A Fuzzy Approach for Delay Proportion Differentiated Service

Mehran Garmehi, and Yasser Mansouri

***Abstract*—**There are two paradigms proposed to provide QoS for Internet applications: Integrated service (IntServ) and Differentiated service (DiffServ).Intserv is not appropriate for large network like Internet. Because is very complex. Therefore, 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 theses networks QoS between classes is constant and it allows low priority traffic to be effected from high priority traffic, which is not suitable. In this paper, we proposed a fuzzy controller, which reduced the effect of low priority class on higher priority ones. Our simulations shows that, our approach reduces the latency dependency of low priority class on higher priority ones, in an effective manner.

***Keywords*—**QoS, Differentiated Service (DiffServ), Fuzzy Controller, Delay.

I. Introduction

QUALITY OF SERVICE (QoS) is a generic term which takes into account several techniques and strategies that could assure application and users a predictable service from the network and other components involved, such as operating systems.

The reasons for supporting QoS models in the future are the appearance of time sensitive applications and the more and more ubiquitous use of Internet as work tool, congestion and uncertainties in delay and delay variation. The traditional Internet, storing and forwarding packets without guaranteed service can provide best-effort service only, and cannot provide acceptable performance. New real-time applications, which are less elastic and less tolerant to delay, packet losses and delay variations cannot be handled properly within the traditional data service architecture.

In some applications, low data loss rate is preferred; however, they tolerate a high data delay, such as file transfer and web. Nevertheless, multimedia applications require a low delay and tolerate the data loss.

In order to provide QoS support in the Internet, two service architectures – the *Integrated Service* (IntServ) [1, 2] and the Differentiated Services (DiffServ) architectures - have been developed [3].

M. Garmehi is with Department of Computer Science and Engineering, Faculty of Engineering, Bojnourd University, Bojnourd, Iran (e-mail: Garmehi@ce.sharif.edu).

Y. Mansouri is with Ferdowsi University of Mashhad, Mashhad, Iran (e-mail: ya_ma20@stu.um.ac.ir).

The IntServ approach supports some quantified services such as minimum-service rate or a maximum tolerable end-to-end delay or loss rate for application sessions.

In order to support this type of service, each router in the network has to maintain state and control information for each flow, which is a stream of packets belong to the same application session. This approach seems to be unfeasible for routers to perform all the above actions efficiently when there are millions of flows traversing through the network simultaneously.

The other approach, DiffServ, is newer than the IntServ approach. It proposes a coarser notion of quality of service, focusing primarily on classes, and intends to qualitatively differentiate services between classes rather than to provide absolute per-flow QoS guarantees. In particular, access routers process packets based on finer traffic granularity such as per-flow or per-organization while routers at the core network do not maintain fine-grained state, but process traffic based on a small number of Per Hop Behaviors (PHBs) encoded in the packet header.

A DiffServ model that draws much attention from the research communities recently is the *Proportional Differentiated Services* Model [4], which provides proportional services between different classes. There exist some studies on mechanisms to provide the proportional service, such as *Proportional Delay Service* (PDDM) and *Proportional Loss Service* [5]. Waiting Time Priority WTP or Backlog Proportional Rates BPR are scheduling mechanisms designed specially for the PDDM model in [4]. Even when such mechanisms are implemented at every router, it is not always possible to receive per-class proportional service in an end-to-end manner.

One of the drawbacks of the DiffServ method is the dependence of the QoS of a class with other classes. If the QoS of a class drops, the other classes also experience a decrease in their QoS level. This situation is not appropriate for critical applications i.e. Internet application. Therefore, we need to define a traffic ratio, not to allow the lower traffic class drops, affect the QoS level of higher traffic classes.

In this paper, we present a fuzzy controller to reduce the dependency of QoS of the higher priority classes on the lower ones. Our simulation shows that our approach is effective. The organization of this paper is as follows: section 2 describes background of proportional differentiated model, section 3 presents the proposed approach, section 4 describes the simulation model and discussion of results. The conclusion of this paper is given in section 5.

II. PRPORTIONAL DIFFERENTIATED MODEL

Formally, the proportional differentiation model of the performance of class i is illustrated by the following equation:

$$\frac{\phi_i}{\phi_j} = \frac{\pi_i}{\pi_j}, 1 \le i, j \le N \qquad (1)$$

Where π_i is the differentiation parameter of class i, and φ_i is the performance metric of the class, such as average delay, delay jitter or loss rate. Note that if lower values of ϕ_i lead to better performance, I must have that $\pi_i < \pi_j$ i>j.

The class, which has lower differentiation parameters, will get better performance (delay, delay jitter or loss rate) than the classes that have higher differentiation parameters. These proportional differentiation parameters are used to adjust performances of classes so that they stay proportional with each other.

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

$$\frac{\overline{d_i}}{\overline{d_j}} = \frac{\delta_i}{\delta_j}, 1 \le i, j \le N \qquad (2)$$

Where δ_i is the *Delay Differentiation Parameter* (DDP) for class *i*. The DDPs are ordered as $\delta_1 > \delta_2 > ... > \delta_N > 0$.

III. OUR APPROACH

In the equation (2) the average queuing delay in class i and j are reflected to which other, since δ_i and δ_j are constant (according to the agreement customer and provider). In our approach, we tried to independent the delay of entrance to the high priority class from low priority class.

Thus, our proposed fuzzy controller has two inputs and output. The inputs are variation of delay (Δd) and variation of rate arrival (Δa). We compare the variation of input variables with average of the indicated as a percent (between -100 to 100). The output of fuzzy controller is ΔL, which shows the variation load for each class.

Because triangular and trapezoidal shaped member shapes membership functions offer more computational simplicity, we have used trapezoidal for variation of delay (Δd), variation of rate arrival (Δa) and variation of load (ΔL). Figure 1 shows the fuzzy sets for our inputs and output. Based on this figure, the term set of input variables and output are Decrease, Increase and Fix.

We design the rule base, based on our experiment and how the systems should work. The design of the rule base comprises two steps: first, linguistic values are set and then membership functions of linguistic values are determined. Our rule base is simple, i.e., whenever the rate of entrance traffic to class i increase, L_i of class i increase too. And whenever the queue delay of this class decreases, L_i decrease too. Figure 2 Shows rule base of our fuzzy system, tuned with our simulation data.

Te design the rule base, we used two methods. Firstly, we used try and fail method. This method is based knowledge obtained from the experiment. A set of IF-THEN rules is constructed and then the system as been tested.

If the system has not had the expected behavior, the rules are modified to gain the required results. The other method for obtaining the fuzzy rules is using theory approach. In this method, rules are designed to satisfy the equation (2).

Fuzzification is the processes of calculating suitable sets of degree of membership, called "fuzzy sets", for crisp inputs. As we mentioned earlier, fuzzy controller utilizes two input variables, variation of delay (Δd), variation of rate arrival (Δa). In certain periods of time, the amount of these two variables is monitored and fed into fuzzy controller. These values are the fuzzified using variable and Fuzzification.

Referring to rule base, rule which their antecedent are fired according to these fuzzified values determined. To select between fired rules, we used *min-max* combination method which defines the zone the input value belonging to it.

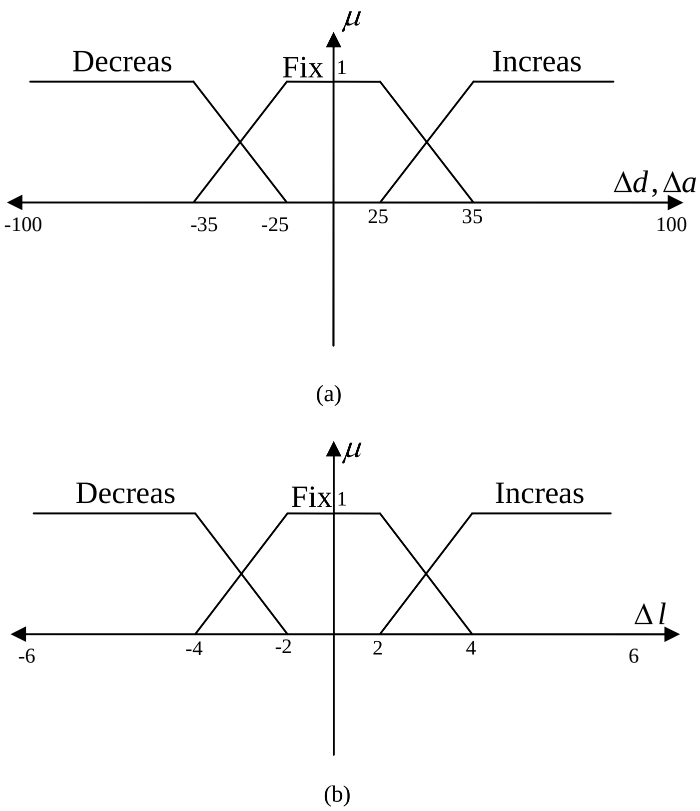


Fig. 1 Membership functions of the representation the inputs and output variables: (a) variation of delay (Δd) and rate arrival (Δa) (b) variation of load (ΔL)

/* Linguistic rules of controller */
If Δd **Decrease** and Δa **Decrease** then Δl is **Decrease**.
If Δd **Decrease** and Δa **Fix** then Δl is **Decrease**.
If Δd **Decrease** and Δa **Increase** then Δl is **Fix**.
If Δd **Fix** and Δa **Decrease** then Δl is **Fix**.
If Δd **Fix** and Δa **Fix** then Δl is **Fix**.
If Δd **Fix** and Δa **Increase** then Δl is **Increase**.
If Δd **Increase** and Δa **Decrease** then Δl is **Fix**.
If Δd **Increase** and Δa **Fix** then Δl is **Fix**.
If Δd **Increase** and Δa **Increase** then Δl is **Increase**.

Fig. 2 Linguistic rules for fuzzy controller

Referring to rule base, rule which their antecedent are fired according to these fuzzified values determined. To select between fired rules, we used *min-max* combination method, which defines the zone the input value belonging to it.

Finally, the implied output sets are combined to formulate a crisp output. The center of gravity (COG) technique, which computed the weighted-average of the center of gravity of each membership function, is used primarily [6].

The COG of the our fuzzy controller can be calculated using

$$\Delta L_{crisp} = \frac{\sum_{k=1}^{n} \mu_k \Delta L_k}{\sum_{k=1}^{n} \mu_k} \quad (3)$$

which μ_k is degree of membership function ΔL_k .

IV. Simulation and Result

To simulate our approach, we used ns2 [7] simulator. In order to test the performance of out algorithm, we used the network topology shown in Fig. 3. As shown in this figure, four traffic sources (S1, S2, S3, S4) and a node D as the destan- tion of the traffic. All the communication between nodes routers has 100 MB link capacity and 2 ms delay.

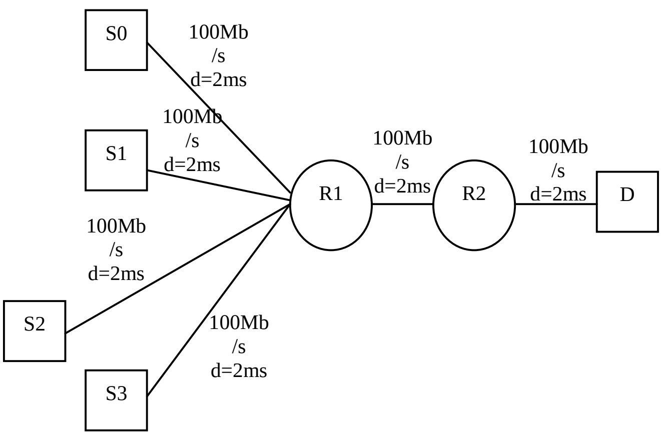


Fig. 3 Network topology used for simulation

The packets are classified into four classes in the routers which class 1 has the highest priority. Assume that the source S1 generate packet with first priority and S2 with second priority and so for. Farther more, the ratio delay differentiation, between classes is as follows:

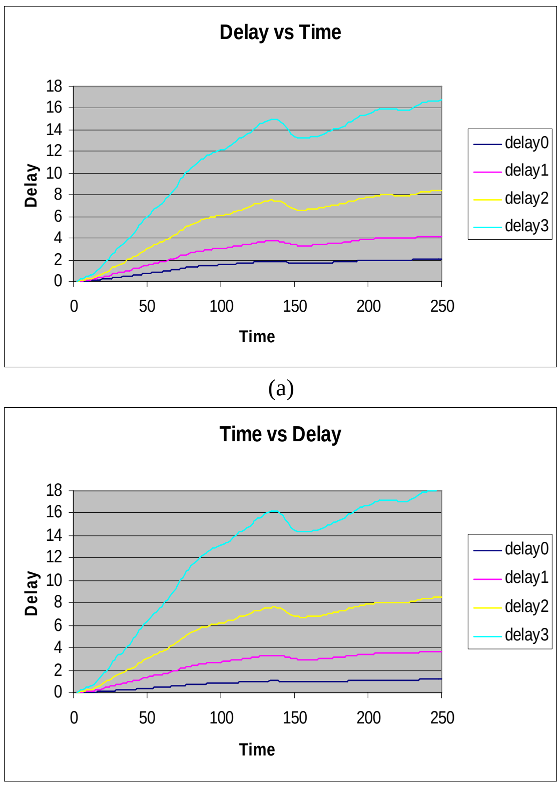
$$\delta_1 = 1, \delta_2 = 2, \delta_3 = 4, \delta_4 = 8$$

Each source generates a 100 Kbps CBR traffic and 250 Kbps exponential traffics. Exponential traffics 400 ms off and 800 ms on. In the simulation, in order to generate a burst traffic which suddenly happens in the network, we generate a 400 Kbps CBR traffic from a node at the 100th second for 60 seconds.

However, in our environment that is described before, we implemented tree experiments that each experiment is running for 250ms.results of these experiments are diagrams that related with router R1, which in these diagrams the horizontal axis is time and vertical axis is qeue delay.

A. Results from Experiment 1

Without enter a burst at the beginning of the simulation. Fig. 4 shows the packet delay variation in the queue of the router R1 in the period of simulation (we run the simulation 250 ms). Fig. 4a is without fuzzy controller (uses WTP) and Fig. 4b uses our approach (fuzzy Controller) the comparison of above simulation shows that the delay of the classes with fuzzy controller is much lower than the non-fuzzy controller.

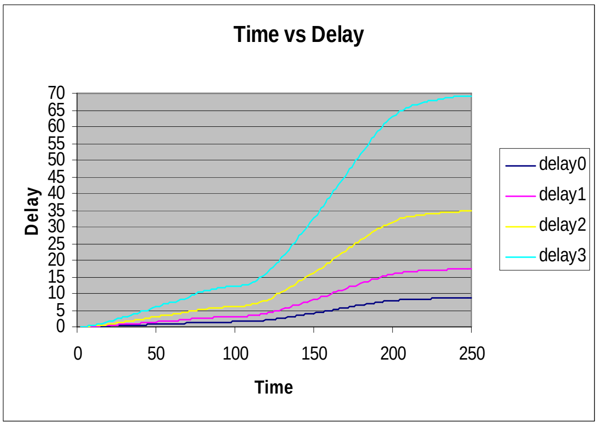


(b)

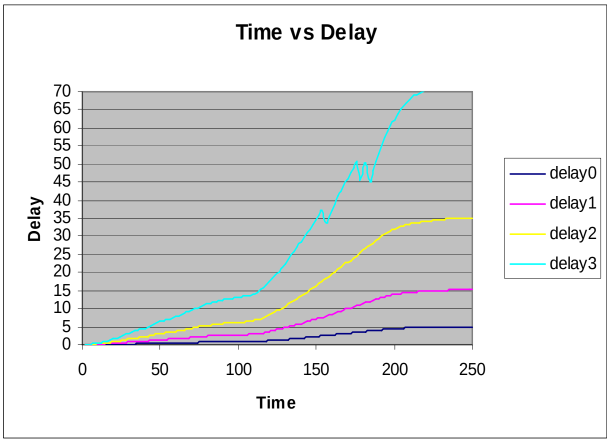
Fig. 4 Results from experiment 1without burst traffic (a) without fuzzy controller (b) with fuzzy controller

B. Results from Experiment 2

We penetrate a burst traffic flow from node S2 to the network. In Fig. 5a, the packet scheduler in R1 and R2 is WTP and in Fig. 5b, the packet scheduler in R1 is fuzzy controller and in R2 is WTP. The comparison between the figures show that the queue delay in class 0 and class 1 with fuzzy controller is much lower than common one.



(a)



(b)

Fig. 5 Results from experiment 2 with burst traffic into class 2 (a) with WTP (b) with fuzzy controller

C. Results from Experiment 3

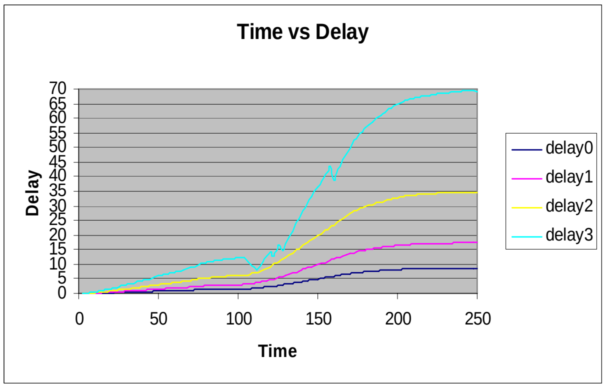
We used the same scenario as in experiment 2, accept that the burst traffic flow is penetrated from node S1; again we can see a lower queue delay from class 0 with fuzzy controller.

V. Conclusion

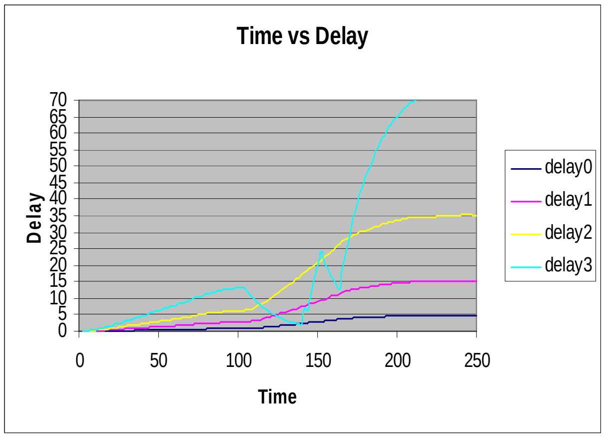
Differentiation services (DiffServ) are becoming varied popular to be implemented to provide quality of Service (QoS) to various internet applications. However, the drawback of the DiffServ method is the dependence of the QoS of a class with other classes. This situation is not suitable for critical application, e.g. multimedia.

Thus in this paper, we presented fuzzy controller for proportional delay differentiated service, which decreased the dependency of the Qos of high priority classes to low priority classes. We have our fuzzy controller work based on two inputs ($\Delta d, \Delta a$) and output (ΔL_i), which shows the variation of load from class i.

To evaluate our system performance, we have done three experiments, in the first experiment, we have not any burst flow, and in two other experiments, we had burst flow. In all experiments, our approach can reduce effect of low priority traffic upon high priority on.



(a)



(b)

Fig. 6 Results from experiment3 with burst traffic into class1 (a) with WTP (b) with fuzzy controller

References

[1] J. Wroclawski, "The Use of RSVP with IETF Integrated Services," September 1997.

[2] R. Braden, D. Clark, and S. Shenker, Integrated Services in the Internet Architecture: an Overview," July 1994. RFC 1633.

[3] K. Nichols, S. Blake, F. Baker and D. L. Black, "Definition of the Differentiated Services Field (DS Field) in the Ipv4 and Ipv6 Header," December 1998. IETF RFC 2474.

[4] C. Dovrolis, D. Stiliadis and P. Ramanathan, "Proportional Differentiated Services: Delay Differentiation and Packet Scheduling," *In Proceedings of the 1999 ACM SIGCOMMM*, Cambridge, MA, Sept. 1999.

[5] C. Dovrolis and P.Ramanathan, Proportional Differentiated Services, Part II: Loss Rate Differentiation and Packet Dropping," *In Proceedings of the 2000 International Workshop on Quality of Service (IWQoS)*, Pittsburgh PA, June 2000.

[6] L.Zadeh, "soft computing and fuzzy logic," IEEE software, vo1. 11, no. 6, 1994.

[7] "Network simulator(ns) version 2 software." [online].Available:http://www.isi.edu/nsnam/ns/.