RULE BASED DELAY PROPORTION ADJUSTMENT FOR DIFFERENTIATED SERVICES

Sunthiti Patchararungruang*, Nirmala Shenoy**, and Saman K. Halgamuge*

* Mechatronics Research Group
Department of Mechanical and Manufacturing Engineering
The University of Melbourne, AUSTRALIA
sunthiti,sam@mame.mu.oz.au

** Department of Information Technology
Rochester Institute of Technology, Rochester, NY 14623
ns@it.rit.edu

Abstract – Differentiated services (DiffServ) are becoming very popular to be implemented to provide quality of service (QoS) to various Internet applications. Proportional Relative-DiffServ [4] is a recently proposed method aiming to guarantee the ratios of service differences between. However, the outcome of such a method is similar to spreading overall load to each class with a fixed ratio and it allows low priority traffic to affect high priority traffic which is unacceptable. In this paper, we introduce a controller using fuzzy rules [6] to reduce the effect of low priority class upon higher priority ones. Our system adjusts bias of each class using the current traffic condition of the class. The simulation shows that dependency of the delay of a high-priority-class, a value that indicates QoS of the traffic, on lower priority classes is significantly reduced by the proposed method.

I. INTRODUCTION

As the Internet becomes an important infrastructure of global communication, the best-effort service cannot meet diverse expectations of applications. Some applications prefer low data-loss rate while tolerate high delay such as world-wide-web (www) and file transfers. However, multimedia applications requirements are low delays but tolerate amount of data loss. Those preferred conditions those applications need for providing good services is called quality-of-service (QoS).

There are two paradigms proposed to provide QoS for Internet applications. They are Integrated Services (IntServ) [1] and Differentiated Service (DiffServ) [2]. IntServ tries to ensure end-to-end and per-flow QoS. All connections should reserve the resources needed and routers should keep reservation parameters. However, for each connection, it is difficult to implement on a complex and large network such as the Internet because of its complexity. To reduce the complexity of QoS management, DiffServ was introduced to provide QoS within a domain using aggregation of flow and per-class service.

In practical, QoS parameters are not easy to control because the Internet is shared network. QoS variation is depended on load of the network related to traffic from all hosts connected to the Internet. The more guarantee requirements, the more complex of the implementation. Currently, there are two important approaches to guarantee the QoS relatively to the higher and lower priority classes. They are price relative [3] and proportional DiffServ schemes [4]. The later one with bias features is the main focus of this paper.

Proportional DiffServ (Prop-DiffServ) has been recently investigated extensively. It tries to keep QoS ratio of each class at a value configured by an administrator. Some contemporary studies adapt the strategies of reducing the management complexity by pre-defining all class parameters. Therefore, call-admission-control (CAC) is unnecessary and users cannot change their traffic preferences. A drawback of this approach is that QoS of a particular class depends on QoS of other classes. If QoS of a class drops, Prop-DiffServ algorithm will drop QoS of others to maintain the QoS ratio. This situation is unsuitable for critical application such as tele-medicine because other traffics can disturb its QoS and lead to severe hazard. Hence, we felt the need for defining the QoS ratio so that high priority traffics are more tolerable to traffic conditions of lower priority classes.

In this paper, we present a fuzzy-controlled bias-based Prop-DiffServ to reduce dependency of QoS of the high priority class on lower priority classes. The result from simulator shows that the proposed solution is effective. The organization of this paper is as follows: Section II gives a brief overview of Prop-DiffServ. Section III presents the proposed approach. Section IV shows the implementation of fuzzy controller into the proposed solution. Section V and VI describe the simulation model and provide the discussion of results. The conclusion of this work is given in Section VII.

II. PROPORTIONAL DIFFERENTIATED SERVICES

In Prop-DiffServ [4], services for each class should be proportional to the ratio set in the service contract to the customer. The contract can be pre-defined or negotiable. Suppose that a domain provides N service classes and

$\overline{q}_i(t, t+\tau)$ is the average QoS value for class i in the time interval $(t, t+\tau)$, where $\tau > 0$. In Prop-DiffServ, the following equation should be satisfied for any pair of classes.

$$\frac{\overline{q}_i(t, t+\tau)}{\overline{q}_j(t, t+\tau)} = \frac{c_i}{c_j} \quad ; \quad i \neq j \tag{1}$$

Where c_i is the service differentiation parameter of class i which is fixed by the contract.

In this paper, the QoS parameter we focus on is queuing delay only. Therefore, the equation 1 can be rewritten as:

$$\frac{\overline{d}_i(t, t+\tau)}{\overline{d}_j(t, t+\tau)} = \frac{\delta_i}{\delta_j} \quad ; \quad i \neq j \tag{2}$$

Where $\overline{d}_i(t, t+\tau)$ is the average queuing delay for class i in the time interval $(t, t+\tau)$ and δ_i is fixed delay differentiation parameter (DDP) [4] of class i.

III. PROPOSED ALGORITHM

In equation 2, average queuing delay of classes i and j are dependent on each other because the value of δ_i and δ_j are fixed. This work attempts to make queuing delay of high priority class independent from lower priority classes by modifying the equation to:

$$\frac{\overline{d}_i(t, t+\tau)}{\overline{d}_j(t, t+\tau)} = \frac{s_i}{s_j} \quad ; \quad i \neq j \tag{3}$$

Where s_i is given as:

$$s_i = \delta_i + b_i \tag{4}$$

Where b_i is differentiation bias of class i and will vary on the load of class i to reduce dependency of class j on class i (given that class j has lower priority than class i). However, we try to maintain the ratio of s_j to s_i at the same value as the ratio of δ_j to δ_i because it is the ratio set by the contract. Therefore, b_j is evaluated from the combination of load of class j and all other classes that have higher priority than class j. Let β_i is the bias value calculated from the load of class i and class priority is sorted in descending order. We can calculate combination bias (b_i) as:

$$b_i = \begin{cases} \beta_i + \dfrac{\delta_i}{\delta_{i-1}} b_{i-1} & ; i > 0 \\ \beta_i & ; i = 0 \end{cases} \tag{5}$$

The equation 5 is recursive equation that can be rewritten as:

$$b_i = \begin{cases} \beta_i + \delta_i \sum_{k=0}^{i-1} \dfrac{\beta_k}{\delta_k} & ; i > 0 \\ \beta_i & ; i = 0 \end{cases} \tag{6}$$

In equation 6, the parameter needed to evaluate is β_i. The evaluation process is described in the next session.

IV. IMPLEMENTATION OF FUZZY CONTROLLER

To find the value of β_i, we have to find parameters and their relationships to evaluate load of class i. There are many associated parameters such as queue backlog and the calculated queue-delay itself. It is very difficult to find equation to combine these parameters emphatically to estimate β_i because traffic characteristic is dependent on user application and user command. However, we generally use fuzzy term to describe those parameters such as "low queue delay means low load". Further, there is strong endorse that fuzzy logic controller can achieve good performance in telecommunication approach [8][9]. Therefore, we implement fuzzy system described in [7] to calculate β_i from these parameters. Our system diagram is shown in Figure 1.

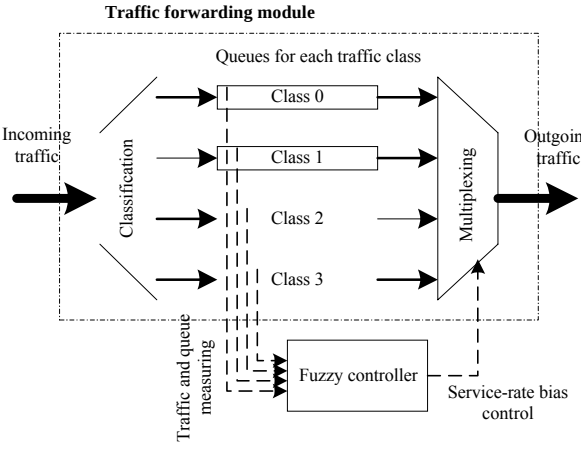


Figure 1: System diagram

The fuzzy system implemented in [7] uses singleton output membership functions, which is simpler and easier to implement than common triangular and trapezoid membership functions. We choose such implementation because our further research is implementing the system into a hardware chip (i.e. VLSI chip) for real-time traffic control. The simulation proves that this simple implementation can operate efficiently and provide impressive results.

Let Δd is the queue-delay change and Δv is arrival rate change as percentage compared to their average values. The average can be calculated by exponential moving average (EMA) with average weight 0.1. The symbol $\Delta\beta$ represents the change of β_i, is used as the output of the controller. We use Δv and Δd as the inputs to the fuzzy controller. Both input parameters are limited to the range [-100, 100]. We define membership functions for the inputs as shown in Figure 2(a) and fuzzy output singletons in Figure 2(b).

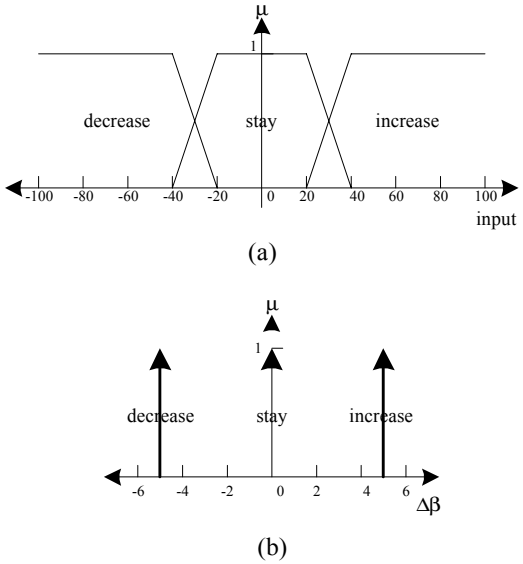


Figure 2: Membership functions

The rule base is quite simple. The value of β_i should increase when traffic input-rate of class i is increased and decrease when the rate and queue-delay is decreased. We define four fuzzy rules associated to all three input membership functions and two inputs:

R1: IF Δv IS increase THEN $\Delta\beta$ IS increase
R2: IF Δd IS decrease THEN $\Delta\beta$ IS decrease
R3: IF Δv IS increase AND Δd IS decrease THEN $\Delta\beta$ IS stay
R4: IF Δv IS stay OR Δv IS decrease OR Δd IS increase OR Δd IS stay THEN $\Delta\beta$ IS stay

After evaluating $\Delta\beta$, we apply weighted average as defuzzification method to calculate crisp value of $\Delta\beta$. The weighted-average equation is:

$$\Delta\beta_{crisp} = \frac{\sum_{k=1}^{n} F_k \quad \Delta\beta_k}{\sum_{k=1}^{n} F_k} \qquad (7)$$

Where F_k is degree of membership function $\Delta\beta_k$. The final output, new β, can be calculated by equation 8. The new value of β is, then, applied to equation 6 to control the service ratio.

$$\beta_{new} = \beta_{prev} + \Delta\beta_{crisp} \qquad (8)$$

V. SIMULATION CONFIGURATION

In order to study our model, simulations were conducted using the topology shown in Figure 3. We use NS2 network simulator developed by National Berkeley Laboratory as the simulation platform [5]. The topology consists of five nodes – four traffic-sources (S0 to S3) and one traffic-destination (D) – and two DiffServ routers (R0 and R1). All connections between nodes and routers have capacity of 1 Mbps with 2 ms delay. All traffic sources send traffic to the destination D through R0 and R1. Packets between routers are classified into 4 classes where class 0 is the highest priority and class 3 is the lowest one. We assume that packets generated from S0 are categorized into class 0, packets generated from S1 are grouped into class 1, and so on for the other sources. We apply Waiting-Time Priority queues (WTP) [4] as the packet scheduler in both routers.

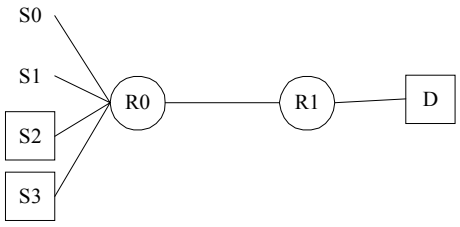


Figure 3: Network topology in the simulations.

Each source generates 250 kbps exponential and 95 kbps constant-bit-rate (CBR) traffic as the normal operation. The exponential traffic has 900ms average "on" time and 500ms average "off" time. We operate the simulation for 250 seconds. To simulate burst bulky data, which suddenly entered the network, we generate 500 kbps CBR traffic to a

selected source starting from 100 second with 50-second duration. We also assume that the size of all packets is 1024 bytes. Moreover, we configure the delay differentiation ratios between classes as δ_0: δ_1: δ_2: δ_3 to 1:2:4:8. All measurements and adjustments are performed every 2 seconds.

VI. SIMULATION RESULT

The results of the simulation are shown as graphs of queue delay in router R0. The simulation aims to compare the delay in conventional Prop-DiffServ and the Prop-DiffServ with our fuzzy controller. We simulate the topology with four scenarios.

- Burst traffic enters from S1 without our bias mechanism.
- Burst traffic enters from S2 without our bias mechanism.
- Burst traffic enters from S1 with our bias mechanism in R0.
- Burst traffic enters from S2 with our bias mechanism in R0.

Four graphs from each scenario are presented in Figure 4 (a) to Figure 4(d) respectively. The X-axis of all graphs is simulation time in seconds and the Y-axis is queue delay in seconds. In the graphs, the results from the first two scenarios, Figure 4(a) and 4(b), are very similar because the system strictly maintains the delay ratio regardless from traffic source. When we turn on the fuzzy controller in R0 and re-simulate the scenarios again, the output graphs, shown in Figure 4(c) and 4(d), indicates that the effect of burst traffic upon the classes which have higher priority than the particular class is significantly reduced. Comparing between Figure 4(a) and 4(c), the area under class 0 line in Figure 4(c) (the first line from bottom) is less than that of Figure 4(a). There is the same truth for Figure 4(b) and 4(d) when the burst traffic is pumped into class 2 (the third line from bottom). In Figure 4(d), the area under class 1 (the second line from bottom) and class 0 are less than the correspondent areas in Figure 4(b).

Figure 4(e) and 4(f) show the outputs without and with fuzzy controller when no bulky data were pumped into the system. The graphs show that even without any bulky data our controller can also make the area underneath high priority class less than lower priority classes.

VII. CONCLUSION

In this paper, we presented a proportional delay differentiated service with fuzzy controller to adjust bias value. Simulation shows that our proposed algorithm can reduce effect of low priority traffic upon higher priority ones. We assume that the queue in each router is infinite and do not consider other QoS parameters such as packet-loss. Moreover, our algorithm can be improved by including more parameters and adjusting fuzzy membership functions and rule bases. These issues will be further explored in future work.

REFERENCES

[1] P.P. White, "RSVP and Integrated Services in the Internet: A Tutorial", IEEE Communications Magazine, pp. 100-106, May 1997.

[2] S. Blake, et al., "An Architecture for Differentiated Services", RFC 2475, December 1998.

[3] A.M. Odlyzko, "Paris Metro Pricing: The Minimalist Differentiated Services Solution", Proceedings IEEE/IFIP International Workshop on Quality of Service, June 1999.

[4] C. Dovrolis, D. Stiliadis, and P. Ramanathan, "Proportional Differentiated Services: Delay Differentiation and Packet Scheduling", Proceeding of ACM SIGCOMM'99, Aug 1999.

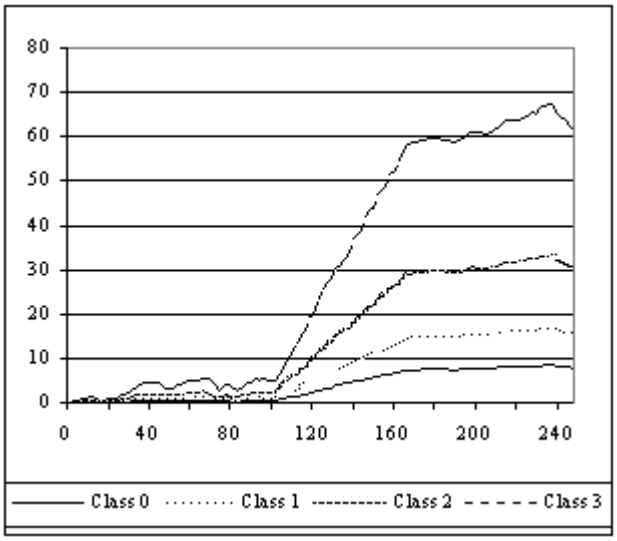
[5] Network Simulator (ns) version 2, "http://www.isi.edu/nsnam/ns/"

[6] L. Zadeh, "Soft Computing and Fuzzy Logic", IEEE Software, Volume 11(6), 1994

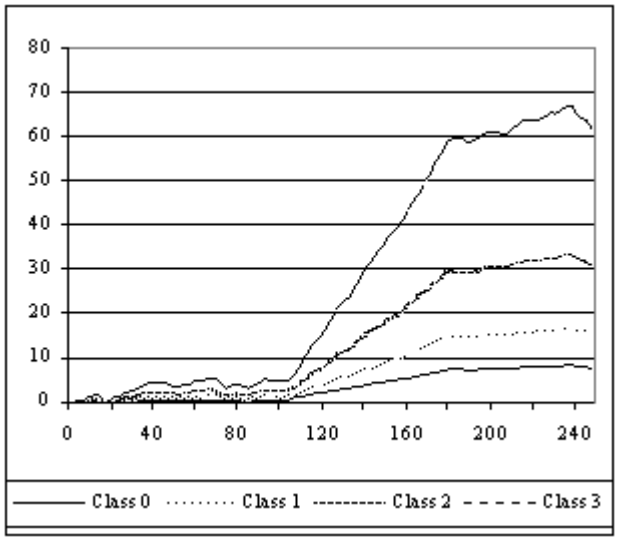
[7] P. Spasov, "Microcontroller Technology: The 68HC11 – 2nd Edition", Prentice-Hall Inc., U.S.A., 1996

[8] N. Shenoy and S.K. Halgamuge, "A Fuzzy Logic Based Call Admission Controller with Feedback for ATM Networks", Proceeding of IEEE Enterprise Networking and Computing Conference 1998, June 1998

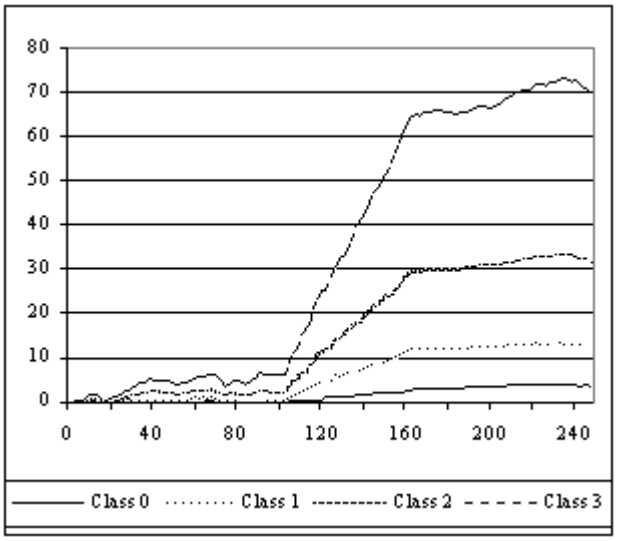
[9] R. Zhang and J. Ma, "Fuzzy QoS Management in Diff-Serv Networks", Proceeding of IEEE International Conference on Systems, Man, and Cybernetics 2000, October 2000



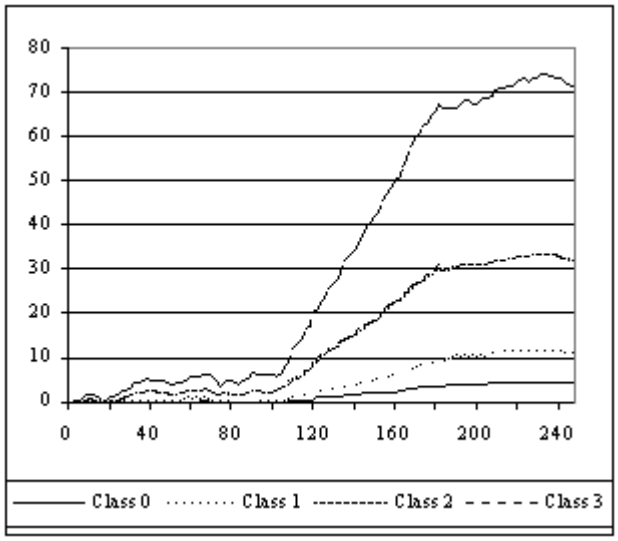
(a) pump into class 1, w/o the controller



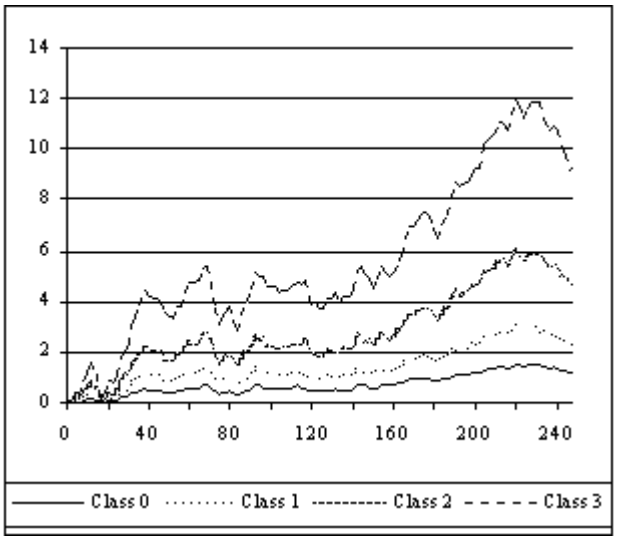
(b) pump into class 2, w/o the controller



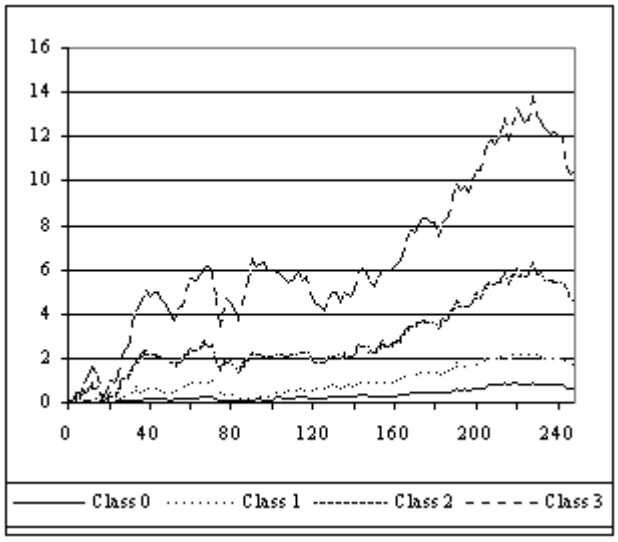
(c) pump into class 1, with the controller



(d) pump into class 2, with the controller



(e) No burst pump, w/o the controller



(f) No burst pump, with the controller

Figure 4: Simulation results