논문 2006-43TC-6-4

차세대 네트워크에서의 적응형 지연 차별화 방식

(Adaptive Delay Differentiation in Next-Generation Networks)

백정훈*, 박재우**, 이유경**

(Jung Hoon Paik, Jae woo Park, and Yoo kyung Lee)

요 약

본 논문에서는 패킷 네트워크에서의 성능 품질을 개선하기 위한 접근방식으로 적응형 절대적/상대적 지연 차별화 방식을 제안한다. 제안된 방식은 타임 슬롯의 시작점에서 추후 도착될 트래픽의 양을 예측하여 지연 차별화 기능을 수행한 후 해당 타임 슬롯의 종단점에서 실제로 입력된 트래픽의 양을 실측하여 예측치와 실측치간의 차이를 도출하고 이를 다음 타임 슬롯의 지연 차별화 동작에 반영하여 성능 개선을 추구하는 것이 특징이다.. 제안된 방식은 지속적으로 예상 트래픽의 오차를 보상함으로서 지수형 트래픽 뿐만 아니라 버스트 트래픽에 대하여 기존 방식에 비하여 우수한 적응성을 제공한다. 모의실험을 통하여 제안된 방식은 엄격한 지연 목표를 충족할 뿐만 아니라 기존 방식에 비하여 트래픽 변동에 우수한 적응성을 제공함을 입증한다.

Abstract

In this paper, an algorithm that provisions absolute and proportional differentiation of packet delays is proposed with an objective for enhancing quality of service (QoS) in future packet networks. It features a scheme that compensates the deviation for prediction on the traffic to be arrived continuously. It predicts the traffic to be arrived at the beginning of a time slot and measures the actual arrived traffic at the end of the time slot and derives the difference between them. The deviation is utilized to the delay control operation for the next time slot to offset it. As it compensates the prediction error continuously, it shows superior adaptability to the bursty traffic as well as the exponential traffic. It is demonstrated through simulation that the algorithm meets the quantitative delay bounds and shows superiority to the traffic fluctuation in comparison with the conventional non-adaptive mechanism

Keywords : QoS, Absolute, Proportional, Delay, Differentiation

I. Introduction

Two broad paradigms for quality-of-service in the Internet have emerged, namely integrated services (IntServ) and differentiated services(DiffServ)[1][2]. The IntServ model, which aims to provide hard end-to-end QoS guarantees to each individual data flow, requires per-flow-based resource allocation and service provisioning and, thus, suffers from the scalability and manageability problems due to the huge amount of data flows.

This lack of scalability is, to a large extent, being addressed within the DiffServ architecture. In the DiffServ model, traffic is aggregated into a finite number of service classes that receive different forwarding treatment. It achieves scalability and manageability by providing quality per traffic aggregate and not per application flow. However, it's drawback is difficulty in contriving efficient resource allocation mechanisms to guarantee the end-to-end QoS of each individual data flow.

With superiority in terms of scalability and manageability, the DiffServ is gaining more

---

* 정회원, 동아방송대학
(DongAh Broadcasting College)
** 정회원, 한국전자통신연구원
(Electronics and Telecommunications Research Institute)

접수일자: 2006년3월17일, 수정완료일: 2006년6월9일

popularity as the QoS paradigm for the future Internet. Several schemes are devised to realize the DiffServ philosophy. At one end of the spectrum, absolute differentiated services seek to provide end-to-end absolute performance measures without per-flow state in the network core[3]. At the other end of the spectrum, relative differentiated services seek to provide per-class relative services[4]. In this model, the traffic from a higher priority class will receive no worse service than the traffic from a lower priority class.

In our view, absolute differentiated service is essential for handling a real-time application which requires guaranteed QoS measures for future Internet. In addition, proportional differentiated service is also needed to handle the soft-real time service which is tolerant to occasional delay violations and hence do not require strict delay bounds.

Consequently, it is perceived that the QoS architecture that provides any mix of absolute and relative differentiated schemes under the DiffServ paradigm is the most suitable service architectures for future Internet.

In this paper, an algorithm that enforces absolute and proportional differentiation of packet delays is proposed. In [5], Joint Buffer Management and Scheduling(JoBS) scheme is suggested, and it provides relative and absolute per-class service differentiation for delays and loss rate. It makes predictions on the delays of backlogged traffic, and uses the predictions to update the service rate of classes and the amount of traffic to be dropped. Our approach is similar to [5] in that it predicts delays of backlogged traffic and uses the predictions to update the service rate of classes, but main difference is whether the prediction error which occurs indispensably is utilized on future control operation. While most conventional schemes don't reflect the prediction error, our algorithm makes use of the deviation to improve the QoS quality. More specifically, it predicts traffic to be arrived at the beginning of a time slot and also measures the actual arrived traffic at the end of a time slot. The prediction deviation is derived at the beginning of a next time slot, and it is quantified to be reflected to the delay control mechanism for the next time slot. The target delay is adjusted by some extent which is determined by the prediction error at every time slot. As the suggested algorithm compensates the prediction error at every time slot, it shows the superior adaptability to the bursty traffic as well as the exponential traffic as compared with conventional approaches.

The remainder of this paper is organized as follows. In Section Ⅱ, an algorithm which provisions the quantitative differentiated services is developed. Following this, in Section Ⅲ, a set of simulation experiments to illustrate the performance of the scheme is presented. Finally, in Section Ⅳ, some concluding remarks are presented.

Ⅱ. Related Work

In DiffServ architecture, an admission control scheme is mainly used to provide QoS guarantees by reserving appropriate resource[6]. There are two basic approaches to admission control. The first, which is called parameter-based approach, computes the amount of network resources required to support a set of flows given a priori flow characteristics. The second, measurement-based approach, relies on measurement of actual traffic load in order to make admission decisions. Measurement-based approaches are classified to two schemes, envelopes-based[7][8], and probing-based[9].

In [10] and [11], the definition of a statistical bound on arriving traffic is employed to obtain the statistical multiplexing gain in a single node with a packet scheduling algorithm under the scalability constraint.

In [12], the probing rate at a receiver is used as the admission condition. The loss probability of probing packets is used as a threshold to admit or reject a flow in [13].

Relative delay differentiation is first discussed in detail in [14]. In [14], two packet schedulers that try to achieve proportional delay differentiation is presented. However, the schedulers are not ideal, in the sense that, the average delays experienced by different classes tend to deviate from the proportional model under light traffic loads.

Joint Buffer Management and Scheduling(JoBS) is suggested in [5], and provides relative and absolute per-class service differentiation for delays and loss rate. It makes predictions on the delays of backlogged traffic, and uses the predictions to update the service rate of classes and the amount of traffic to be dropped.

In [15], extended weight fair queueing(WFQ) is devised and applied to proportional delay differentiation service. It shows that the delay requirements can be achieved efficiently.

A new scheduler, Deadline Fair Sharing(DFS), is suggested in [16]. It operates in a dynamic weighted fair manner to provide an absolute delay guarantee and proportional delay and loss differentiation guarantees.

Probing mechanism which is incorporated into the EEAC-SV scheme is devised to enhance the end-to-end QoS granularity in the DiffServ network in [17].

III. Adaptive Delay Differentiation Model

1. Service Differentiation Model

It is assumed that there are N service classes, and class i+1 is better than class i, in terms of service metrics. With this convention, the service guarantees for the classes can be expressed. An absolute delay guarantee on class i is specified as

$$D_i = D_i^*, \quad \forall i \in \{1, ..., M\} \tag{1}$$

where D_i^* is a desired delay bound of class i. The proportional delay guarantee between class i and class i+1 is defined as

$$\frac{D_{i+1}}{D_i} = \alpha_i^*, \quad \forall i \in \{M+1, ..., N\} \tag{2}$$

where α_i^* is a constant that quantifies the proportional differentiation desired.

2. Node Architecture

The proposed node architecture is shown in Figure 1. The classifier classifies incoming traffic into a number of classes and the scheduler then serves traffic in class buffers. Input traffic is predicted at the beginning of the time slot and measured at the end of the time slot, and the difference will feed into a process to adjust the service rate in the scheduler periodically.

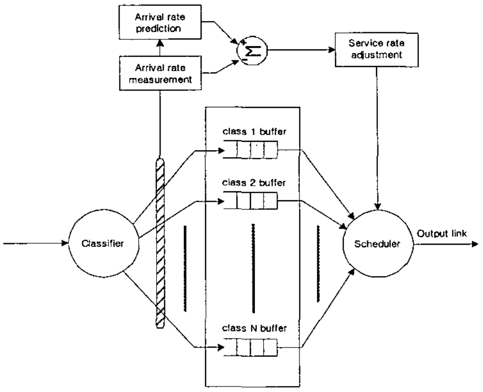


그림 1. 제안된 시스템 구조

Fig. 1. The proposed system architecture.

3. Service Rate Adjustment

가. Absolute Differentiation Service

As illustrated in Figure 2, time axis is slotted with interval T, and time slot n spans the time interval [t_{n-1}, t_n].

The input rate $\tilde{\lambda}_i(n)$ of class i for the time slot n is predicted with the weighted moving average schemes like equation (3) with $\rho = 0.9$. Specifically, predicted values are indicated by a tilde(~).

$$\tilde{\lambda}_i(n) = (1-\rho)\frac{\sum_{k=n-N-1}^{n-2} \lambda_i(k)}{N} + \rho\lambda_i(n-1) \tag{3}$$

The backlog $B_i(t)$ of class i at time t is derived

그림 2. 시간 축 표기

Fig. 2. Time axis notation.

from $R_i^{in}(t)$ and $R_i^{out}(t)$ like equation (4) where $R_i^{in}(t)$ is the arrived traffic at class i buffer and $R_i^{out}(t)$ is the serviced traffic from class i buffer in the interval [0, t] respectively.

$$B_i(t) = R_i^{out}(t) - R_i^{in}(t) \tag{4}$$

Now, some parameters related a class i are predicted to derive the service rate for the next time slot n. With the predicted input rate for the next time slot n of equation (3), the prediction of the class i input traffic for next time slot n, $\widetilde{R}_i^{in}(t;t \in [t_{n-1}, t_n])$, is given by

$$\widetilde{R}_i^{in}(t;t \in [t_{n-1}, t_n]) = \widetilde{\lambda}_i(n) \times (t - t_{n-1}). \tag{5}$$

Similarly, with the definition of service rate $\gamma_i(n)$ of class i buffer for next time slot n, the predicted serviced traffic of class buffer i for next time slot n, $\widetilde{R}_i^{out}(t;t \in [t_{n-1}, t_n])$, is given by

$$\widetilde{R}_i^{out}(t;t \in [t_{n-1}, t_n]) = \gamma_i(n) \times (t - t_{n-1}) \tag{6}$$

With the equation (5) and (6), the predicted backlog $\widetilde{B}_i(t;t \in [t_{n-1}, t_n])$ of class buffer i for next time slot n is derived as

$$\widetilde{B}_i(t;t \in [t_{n-1}, t_n]) = B_i(t_{n-1}) + \{\lambda_i(n) - \gamma_i(n)\} \times (t - t_{n-1}). \tag{7}$$

Now, the predicted delay $\widetilde{D}_i(t;t \in [t_{n-1}, t_n])$ of an class i input packet arriving at time t, $t \in [t_{n-1}, t_n]$, is described as equation (8).

$$\begin{aligned}\widetilde{D}_i(t;t \in [t_{n-1}, t_n]) &= \frac{\widetilde{B}_i(t;t \in [t_{n-1}, t_n])}{\gamma_i(n)} \\ &= \frac{B_i(t_{n-1}) + \{\widetilde{\lambda}_i(n) - \gamma_i(n)\} \times (t - t_{n-1})}{\gamma_i(n)}\end{aligned} \tag{8}$$

Averaging the instantaneous delay $\widetilde{D}_i(t)$ over a time slot n provides a simple measure for the history of delays experienced by typical class i packets. It is given by equation (9).

$$\begin{aligned}\widetilde{D}_i^{avg}(n) &= \frac{1}{T}\int_{t_{n-1}}^{t_n} \widetilde{D}_i(x)dx \\ &= \frac{1}{\gamma_i(n)}\left[B_i(t_{n-1}) + \frac{T}{2}\{\widetilde{\lambda}_i(n) - \gamma_i(n)\}\right]\end{aligned} \tag{9}$$

It is a feature of our algorithm that the prediction error on the input rates over time slot n is reflected on the derivation of the service rates over next time slot n+1. In order to reflect the prediction error on the input rates on the derivation of the service rates, the error $\Delta\lambda_i$ between the measured input rates $\lambda_i(n)$ and the predicted input rates $\widetilde{\lambda}_i(n)$ is defined as equation (10).

$$\Delta\lambda_i(n) = \lambda_i(n) - \widetilde{\lambda}_i(n) \tag{10}$$

With the definition of equation (10), the delay difference $\Delta D_{i,\Delta\lambda_i}(n)$ caused by the prediction error $\Delta\lambda_i$ on input rates is derived from equation (9) and given by equation (11).

$$\Delta D_{i,\Delta\lambda_i}(n) = \frac{T}{2} \times \frac{\Delta\lambda_i(n)}{\gamma_i(n)} \tag{11}$$

The actual averaged delays D_i^{avg} over time slot n is adjusted with that extent of equation (11) and expressed as equation (12).

$$\begin{aligned}D_i^{avg} &= \widetilde{D}_i^{avg} + \Delta D_{i,\Delta\lambda_i} \\ &= \frac{1}{\gamma_i(n)}\left[B_i(t_{n-1}) + \frac{T}{2}\{\lambda_i(n) - \gamma_i(n)\}\right]\end{aligned} \tag{12}$$

With the derivation of equation (12), the delay difference ΔD_i from the target delay D_i^* over time slot n is given by

$$\Delta D_i(n) = D_i^* - \frac{B_i(t_{n-1}) + \frac{T}{2}\{\lambda_i(n) - \gamma_i(n)\}}{\gamma_i(n)} \tag{13}$$

As equation (13) indicates the deviation from the desired delay over the time slot n, it is compensated by that extent over next time slot n+1 such as equation (14).

$$D_i(n+1) = D_i^* + \Delta D_i(n) \tag{14}$$

In case that the actual delay is two larger than the target delay, there is a possibility that $D_i(n+1)$ might become negative in equation (14). Delay cannot be negative, it is fixed at zero for the case such as equation (15).

$$D_i(n+1)=\begin{cases}0 & D_i(n+1)\leqq 0\\ 2D_i^*-\dfrac{B_i(t_{n-1})+\frac{T}{2}\{\lambda_i(n)-\gamma_i(n)\}}{\gamma_i(n)} & otherwise\end{cases}\tag{15}$$

As the target delay for the time slot n+1 is derived, the service rates for the time slot n+1 is derived from the equation (9). It is given by equation (16).

$$\gamma_i(n+1)=\begin{cases}\dfrac{B_i(t_n)+\frac{T}{2}\tilde{\lambda}_i(n+1)}{\frac{T}{2}} & D_i(n+1)\leqq 0\\ \dfrac{B_i(t_n)+\frac{T}{2}\tilde{\lambda}_i(n+1)}{2D_i^*-\dfrac{B_i(t_n)+\frac{T}{2}\lambda_i(n)-\gamma_i(n)}{\gamma_i(n)}+\dfrac{T}{2}} & otherwise\end{cases}\tag{16}$$

나. Proportional Differentiation Service

In proportional differentiation services, the delay ratio between two adjacent classes should be fixed such as (2). As (12) means the actual delay for the time slot n, the actual relative ratio on delay for the service classes i and i+1 at the time slot n is described as (17).

$$\frac{D_{i+1}^{avg}(n)}{D_i^{avg}(n)}=\frac{\dfrac{B_{i+1}(t_{n-1})+\frac{T}{2}\{\lambda_{i+1}(n)-\gamma_{i+1}(n)\}}{\gamma_{i+1}(n)}}{\dfrac{B_i(t_{n-1})+\frac{T}{2}\{\lambda_i(n)-\gamma_i(n)\}}{\gamma_i(n)}}=\alpha_i(n)\tag{17}$$

As there is the possibility that the $\alpha_i(n)$ in (17) deviates from the target value α_i^*, the difference $\Delta\alpha_i(n)=\alpha_i^*-\alpha_i(n)$ is applied to the updated target value at the next time slot such as (18) to compensate the deviation at the previous time slot n.

$$\alpha_i(n+1)=\alpha_i^*+\Delta\alpha_i(n)\tag{18}$$

The delay relation between time slot n and n+1 is given by (19) from (17) and (18).

$$\frac{D_{i+1}(n+1)}{D_i(n+1)}=K\cdot\frac{D_{i+1}^{avg}(n)}{D_{i+1}^{avg}(n)},\ K=\frac{2\alpha_i^*}{\alpha_i(n)}-1\tag{19}$$

As it is possible to notate K in (19) in a fraction form $\frac{y}{x}$, (19) is expressed as (20).

$$\begin{aligned}D_{i+1}(n+1)&=x\cdot D_{i+1}^{avg}(n)\\ D_i(n+1)&=y\cdot D_i^{avg}(n)\end{aligned}\tag{20}$$

To determine the service rate for the time slot n+1, the case of $\Delta\alpha_i(n)>0$ is considered in the first place. As this case indicates the situation that delay of higher class has been rather shortened and/or that of lower class lengthened, the delay of higher class and that of lower class for the next time slot should be decreased and increased respectively. That means x and y should be x≤1 and y≥1. The possible areas for x and y are illustrated in Fig. 3. In this paper, we picked the corner point of the feasible area for the unique value of x and y since it reduces the delay unless the service rate cannot support the rate.

Next, the case of $\Delta\alpha_i(n)<0$ is considered. There

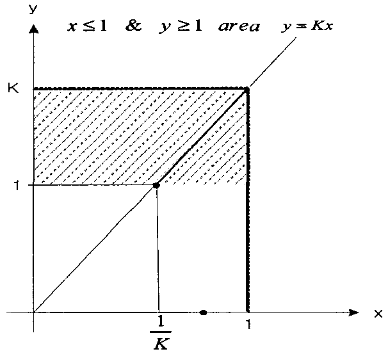


그림 3. $\Delta\alpha_i(n)>0$ 의 경우에 x 와 y 값 결정
Fig. 3. Determination of x and y for the case of $\Delta\alpha_i(n)>0$.

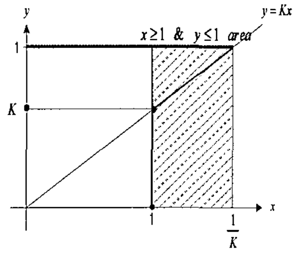


그림 4. $\Delta\alpha_i(n)<0$ 과 $|\Delta\alpha_i(n)|<\alpha^{*i}$의 경우에 x 와 y 값 결정
Fig. 4. Determination of x and y for the case of $\Delta\alpha_i(n)<0$ & $|\Delta\alpha_i(n)|<\alpha_i^*$.

are two possible ways that satisfy the condition. As the delay ratio should not be negative, two cases, $|\Delta\alpha_i(n)| < \alpha_i^*$ and $|\Delta\alpha_i(n)| \geq \alpha_i^*$, have different solving procedures. First, the case of $|\Delta\alpha_i(n)| < \alpha_i^*$ where delay ratio is not negative is touched. Applying the same logic as the case of $\Delta\alpha_i(n) > 0$, the area of x and y is x >1 and y < 1. The value of x and y is determined to be a corner point from possible values shown in Fig. 4.

Finally, the case of $|\Delta\alpha_i(n)| \geq \alpha_i^*$ is considered. The interpretation of this case is that the higher class has much higher delay than lower delay. Since K becomes negative in this condition, x and y should be negative and positive respectively. Then K can be recalculated as $K = \frac{x}{x-y}$, and K always satisfies the bound K < 1. The next step is the same as in the case of $|\Delta\alpha_i(n)| < \alpha_i^*$.

This description is summarized at (21).

$$
\begin{array}{l}
(case1)\ \Delta\alpha_i(n) > 0 \\
\qquad D_{i+1}(n+1) = D_{i+1}(n) \\
\qquad D_i(n+1) = K \cdot D_i(n) \\
(case2)\ \Delta\alpha_i(n) < 0 \\
\quad (case2-1)\ |\Delta\alpha_i(n)| < \alpha_i^* \\
\qquad\qquad D_{i+1}(n+1) = K \cdot D_{i+1}(n) \\
\qquad\qquad D_i(n+1) = D_i(n) \\
\quad (case2-2)\ |\Delta\alpha_i(n)| \geq \alpha_i^* \\
\qquad\qquad K = \frac{x}{x-y},\ x < 0, y > 0 \\
\qquad\qquad D_{i+1}(n+1) = K \cdot D_{i+1}(n) \\
\qquad\qquad D_i(n+1) = D_i(n)
\end{array}
\qquad (21)
$$

With the values $D_{i+1}(n+1)$ and $D_i(n+1)$, we can derive the service rate which is given by (22).

$$\gamma_k(n+1) = \frac{B_k(t_n) + \frac{T}{2}\tilde{\lambda}_k(n+1)}{D_k(n+1) + \frac{T}{2}},\ k = i, i+1 \quad (22)$$

Ⅳ. Evaluation

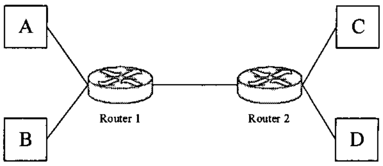


그림 5. 모의실험용 네트워크 형상
Fig. 5. Simulated network topology.

Simulations for the examination of efficiency and comparisons between three proposed algorithms and the conventional algorithm have been conducted in this section with OPNET simulator. We fix the time period T to 0.1s. The value N is set to 5. The delay scale is set to seconds for all simulations. Since the main difference of our algorithm is the adaptability of the traffic prediction error, we call our algorithm as adaptive algorithm and the conventional algorithm as non-adaptive scheme. We simulate our algorithm and non-adaptive algorithm with the simple network topology illustrated in Fig. 5. Each source node generates number of traffic flows whose time inter-arrival and packet size are exponentially distributed with mean 0.001 seconds and 1000 bits. We create two absolute service classes 1 and 2, and two proportional classes 3 and 4 in node A and B. The delay requirements are set to 20ms and 40ms respectively for the absolute delay, and $\alpha_3^* = 0.5$ for the proportional delay. Traffic load distribution is set to 30%, 20%, 30%, and 20% respectively. Link capacity is set to 100Mbps and the link propagation delay is assumed to be negligible.

1. Absolute Delay Evaluation

Fig. 6 shows a result of queueing delay of class 1 for an exponential traffic. It shows that the adaptive bound.

Since current Internet traffic is not exponential, more realistic traffic that reflects bursty characteristic needs to be considered. For realistic traffic, we create hundreds of flows which follow Pareto distribution with shaper value 1.9. In addition, the duration of flows follows Pareto distribution with shaper value of 1.9. The example of the input traffic is shown as Fig. 7.

We simulate two algorithms using the traffic and

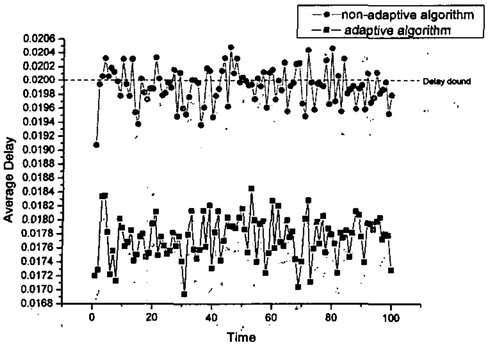


그림 6. 지수 트래픽에 대한 클래스 1 지연
Fig. 6 Delay of class 1 with exponential traffic load.

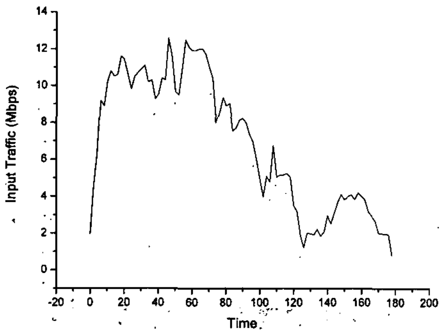


그림 7. 버스트 입력 트래픽
Fig. 7. Bursty input traffic.

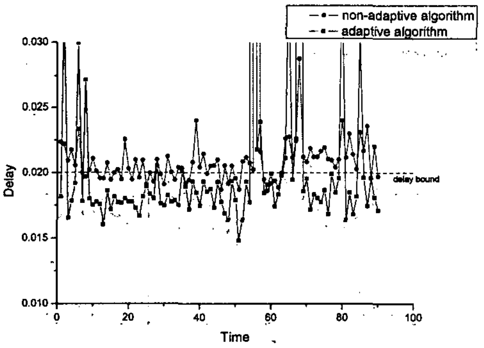


그림 8. 버스트 트래픽에 대한 절대적 지연
Fig. 8. Absolute delay to the bursty traffic.

results are shown as Fig. 8. In this scenario, most of average delay does not meet the delay boundary of 20ms for non-adaptive algorithm while most of them meet the bound in our algorithm. This superiority to the bursty traffic is anticipated since our algorithm operates to compensate the deviation caused by the bursty characteristic of the traffic continuously. scheme clearly meets the delay bound while non-adaptive scheme frequently exceeds the delay.

2. Proportional Delay Evaluation

Another observation is for the proportional delay which is shown in Fig. 9. Non-adaptive algorithm provides more precise delay ratio which is almost 0.50 than our algorithm where it is 0.56. However, the delay of each class is fixed in non-adaptive scheme while it is dynamically varied in adaptive algorithm. This feature relates to an effectiveness of a scheduler. That is, our scheme tries to maintain the target delay ratio with the delay of each class varied according to the availability of service link.

Our algorithm also shows better performance when using traffic with Pareto distribution. We set the same traffic generation parameters that are used in the absolute delay and its result is shown as Fig. 10.

In this simulation, adaptive scheme also shows overall improvement over the non-adaptive algorithm. That means that our proportional differentiation algorithm which is based on the continual compensation of the error caused by the bursty characteristic of the traffic shows better adaptability to the real traffic environment.

A simulation for delay ratio is performed when traffic load is different. Most of the work-conserving scheduler cannot meet the proportional delay differentiation when the traffic load is low [16]. Authors in [16] explains it is because their algorithms assume that the queue buffer is always

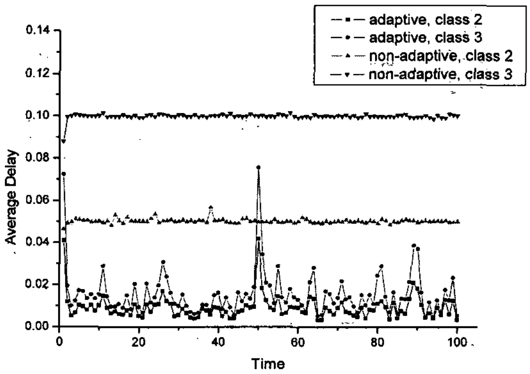
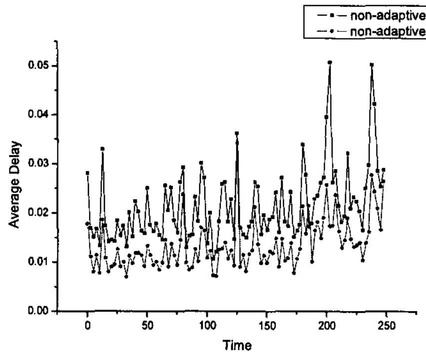
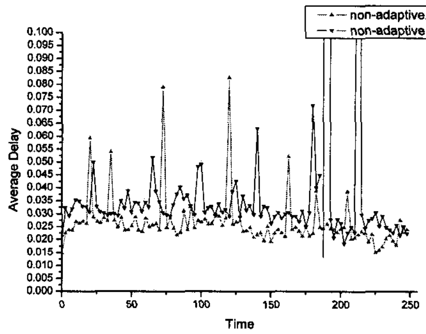


그림 9. 상대적 서비스에 대한 큐잉 지연
Fig. 9. The queueing delay for the proportional service.



(a) Adaptive scheme



(b) Non-adaptive scheme

그림 10. 버스트 트래픽에 대한 상대적 지연
Fig. 10. Proportional delay to bursty traffic.

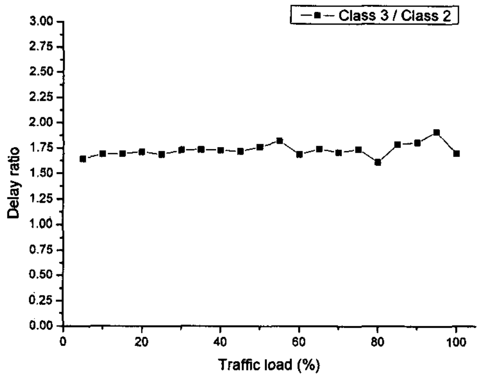


그림 11. 입력 부하에 대한 절대적 지연의 의존정
Fig. 11. The dependency of proportional delay on input load.

saturated. However, there exists more possibility of queue buffers not being empty when traffic load is low. Our algorithm solvesthis problem by dynamically allocating the service rate and adapting the prediction error though it is based on the work-conserving scheduler. In Fig. 11, it is shown that the delay ratio does not depend on the traffic load, although accuracy of the delay ratio is a little worse than other algorithms.

V. Conclusion

In this paper, a delay differentiation algorithm that achieves proportional and absolute QoS provisioning is proposed. The main feature of this algorithm is that it continually adjusts the target delay with reference to the traffic prediction deviation in previous time section.

It has founded that the suggested scheme performs well in terms of achieving proportional and absolute QoS provisioning. In addition, it shows superior adaptability to the traffic fluctuation in comparison with conventional approach, and it presents a feasible approach to future Internet where QoS differentiation is essentially required and bursty traffic is prevailed.

References

[1] S. Shenker, and J. Wroclawski, "General Characterization Parameters for Integrated Service Network Elements," IETF RFC 2215, September 1997.
[2] S. Blake, D. Black, M, Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated service," IETF RFC 2475, December 1998.
[3] I. Stoica and H. Zhang, "Providing Guaranteed Services Without Per Flow Management," ACM SIGCOMM '99, Cambridge, MA, September 1999.
[4] C. Dovrolis, D. Stiliadis and P. Ramanathan, "Proportional Differentiated Services: Delay Differentiation and Packet Scheduling," ACM SIGCOMM'99, Cambridge, MA, September 1999.
[5] L. Liebeherr and N. Christin, "Buffer Management and Scheduling for Enhanced Differentiated Service," Technical Report: University of Virginia, CS-2000-2
[6] C. Cetinkaya and E. Knightly, "Egress admission control," Proc. IEEE INFOCOMM 2000, vol.3, no. 1, pp.1471-1480, Tel Aviv, Israel, March 2000.
[7] C. Cetinkaya, V. Kanodia, and E. Knightly, "Scalable Services via Egress Admission Control," IEEE TRANSACTION ON MULTIMEDIA

Special Issue on Multimedia over IP, vol. 3, no. 1, March 2001.

[8] J. Qiu and E. Knightly, "Measurement-based admission control with aggregate traffic envelopes," Proc. 10th IEEE ITWDC'98, vol.9, no.2, pp.199-210, Ischia, Italy, Sept. 1998.

[9] L. Breslau, E. Knightly, S, Schenker, I. Stoica, and H. Zhang, "Endpoint admission control: Architectural issue and performance," ACM SIGCOMM 2000, vol.30, no.4, pp.57-69, Stockholm, Sweden, Aug. 2000.

[10] R. Boorstyn, A. Burchard, J. Liebeherr, and C. Oottamakorn, "Statistical service envelopes : Statistical bounds on multiplexed traffic in packet network," Proc. IEEE INFOCOMM 2000, vol.3, no. 1, pp.1223-1232, Tel Aviv, Israel, March 2000.

[11] R. Boorstyn, A. Burchard, J. Liebeherr, and C. Oottamakorn, "Statistical service assurances for traffic scheduling algorithm," IEEE. J. Sel. Areas Commun., vol.18, no.12, pp.2651-2664, Dec. 2000.

[12] G. Bianchi, A. Capone, and C. Petrioli, "Throughput analysis of end-to-end measurement-based admission control in IP," Proc. IEEE INFOCOM 2000, vol. 3, no. 1, pp.1461-1470, Tel Aviv, Israel, March 2000.

[13] V. Elek, G. Karlsson, and R. Ronngren, "Admission control based on end-to-end measurements," Proc. IEEE INFOCOM 2000, vol. 2, no. 1, pp.623-630, Tel Aviv, Israel, March 2000.

[14] C. Dovrolis, D. Stiliadis and P. Ramanathan, "Proportional Differentiated Services: Delay Differentiation and Packet Scheduling," ACM SIGCOMM '99, Cambridge, MA, September 1999.

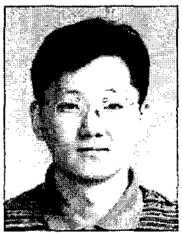
[15] Chin-Chang Li, Shiao-Li Tsao, Meng Cheng Chen, Yeali Sun, and Yueh-Min Huang, "Proportional Delay Differentiation Service Based on Weighted Fair Queueing," 2000 IEEE ICCCN, p418-423, Oct. 2000.

[16] Yuan-Chang and Chang.A, "A non-work conserving scheduler to provide proportional delay differentiated services," GLOBECOM, vol.3, p1723-1727, 29 Nov.-3 Dec. 2004.

저 자 소 개

백 정 훈(정회원)
1986년 한양대학교 전자공학과 학사 졸업.
1988년 한양대학교 전자공학과 석사 졸업.
1999년 한양대학교 전자공학과 박사 졸업.
2002년～현재 동아방송대학 정보통신과 교수
<주관심분야 : QoS, 스위치, 전송>

이 유 경(정회원)
1978년 항공대학교 전자공학과 학사 졸업.
1980년 연세대학교 전자공학과 석사 졸업.
1984년～현재 한국전자통신연구원 책임연구원.
<주관심분야 : 캐리어 클래스 이더넷, 네트워크, 스위치>



박 재 우(정회원)
1993년 인하대학교 전기공학과 학사
1995년 한국과학기술원 전기 및 전자공학과 석사
1995년～1999년 삼성중공업
2000년～현재 한국전자통신연구원 선임연구원.
<주관심분야 : 인터넷, 이동통신 네트워크>