following sections, we refer to a rule in this table by the notation ($\overline{P}^{j}_{1}$, $\overline{P}^{k}_{2,}$ $\overline{U}^{l}$), where j, k, l = 1, 2. . . N, e.g. ($\overline{P}^{5}_{1}$, $\overline{P}^{2}_{2,}$ $\overline{U}^{2}$) = (ZR, NL, ES).

TABLE I Rule Table For Intelrate Controller (9 Lvs)

Allowed Throughput *u(t)*		*e(t)*								
		NV	NL	NM	NS	ZR	PS	PM	PL	PV
g(e(t))	NV	ZR	ZR	ZR	ZR	ZR	ES	VS	SM	MD
	NL	ZR	ZR	ZR	ZR	ES	VS	SM	MD	BG
	NM	ZR	ZR	ZR	ES	VS	SM	MD	BG	VB
	NS	ZR	ZR	ES	VS	SM	MD	BG	VB	EB
	ZR	ZR	ES	VS	SM	MD	BG	VB	EB	MX
	PS	ES	VS	SM	MD	BG	VB	EB	MX	MX
	PM	VS	SM	MD	BG	VB	EB	MX	MX	MX
	PL	SM	MD	BG	VB	EB	MX	MX	MX	MX
	PV	MD	BG	VB	EB	MX	MX	MX	MX	MX

B. Membership function, Fuzzification and reference

Intel Rate 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.

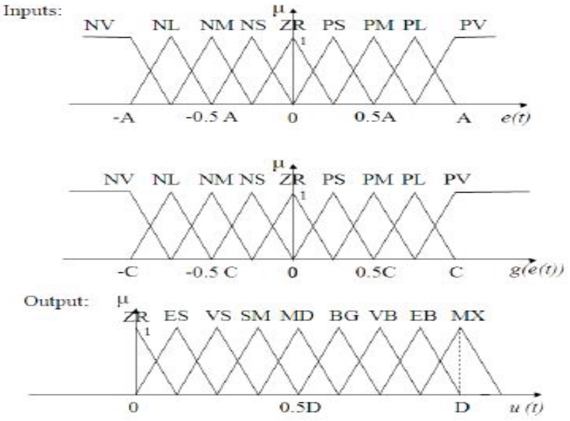

Fig.3 Membership functions without FS.

The above figure 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 is no boundaries for the input g(e(t)). While Fig. 4 is used to illustrate the design of a general FS, the designated values are 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.

In this e(t) is bounded by the physical size of a queue, we have the boundaries according to the limits q0−B ≤ e(t) ≤q0. The vertical dashed lines in Fig. 4 denote those boundaries of inputs or output.

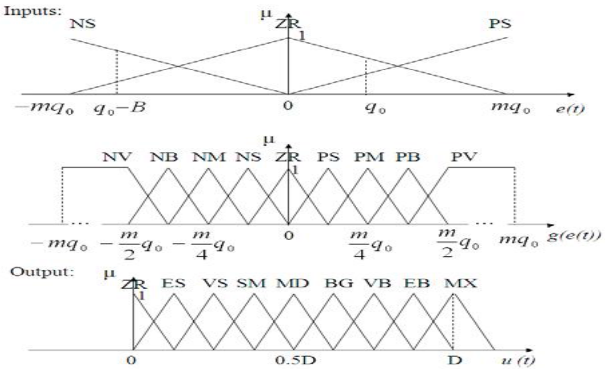

Fig.4 Membership functions with FS.

C. Defuzzification

Using the defuzzification algorithm, the IntelRate controller applies the COG (Center of Gravity) method to obtain the crisp output with the equation [26].

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,

- The router extracts Req_rate from the congestion header of the packet.
- Sample IQSize q(t) and update e(t) and g(e(t)).
- Compute the output u(t) and compare it with Req_rate.
- If an operation cycle d is over, update the crisp output u(t) and the output edge value of D.

This procedure 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 is used.

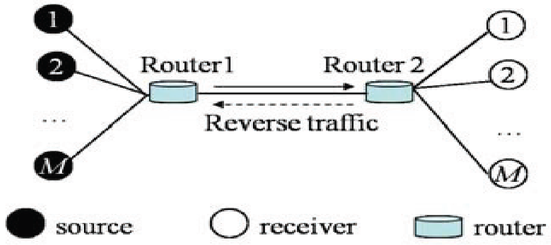

Fig. 5. Simulation network.

The desired sending rate is smaller than u(t) along their data path have no such a 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.

V. An Algorithm Implementation Of Allocation And Reduce Packet Loss Rate And Utilization

ER Allocation Algorithm(Explicit-Rate)

In an explicit rate (ER) allocation algorithm for the MAX-MIN flow control of elastic traffic services with minimum rate guarantee in the setting of the ATM ABR (Available Bit Rate) service. The proposed ER algorithm is simple in that the number of operations required to compute it at a switch is minimized, scalable in that per-VC (virtual circuit) operations including per-VC queuing, per-VC accounting and per-VC state management are virtually removed, and stable in that by employing it the user transmission rates and the network queues are asymptotically stabilized at a unique equilibrium point at which MAX-MIN fairness with minimum rate guarantee and target queue lengths are achieved respectively. To improve the speed of convergence we normalize the controller gains of the algorithm by the estimate of the number of locally-bottlenecked VCs. The estimation scheme is also computationally simple and scalable since it does not require per-VC accounting either. We analyze the theoretical performance of the proposed algorithm and verify its agreement with the practical performance through simulations in the case of multiple bottleneck nodes. We believe that the proposed algorithm will serve as an encouraging solution to the MAX-MIN flow control of elastic traffic services, the deployment of which has been debated long due to their lack of theoretical foundation and implementation complexity.ER allocation algorithm is as follows.

$$r[k + 1] = r[k] -\sum_{i=0} \alpha_i (q[k - i] - q_T) -\sum_{j=0} \beta_j r[k - j]$$

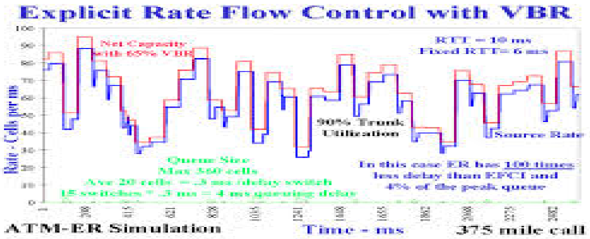

Fig. 7.ER Allocation Algorithm

VI. Performance Evaluation

In this Intel Rate controller capacity demonstrate the performance evaluations through a series of experiments.

A. Simulated Network

Each and every controller is evaluated by the following performance measures.

- Source throughput is defined the average number of bits successfully sent out by a source per second, i.e. bits/second. Consider the bit is successfully sent out if it is part of a packet that has been successfully sent.
- IQ Size is the length of the bottleneck buffer queue seen by a departing packet.
- Queuing delay is the waiting time of a packet in the router queue before its service.
- Queuing jitter is the variation of queuing delay due to the queue length dynamics, and it can be defined as the variance of the queuing delay.
- Link 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.
- Packet loss rate is the ratio between the number of packet dropped and the number of total packets received per second by the bottleneck.
- 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.

B. Robustness to Large Network Charges

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

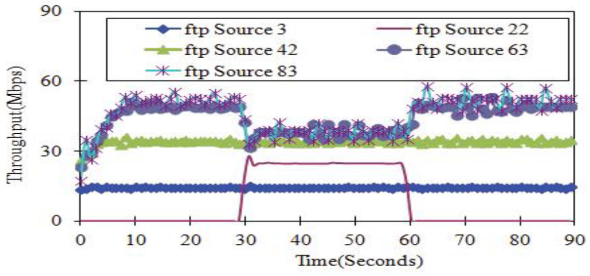

Fig. 8 Source and IQ Size dynamics under traffic change.

C. Queuing Jitter Control

One main source of the network latency oscillations comes from the dynamics of queuing delay in the routers. In Fig.8, 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 used.

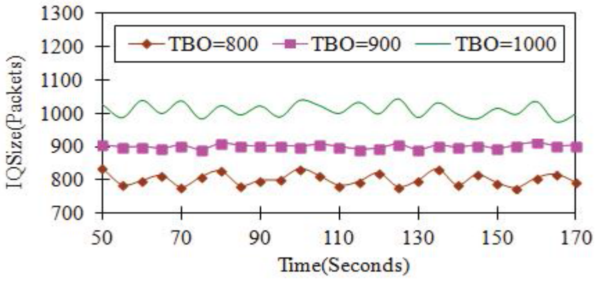

Fig.9 IQSize and queuing delay under different TBOs.

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 bottleneck. In this 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.

VIII. conclusion

The IntelRate Controller has managed the internet traffic in order to reduce the QoS for different application. The IQSize has used for effective buffer capacity and max-min fairness is used to monitor the url to avoid the congestion with high performance speed dynamically. The IntelRate controller can overcome some fundamental deficiency such as potential performance problem or high router resource consumption and memory resource in router. The intelligent Fuzzy Logic Control (FLC) will tackle the non-linearity of the traffic control system.

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] 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-speed heterogeneous networks," in Proc. 2006 IEEE INFOCOM, pp. 1–13.

[13] B. Wydrowski, L. Andrew, and M. Zukerman, "MaxNet: a congestion control architecture for scalable networks," IEEE Commun. Lett., vol. 7, no. 10, pp. 511–513, Oct. 2003.

[14] L. Benmohamed and S. M. Meerkov, "Feedback control of congestion in packet switching networks: the case of a single congested node," IEEE/ACM Trans. Netw., vol. 1, no. 6, pp. 693–708, Dec. 1993.

[15] Y. Hong and O. Yang, "Design of adaptive PI rate controller for besteffort traffic in the Internet based on phase margin," IEEE Trans. Parallel Distrib. Syst., vol. 18, no. 4, pp. 550–561, 2007.

[16] 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.

[17] S. Chong, S. Lee, and S. Kang, "A simple, scalable, and stable explicit rate allocation algorithm for max-min flow control with minimum rate guarantee," IEEE/ACM Trans. Netw., vol. 9, no. 3, pp. 322–335, June 2001.

[18] Y. Hong and O. Yang, "An API-RCP design using pole placement technique," in Proc. 2011 IEEE ICC, pp. 1–5.

[19] B. Ribeiro, T. Ye, and D. Towsley, "Resource-minimalist flow size histogram estimator," in Proc. 2008 ACM SIGCOMM Conf. Internet Measurement, pp. 285–290.

[20] Y. H. Long, T. K. Ho, and A. B. Rad, "An enhanced explicit rate algorithm for ABR traffic control in ATM networks," Int. J. Commun. Syst, vol. 14, pp. 909–923, 2011.

[21] L. Roberts, "Enhanced PRCA proportional rate control algorithm," AFTM- R, Aug. 1994.

[22] S. J. Lee and C. L. Hou, "A neural-fuzzy system for congestion control in ATM networks," IEEE Trans. Syst. Man Cybern. B., Cybern., vol. 30, no. 1, pp. 2–9, 2000.

[23] 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.

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

[25] Y. Yan, A. El-Atawy, and E. Al-Shaer, "A game-theoretic model for capacity-constrained fair bandwidth allocation," Int. J. Netw. Manage., vol. 18, no. 6, pp. 485–504, Nov. 2008.

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

[27] 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.

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

[29] 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.

[30] 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.

[31] J. Harju and K. Pulakka, "Optimization of the performance of a ratebased congestion control system by using fuzzy controllers," in Proc. 1999 IEEE IPCCC, pp. 192–198.