



## International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2014

# Fuzzy Logic Control to Offer Intellectual Transfer Management Service & Progressive Packet Arrivals for High-Speed Networks

B.Surendra<sup>1</sup>, T.Vivekanandan<sup>2</sup> Dr.M.Giri<sup>3</sup>

<sup>1</sup>Student, Dept of Computer Science & Engineering, Sreenivasa Institute of Technology And Management Studies,  
Chittoor, India

<sup>2</sup>Associate Professor, Dept of Computer Science & Engineering, Sreenivasa Institute of Technology And Management  
Studies, Chittoor, India

<sup>3</sup>HOD, Professor , Dept of Computer Science & Engineering, Sreenivasa Institute of Technology And Management Studies,  
Chittoor, India

**ABSTRACT:** In particular, we determine the conditions for optimality in terms of probability of successful delivery and mean delay and we devise optimal policies, so-called piecewise-threshold policies. We account for linear block-codes and rate less random linear coding to efficiently generate redundancy, as well as for an energy constraint in the optimization. We numerically assess the higher efficiency of piecewise-threshold policies compared with other policies by developing heuristic optimization of the thresholds for all flavors of coding considered. A dispersed traffic management framework, in which routers are deployed with intelligent data rate controllers to tackle the traffic mass. Unlike other explicit traffic control protocols that have to estimate network parameters (e.g., link latency, bottleneck bandwidth, packet loss rate, or the number of flows) in order to total the allowed source sending rate, our fuzzy-logic-based manager can measure the router queue size directly; hence it avoids various potential performance problems arising from parameter estimations while reducing much consumption of computation and memory resources in routers. As a network parameter, the queue size can be accurately monitored and used to proactively decide if action should be taken to regulate the source sending rate, thus increasing the resilience of the network to traffic congestion. The statement QoS (Quality of Service) is assured by the good performances of our scheme such as max-min fairness, low queueing delay and good robustness to network dynamics. Simulation results and comparisons have verified the effectiveness and showed that our new traffic management scheme can achieve better performances than the existing protocols that rely on the estimation of network parameters.

### I. INTRODUCTION

In the basic scenario, the source has initially all the packets. Under this assumption it was shown in that the transmission policy has a threshold structure: it is optimal to use all opportunities to spread packets till some time  $\sigma$  depending on the energy constraint, and then stop. This policy resembles the well-known "Spray-and-Wait" policy. In this work we assume a more general arrival process of packets: they need not to be simultaneously available for transmission initially, i.e., when forwarding starts, as assumed in This is the case when large multimedia files are recorded at the source node (from, e.g., a cellular base station) that sends them out (in a DTN fashion) without waiting for the whole file reception.

**Contributions:** This paper focuses on general packet arrivals at the source and two-hop routing. We distinguish two cases: when the source can overwrite its own packets in the relay nodes, and when it cannot. The contributions are fourfold:

- For work-conserving policies (i.e., the source sends systematically before stopping completely), we derive the conditions for optimality in terms of probability of successful delivery and mean delay.



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2014

- In the case of non-overwriting, we prove that the best policies, in terms of delivery probability, are piecewise threshold. For the overwriting case, work-conserving policies are the best without energy constraint, but are outperformed by piecewise-threshold policies when there is an energy constraint.
- We extend the above analysis to the case where copies are coded packets, generated both with linear block codes and rateless coding. We also account for an energy constraint in the optimization.
- We illustrate numerically, in the non-overwriting case, the higher efficiency of piecewise-threshold policies compared with work-conserving policies by developing a heuristic optimization of the thresholds for all flavors of coding considered. As well, in the overwriting case, we show that work-conserving policies are the best without any energy constraint.

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 jamming signaling, and therefore cannot overcome the throughput fluctuations and conservative behavior of TCP sources. In light of the above review of different protocol 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 rate based 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 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 (Hence, we make this controller “intelligent”); 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 signals 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. The contributions of our work lie in: 1) using fuzzy logic theory to design an explicit rate-based traffic management scheme (called the Intel Rate controller) for the high-speed IP networks; 2) 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.

## II. BEYOND WC POLICIES

We have obtained the structure of the best WC policies, and identified cases in which these are globally optimal. We show the limitation of WC policies.

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2014

## A. The case $K=2$

1) *The non-overwriting case:* Consider two packets, arriving at the source at  $t_1$  and  $t_2$ , respectively. Consider the policy  $\mu(s)$  where  $0 = t_1 < s \leq t_2$  which transmits packet 1 during  $[t_1, s)$ , does not transmit anything during  $[s, t_2)$  and then transmits packet 2 after  $t_2$ . Let us define  $X(t) = 1 - \exp(-\beta t)$ .

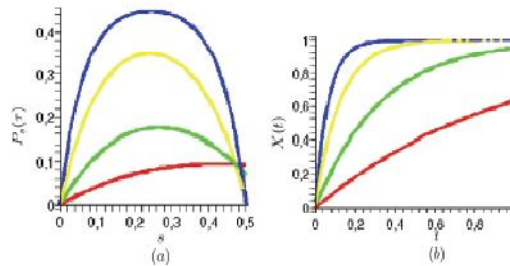


Fig. 1.  $K = 2$ ,  $\beta = 1, 3, 8, 15$ ,  $t_1 = 0$ ,  $t_2 = 0.8$ . (a) Success probability under non WC policy  $u(s)$  as a function of  $s$ ; top curve corresponds to largest value of  $\beta$ ; second top corresponds to second largest  $\beta$  etc. (this order changes only at  $s$  very close to 0.5). (b) The evolution of  $X(t)$  as a function of  $t$  under the best WC policy. The curves are ordered according to  $\beta$  with the top curve corresponding to the largest  $\beta$  etc.

Then it holds

$$X_1(t) = \begin{cases} X(t) & 0 \leq t \leq s \\ X(s) & s \leq t \leq \tau \end{cases}$$

$$X_2(t) = \begin{cases} 0 & 0 \leq t \leq t_2 \\ X(t - (t_2 - s)) - X(s) = \frac{e^{-\beta s} - e^{-\beta(t - (t_2 - s))}}{e^{-\beta s} - e^{-\beta(t - (t_2 - s))}} & t_2 \leq t \leq \tau \end{cases}$$

This gives

$$\int_0^\tau X_1(t) dt = \frac{-1 + \beta s + e^{-\beta s}}{\beta} + (\tau - s)(1 - e^{-\beta s})$$

$$\int_0^\tau X_2(t) dt = \frac{e^{-\beta s}}{\beta} (\beta(\tau - t_2) - 1 + e^{-\beta(\tau - t_2)})$$

### Using the above dynamics, we can illustrate

The improvement that non WC policies can bring. We took  $\tau = 1$ ,  $t_1 = 0$ ,  $t_2 = 0.8$ . We vary  $s$  between 0 and  $t_2$  and compute the probability of successful delivery for  $\beta = 1, 3, 8, 15$ . The probability of successful delivery under the piecewise-threshold policies  $u(s)$  are depicted in Figure 1 as a function of  $s$  which is varied between 0 and  $t_2$ . The corresponding optimal policies  $u(s)$  are given by the thresholds  $s = 0.425, 0.265, 0.242, 0.242$  for the  $\beta$  above, respectively.

In all these examples, there is no optimal policy among those that are WC (i.e., with  $s = t_2$ ). We can verify that a WC policy is optimal for all  $\beta \leq 0.9925$ . Note that under any WC policy, as  $\beta$  increases to infinity,  $X_1(t_2)$  and hence  $X(t_2)$  increase to one. Thus  $\int_0^\tau X_2(t) dt$  tends to zero. We conclude that the success delivery probability tends to zero, uniformly under any WC policy.

2) *The overwriting case:* WC policies are optimal for P2 and P1 provided that no energy constraint is enforced. WC policies are not necessarily optimal anymore with an energy constraint (e.g., if  $X_e$  is lower than  $X_1(t_2)$ ).

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2014

## B. Time changes and policy improvement

**Lemma 4.1:** Let  $p < 1$  be some positive constant. For any multi-policy  $\mathbf{u}(t) = (u_1(t), \dots, u_n(t))$  satisfying  $u_i(t) = u_i(t) \leq p$  for all  $t$ , define the policy  $\mathbf{v}(t) = (v_1(t), \dots, v_n(t))$  where  $v_i(t) = u_i(t/p)$  or equivalently,  $u_i(t) = p v_i(tp)$ ,  $i = 1, \dots, n$ . Define by  $X_i(t)$  the state trajectories under  $\mathbf{u}(t)$ , and let  $\bar{X}_i(t)$  be the state trajectories under  $\mathbf{v}(t)$ . Then  $\bar{X}_i(t) = X_i(tp)$ , and hence  $X(t) = X(tp)$ .

**Proof.** We look for  $s(t)$  such that  $X_i(t) = X_i(s(t))$ . In the non-overwriting and overwriting cases, respectively:

$$\frac{d\bar{X}_i(s(t))}{dt} = \frac{d\bar{X}_i(s)}{ds} \frac{ds(t)}{dt} = [\beta v_i(s)(1 - \bar{X}(s))] \frac{ds(t)}{dt},$$

$$\frac{d\bar{X}_i(s(t))}{dt} = \frac{d\bar{X}_i(s)}{ds} \frac{ds(t)}{dt} = [\beta v(s)(1 - \bar{X}(s))] \frac{ds(t)}{dt}.$$

Taking  $s(t) = pt$ , we end up with the desired result for all  $i = 1, \dots, K$ .

An acceleration  $\mathbf{v}(t)$  of  $\mathbf{u}(t)$  from a given time  $t_-$  is defined as  $v_i(t) = u_i(t)$  for  $t \leq t_-$  and  $v_i(t) = u_i(t_- + (t - t_-)/p)/p$  otherwise, for all  $i = 1, \dots, n$ . We now introduce a policy Improvement procedure.

**Definition 5.1:** Consider some policy  $\mathbf{u}(t)$  such that  $0 < u_i(t) \leq p$  for some  $0 < p < 1$  for all  $t$  in some interval  $S = [a, b]$  and that

$$\int_b^c u_i(t) dt > 0 \quad \text{for some } c > b. \text{ Let } \mathbf{w}(t) \text{ be the policy obtained from } \mathbf{u}(t) \text{ by}$$

- (i) accelerating it by a factor of  $1/p$  between instants  $a$  and  $d := a + p(b - a)$ ,
- (ii) from time  $d$  till time  $e := c - (1 - p)(b - a)$ , use  $\mathbf{w}(t) = \mathbf{u}(t + b - d)$ . Then use  $\mathbf{w}(t) = 0$  till time  $c$ .

**Lemma 5.2:** There exists some policy improvements of  $\mathbf{u}(t)$  by  $\mathbf{w}(t)$  such that  $P_s(\tau = c, \mathbf{w}) > P_s(\tau = c, \mathbf{u})$ .

**Proof.** Let  $\mathbf{X}(t)$  and  $\bar{\mathbf{X}}(t)$  be the stat processes under  $\mathbf{u}(t)$  and  $\mathbf{w}(t)$ , respectively.

**For non-overwriting policies:** Consider  $\mathbf{w}(t)$  obtained by the improvement of Def. 4.1. We can easily show that

$$Z_i(c) := \int_a^c X_i(t) dt \leq Z'_i(c) := \int_a^c \bar{X}_i(t) dt \quad \text{for all } i, \text{ with strict inequality for some } i. \text{ Owing to the expression of } P_s(c, \mathbf{u}) = \prod_{i=1}^K \zeta(Z_i(c)) \text{ given in Section II.A, we end up with the lemma for non-overwriting policies.}$$

**For overwriting policies:** From a policy  $\mathbf{u}(t)$  with the features of Def. 5.1, consider a policy  $\mathbf{w}(t)$  obtained by substituting  $\mathbf{w}(t)$  by  $w(t) = \sum_{i=1}^K w_i(t)$  in Def. 5.1. Then, reasoning with  $X(t)$  instead of  $X_i(t)$ , we can show in the same way as above

that  $Z_{-total}(c) > Z_{total}(c)$ ,  $X(t) \leq X_{-}(t)$  and  $X(t) < X_{-}(t)$  for some  $t \in [a, c]$ . Having  $\mathbf{u}(t)$ ,  $\mathbf{w}(t)$ ,  $X(t)$  and  $X_{-}(t)$  fixed, we can choose  $\mathbf{w}(t)$  such that  $X_{-1}(t) = X_1(t) + X_{-}(t) - X(t)$  and  $X_{-i}(t) = X_i(t)$  for  $t = 2, \dots, K$  (it is sufficient to express  $X_i(t)$  as a

function of  $v_i(t)$  and  $v(t)$  from the ODE  $\frac{d\bar{X}_i(t)}{dt} = \beta(v_i(t) - v(t)\bar{X}_i(t))$  and then express  $v_i$  by equalizing  $X_i(t)$  to the desired function). Finally we get  $Z_{-1}(t) > Z_1(t)$  and  $Z_{-i}(t) = Z_i(t)$  for  $t = 2, \dots, K$ . Whereby the result for overwriting policies.

## C. General optimal policies

**Theorem 5.1:** Let  $K \geq 2$ . Then an optimal policy for P1, named *piecewise-threshold policy*, exists with the following structure:

- (i) There are thresholds,  $s_i \in [t_i, t_{i+1}]$ ,  $i = 1, \dots, K - 1$  for which  $u(t) = 1$  for  $t \in [t_i, s_i]$ . During the intervals  $[s_i, t_{i+1})$ ,  $u(t) = 0$



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2014

- (ii) After time  $tK$  it is optimal to always transmit a packet. An optimal policy  $u$  satisfies  $u(t) = 1$  for all  $t \geq tK$ .

**Proof.** (i) Let  $\mathbf{u}(t)$  be an arbitrary policy. Remember that  $u(t) = \sum_j u_j(t)$ . Assume that it does not satisfy (i) above. Then there exists some  $i = 1, \dots, K-1$ , such that  $\mathbf{u}(t)$  is not a threshold policy on the interval  $T_i := [ti, (i+1)t]$ . Hence there is a closed interval  $S = [a, b] \subset T_i$  such that for some  $p < 1$ ,  $u(t) \leq p$  for all  $t \in S$  and  $\int_b^{t+1} u(t)dt > 0$ . Then  $\mathbf{u}(t)$  can be strictly improved according to Lemma 5.2 and hence cannot be optimal.

(ii) Assume that the threshold  $sK$  satisfies  $sK < \tau$ . It is straightforward to show that by following  $\mathbf{u}(t)$  till time  $sK$  and then switching to any policy that satisfies  $u_i(t) > 0$  for  $sK < t \leq \tau$  for all  $i$ ,  $P_s(\tau)$  strictly increases.

**Remark:** The above theorem allows to conclude that the optimal policy for P1, i.e., for maximizing  $P_s(\tau)$  over all possible policies, can be searched in the set of the policies given by  $\{u(t), \{s_i/j_i\}$  for  $i = 1, \dots, K$ , where  $u(t) = 1$  in every interval  $[ti, (i+1)t]$ , in  $[tK, \tau]$ , and is zero outside. It is worth noting that WC policies are a particular case of piecewise threshold policies when all the  $s_i = (i+1)t$ , for  $i = 1, \dots, K-1$ .

### III. THE INTEL RATE CONTROLLER DESIGN

fuzzy logic traffic controller for controlling traffic in the network system defined. Called the IntelRate, it is a TISO (Two-Input Single Output) controller. The TBO (Target Buffer Occupancy)  $q_0 > 0$  is the queue size level we aim to achieve upon congestion. The queue deviation  $e(t) = q_0 - q(t)$  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,  $g(e(t)) = \int e(t) dt$ . The aggregate output is  $y(t) = \sum u_i(t - \tau_i)$ . Under heavy traffic situations, the IntelRate controller would compute an allowed sending rate  $u_i(t)$  for flow  $i$  according to the current IQSize so that  $q(t)$  can be stabilized around  $q$ . In our design, IQSize  $q(t)$  is the only parameter each router needs to measure in order to complete the closed-loop control. FLC is a non-linear map 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 normal tutorials/texts like [36], [45] 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  $\tilde{e}(t)$ ,  $\tilde{g}(e(t))$  and  $\tilde{u}(t)$ , respectively. There are  $N$  ( $N = 1, 2, 3, \dots$ ) LVs (Linguistic Values) assigned to each of these linguistic variables. Specifically, we let  $\overline{P}_i = \{\overline{P}_i^j : j = 1, 2, \dots, N\}$  be the input LVs with  $i = 1$  for  $\tilde{e}(t)$  and  $i = 2$  for  $\tilde{g}(e(t))$ , and let  $\overline{U} = \{\overline{U}^j : j = 1, 2, \dots, N\}$  for  $\tilde{u}(t)$ . For example, when  $N = 9$ , we can assign the following values for both the inputs  $e(t)$  and

$g(e(t))$ .  $\overline{P}_i^1 =$  "Negative Very Large (NV),"



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2014

Large (NL),”  $\bar{P}_i^3$  = “Negative Medium (NM),”  $\bar{P}_i^4$  = “Negative Small (NS),”  $\bar{P}_i^5$  = “Zero (ZR),”  $\bar{P}_i^6$  = “Positive Small (PS),”  $\bar{P}_i^7$  = “Positive Medium (PM),”  $\bar{P}_i^8$  = “Positive Large (PL),” and  $\bar{P}_i^9$  = “Positive Very Large (PV),”  $i = 1, 2$ . Similarly, we can designate the output when  $N = 9$  with the following linguistic values.  $\bar{U}^1$  = “Zero (ZR),”  $\bar{U}^2$  = “Extremely Small (ES),”  $\bar{U}^3$  = “Very Small (VS),”  $\bar{U}^4$  = “Small (SM),”  $\bar{U}^5$  = “Medium (MD),”  $\bar{U}^6$  = “Big (BG),”  $\bar{U}^7$  = “Very Big (VB),”  $\bar{U}^8$  = “Extremely Big (EB),” and  $\bar{U}^9$  = “Maximum (MX).”

## 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. We let  $P_1$  be the UoD<sup>1</sup> for the input  $p_1 = e(t)$ , and  $P_2$  be the UoD for the input  $p_2 = g(e(t))$ . The value of MFs (i.e., the certainty degree) for crisp inputs  $p_i (i = 1, 2)$  is designated by  $\mu_{P_i^j}(p_i)$ . Similarly, we let  $Z$  be the UoD of the output  $z = u(t)$ , and the certainty degree of the crisp output  $z$  is designated by  $\mu_{U^j}(z)$ . The above  $\mu_{P_i^j}(p_i)$  or  $\mu_{U^j}(z)$  is obtained by the “Singleton Fuzzification” method [36]. Thus the input and output fuzzy sets can be defined with  $P_i^j = \{(p_i, \mu_{P_i^j}(p_i) : p_i \in P_i)\}$ ,  $i = 1, 2$  and  $U^j = \{(z, \mu_{U^j}(z)) : z \in Z\}$ ,  $j = 1, 2, \dots, N$ , respectively.

## C. Defuzzification

For the defuzzification algorithm, the IntelRate controller applies the COG (Center of Gravity) method to obtain the crisp output with the equation  $u(t) = \left( \sum_{j=1}^k c_j S_j \right) / \left( \sum_{j=1}^k S_j \right)$ , where  $k$  is the number of rules;  $c_j$  is the bottom centroid of a triangular in the output MFs, and  $S_j$  is the area of a triangle with its top chopped off as per  $\mu_{P_1^m \cap P_2^m}$  discussed above. Since each parameter in the crisp input pair (p1,p2) can take on two different values in the IntelRate controller, we have altogether  $k=4$ rules for defuzzification each time.

## D. Design Parameters

From our design above, one can see there are different parameters which ultimately will affect the performance of our traffic controller. Below are the discussions of some important design issues we have experienced. Some of them were determined via our extensive experiments.

### a) TBO

From the perspective of the queueing delay, the TBO value should be as small as possible. This is especially true under state queueing delay, which is not desirable to some Internet applications such as the real-time video. the heavy traffic conditions when the queue is to be stabilized at TBO. Therefore, a bigger TBO will result in longer steady In short, the TBO should be chosen such that the network can have a reasonable queueing delay while maintaining good throughput and link utilization. For the IntelRate controller, we would choose a queue size (and therefore the worst node queueing delay,



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2014

e.g. less than 10 ms) in order to give a network delay acceptable to most of the real-time traffic while maintaining good performance.

## b) The number N of LVs

The choice of N has to consider the trade-off between the throughput performance and the computation complexity of the controller. Big N complicates the controller in the sense that it has to do more logic computations on choosing the allowed sending rate according to the rules. Such a computation complexity can affect the rise time in the transient response of the controller as well as the control performance in the case of network parameter changes (e.g. the settling time during large bandwidth variation). On the other hand, a small N may lead the controller output to oscillate due to the big partitions of the LVs.

## c) The Output Edge Value D

With reference to the outermost edge value D in the output MFs corresponds to the maximum sending rate that the controller can output. This parameter is chosen to be the maximum value of the Req\_rate field among the active incoming flows, i.e.  $D = \max(\hat{u}_1, \hat{u}_2, \dots, \hat{u}_R)$ , where  $i = 1, 2, \dots, R$ , is the value recorded in Req\_rate of each packet.

## d) The Width Limit m

The parameter m defines the base width of each membership function in the FS. Since it also affects the extent of overlapping between the adjacent MFs, the basic consideration to choose an appropriate parameter m is to have a smaller TBO while remaining the controller output smooth. An inappropriate m may have similar side effects like the parameter N. Too big of an m value leads to small partitions along e(t) or g(e(t)), and thus may affect the response time of the controller. On the other hand, having m too small may cause fluctuations in the controller output due to the too big partitions along e(t) and g(e(t)).

## e) Buffer Size B

The determination of buffer size B is closely related to the chosen value of TBO. Although one can choose  $B = q_0$ , the smallest possible value, this is usually not desirable for the following two basic reasons: 1) a controller usually has various steady state error issues, and it is impossible that the queue size can be exactly pegged at TBO; 2) the dynamic Internet traffic can sometimes cause a surge on the queue size, e.g., due to a sudden traffic swarm-in or an unexpected bandwidth reduction. Therefore, the B should be greater than the TBO.

## E. The Control Procedure

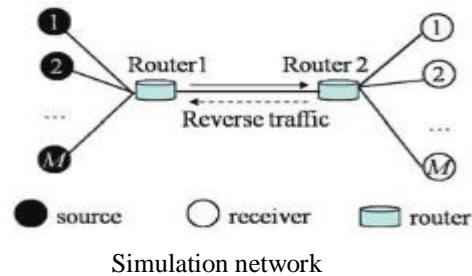
Below is a summary of the traffic-handling procedure of the IntelRate controller in a router.

- (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.
  - (3a) If  $u(t) < Req\_rate$ , it means that the link does not have enough bandwidth to accommodate the requested amount of sending rate. The Req\_rate field in the congestion header is then updated by u(t).
  - (3b) Otherwise the Req\_rate field remains unchanged.
- (4) If an operation cycle d is over, update the crisp output u(t) and the output edge value of D.

# International Journal of Innovative Research in Computer and Communication Engineering

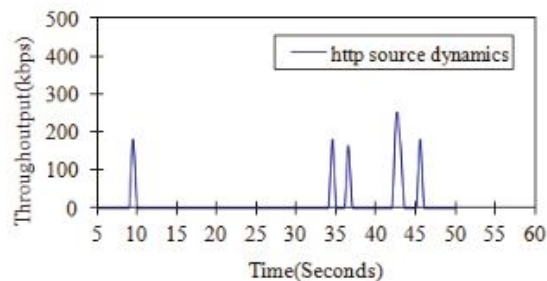
(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2014



## IV. PERFORMANCE EVALUATION

The single bottleneck network in Fig. 6 is used to investigate the controller behavior of the most congested router. We choose Router 1 as the only bottleneck in the network, whereas Router 2 is configured to have sufficiently high service rate and big buffer  $B$  so that congestion never happens there. The numbers in Fig. are the IDs of the subnets/groups attached to each router. Their configuration is summarized in which there are  $M = 11$  subnet pairs, which form the source-destination data flows in the network, and they run various Internet applications such as the long-lived ftp, short-lived http, or the unresponsive UDP-like flows (also called uncontrolled ftp flows [19]). Since the link bandwidth we want to simulate have a magnitude of Giga bits per second, we need to use 20 flows in each subnet to generate enough traffic to produce congestion. All flows within each group have the same RTPD and behavior, but different from the flows of other groups. The RTPD includes the forward path propagation delay and the feedback propagation delay, but does not include the queueing delay, which may vary according to our settings of TBO size in the experiments. The reverse traffic is generated by the destinations when they piggyback the ACK information back to the sources.



Http sessions example.

The simulation time depends on bottleneck bandwidth and the simulated time. A typical simulation run usually takes hours or days. For example, in order to observe the source throughput behavior before and after the network parameter change, we set a longer simulated time for such an experiment than a max-min fairness experiment. The number of packets generated in an experiment is related to the TBO value, the bandwidth, the simulated time and the traffic intensity. 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 [51]. Here, a bit is considered to be successfully sent out if it is part of a packet that has been successfully sent [51]. 2) IQSize is the length of the bottleneck buffer queue (measured in packets) seen by a departing packet [52]. 3) Queueing delay is the waiting time of a packet in the router queue before its service. Measurements are taken from the time the first bit of a packet is received at the queue until the time the first bit of the packet is transmitted. 4) Queueing jitter is the variation of queueing delay due to the queue length dynamics, and is defined as the variance of the queueing delay. 5) Link (or bottleneck) utilization is the ratio between

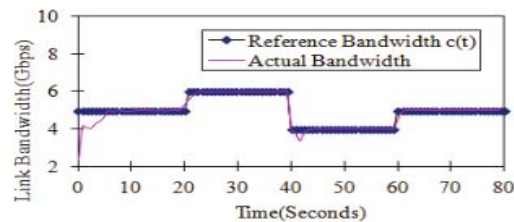


# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2014

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



**Bandwidth variations.**

We evaluate the utilization and packet loss rate performance of the IntelRate 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 45Mbps to 10Gbps. The simulation results show that the IntelRate 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). The reason of the zero packet loss is that the IntelRate controller can always control the variations of the IQSize around the TBO position. Therefore, the buffer never overflows and packets are never lost upon heavy traffic. In the meanwhile, the stable feature in IQSize and throughput guarantees the full bandwidth utilization. Next we fixed the bottleneck bandwidth at 1Gbps, and then change TBO from 800 to 2000 packets to check the link utilization and packet loss rate again. The same experiment results are observed, where the same reasons pertain.

## V. 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 existing congestion 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 of the 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.

## REFERENCES

- [1] M. Welzl, *Network Congestion Control: Managing Internet Traffic*. John Wiley & 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.



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2014

- [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. Dukkupati, 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] Y. Zhang and M. Ahmed, "A control theoretic analysis of XCP," in *Proc. 2005 IEEE INFOCOM*, vol. 4, pp. 2831-2835.
- [15] Y. Zhang and T. R. Henderson, "An implementation and experimental study of the explicit control protocol (XCP)," in *Proc. 2005 IEEE INFOCOM*, vol. 2, pp. 1037-1048.
- [16] J. Pu and M. Hamdi, "Enhancements on router-assisted congestion control for wireless networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, pp. 2253-2260, June 2008.
- [17] F. Abrantes, J. Araujo, and M. Ricardo, "Explicit congestion control algorithms for time varying capacity media," *IEEE Trans. Mobile Comput.*, vol. 10, no. 1, pp. 81-93, Jan. 2011.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] Y. Hong and O. Yang, "An API-RCP design using pole placement technique," in *Proc. 2011 IEEE ICC*, pp. 1-5.
- [23] B. Ribeiro, T. Ye, and D. Towsley, "Resource-minimalist flow size histogram estimator," in *Proc. 2008 ACM SIGCOMM Conf. Internet Measurement*, pp. 285-290.
- [24] 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.
- [25] L. Roberts, "Enhanced PRCA proportional rate control algorithm," AFTM-R, Aug. 1994.
- [26] 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.
- [27] 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.
- [28] D. Toelle and R. Knorr, "Congestion control for carrier ethernet using network potential," *Proc. 2006 IEEE/IFIP Netw. Operations Manage. Symp.*, pp. 1-4.
- [29] 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.
- [30] 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.
- [31] G. Pavlou, "Traffic engineering and quality of service management for IP-based NGNs," in *Proc. 2006 IEEE/IFIP Netw. Operations Manage. Symp.*, p. 589.
- [32] D. Chalmers and M. Sloman, "A survey of quality of service in mobile computing environments," *IEEE Commun. Surveys & Tutorials*, vol. 2, no. 2, pp. 2-10, 1999.
- [33] G. Kesidis, "Congestion control alternatives for residential broadband access," *Proc. 2010 IEEE Netw. Operations Manage. Symp.*, pp. 874-877.
- [34] J. Wang and V. Leung, "Incentive engineering at congested wireless access points using an integrated multiple time scale control mechanism," in *Proc. 2006 IEEE/IFIP IEEE Netw. Operations Manage. Symp.*, pp. 1-4.
- [35] S. Secci, M. Huaiyuan, B. Helvik, and J. Rougier, "Resilient inter-carrier traffic engineering for Internet peering interconnections," *IEEE Trans. Netw. Service Manage.*, vol. 8, no. 4, pp. 274-284, 2011.
- [36] K. M. Passino and S. Yurkovich, *Fuzzy Control*. AddisonWesley Longman Inc., 1998.
- [37] 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.
- [38] T. W. Vaneck, "Fuzzy guidance controller for an autonomous boat," *IEEE Control Syst. Mag.*, vol. 17, no. 2, pp. 43-51, Apr. 1997.
- [39] 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.
- [40] 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.
- [41] 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.
- [42] 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.
- [43] 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.



ISSN(Online): 2320-9801  
ISSN (Print): 2320-9798

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2014

- [44] 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.
- [45] S. Kaehler. Available: <http://www.seattlerobotics.org/encoder/mar 8/fuz/ flindex.html>
- [46] H. Ying, W. Siler, and J. J. Buckley, "Fuzzy control theory: a nonlinear case," *Automatica*, vol. 26, no. 3, pp. 513–520, 1990.
- [47] H. Jiang and C. Dovrolis, "Passive estimation of round-trip times," *ACM SIGCOMM Computer Commun. Rev.*, vol. 32, no. 3, 2002.
- [48] Available: <http://www.icir.org/floyd/ccmeasure.html>
- [49] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, 11th edition. Pearson Prentice Hall, 2008.
- [50] M. E. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: evidence and possible causes," *IEEE/ACM Trans. Netw.*, vol.5, no. 6, pp. 835–846, Dec. 1997.
- [51] "OPNET modeler manuals," OPNET version 11.5, OPNET Technologies Inc., 2005.
- [52] D. Gross, J. Shortle, J. Thompson, *et al.*, *Fundamentals of Queueing Theory*, 4th edition. John Wiley & Sons Inc., 2008.
- [53] J. Liu and O. Yang, "Stability analysis and evaluation of the IntelRate controller for high-speed heterogeneous networks," in *Proc. 2011 IEEE ICC*, pp. 1–5.