차세대 네트워크에서 상대적 지연 차별화를 위한 적응형 입력 트래픽 예측 방식

백 정 훈*

요 약

본 논문에서는 차세대 패킷 네트워크에서의 서비스 품질 기능 고도화를 목적으로 상대적 지연 차별화 기능을 제공하는 알고리듬을 제시하고 시뮬레이션을 통해 성능 분석을 수행한다. 또한, 제안된 알고리듬을 XPC 860 CPU 기반의 시험 보드상에서 VHDL로 구현하여 실제 트래픽 입력 상황하에서의 성능 분석을 수행한다. 제안된 알고리듬은 매 시간 구간마다 입력되는 트래픽을 측정하고 이를 기반으로 다음 시간 구간 동안 입력될 트래픽의 양을 예측한 후 실제로 다음 시간 구간 동안에 입력된 트래픽과 비교하여 오차분을 도출하여 이를 다음 타임 슬롯의 지연 차별화 동작에 지속적으로 반영하는 것이 특징적 요소이므로 오차분을 고려하지 않는 기존 방식에 비해 버스트 트래픽에 대하여 우수한 적응성을 보여준다. 제안된 방식의 성능은 시뮬레이션과 실제 보드상에서의 시험을 통해 절대적 지연 목표를 충족시킴과 동시에 기존 방식에 비해 버스트 트래픽에 대하여 성능 개선 효과가 달성됨이 확인된다.

Adaptive Input Traffic Prediction Scheme for Proportional Delay Differentiation in Next-Generation Networks

Jung Hoon Paik*

ABSTRACT

In this paper, an algorithm that provisions proportional differentiation of packet delays is proposed with an objective for enhancing quality of service (QoS) in future packet networks. It features an adaptive scheme that adjusts the target delay every time slot to compensate the deviation from the target delay which is caused by the prediction error on the traffic to be arrived in the next time slot. 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. The difference between them 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 rate traffic. It is demonstrated through simulations that the algorithm meets the quantitative delay bounds and shows superiority to the traffic fluctuation in comparison with the conventional non-adaptive mechanism. The algorithm is implemented with VHDL on a Xilinx Spartan XC3S1500 FPGA and the performance is verified under the test board based on the XPC860P CPU.

Key words : Proportional Delay, Quality of Service

* 동아방송예술대학 방송통신학과 교수

1. Introduction

Two broad paradigms for quality-of-service (QoS) 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 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 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 applied 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 con-

tinually compensates the prediction error every time slot, it shows the superior adaptability to the bursty traffic as well as the constant rate traffic as compared with conventional approaches.

The remainder of this paper is organized as follows. In Section 2, related work is overviewed. In Section 3, an algorithm which provisions the quantitative differentiated services is developed. Following this, in Section 4, a set of simulation experiments to illustrate the performance of the scheme is presented. Section 5 handles the contents relating to the implementation of the algorithm. Finally, in Section 6, some concluding remarks are presented

2. Adaptive Delay Differentiation Model

2.1 Objective

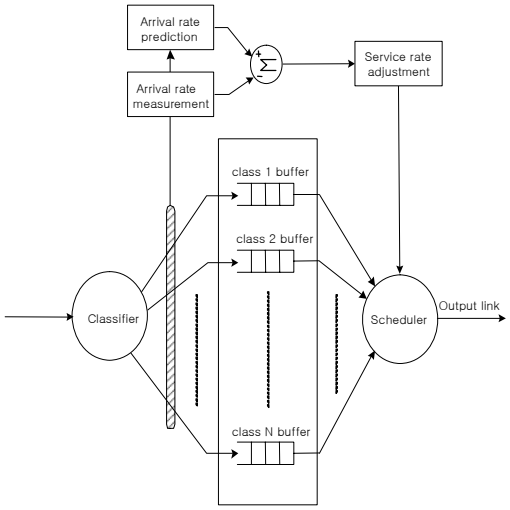
It is assumed that there are N service classes, and class i + 1 is better than class i for $2 \le i \le N$, in terms of service metrics. With this convention, the service guarantees for the classes can be expressed. An absolute delay guarantee on class i is specified as

$$D_i = D_i^*, \qquad \forall\, i \in \{1, \cdots, M\} \tag{1}$$

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

2.2 Node Architecture

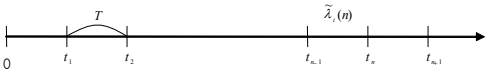
The proposed node architecture is shown in (Figure 1). The classifier classifies incoming traffic into a number of classes and the scheduler then serves traffic in class buffers. Input traffic is predicted at the beginning of the time slot and measured at the end of the time slot, and the difference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



(Figure 1) The proposed system architecture

2.3 Service Rate Adjustment

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



(Figure 2) Time axis notation

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 (2) with $\rho = 0.9$. Specifically, predicted values are indicated by a tilde(~).

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

The backlog $B_i(t)$ of class i at time t is derived from $R_i^{in}(t)$ and $R_i^{out}(t)$ like equation (3) 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.

$$B_i(t) = R_i^{in}(t) - R_i^{out}(t) \quad (3)$$

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 (2), 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

$$\widetilde{R_i^{in}}(t; t \in [t_{n-1}, t_n]) = \tilde{\lambda}_i(n) \times (t - t_{n-1}) \quad (4)$$

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, $\tilde{R}_i^{in}(t; t \in [t_{n-1}, t_n])$, is given by

$$\widetilde{R_i^{out}}(t; t \in [t_{n-1}, t_n]) = \gamma_i(n) \times (t - t_{n-1}) \quad (5)$$

With the equation (4) and equation (5), 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 equation (6).

$$\widetilde{B}_i(t; t \in [t_{n-1}, t_n]) = B_i(t_{n-1}) + \{\lambda_i(n) - \gamma_i(n)\} \times (t - t_{n-1}) \quad (6)$$

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 (7).

$$\tilde{D}_i(t; t \in [t_{n-1}, t_n]) = \frac{\widetilde{B}_i(t; t \in [t_{n-1}, t_n])}{\gamma_i(n)} = \frac{B_i(t_{n-1}) + \{\tilde{\lambda}_i(n) - \gamma_i(n)\} \times (t - t_{n-1})}{\gamma_i(n)} \quad (7)$$

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 (8).

$$\tilde{D}_i^{avg}(n) = \frac{1}{T}\int_{t_{n-1}}^{t_n} \tilde{D}_i(x)dx = \frac{1}{\gamma_i(n)}\left[B_i(t_{n-1}) + \frac{T}{2}\{\tilde{\lambda}_i(n) - \gamma_i(n)\}\right] \quad (8)$$

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 $\tilde{\lambda}_i(n)$ is defined as equation (9).

$$\Delta\lambda_i = \lambda_i(n) - \tilde{\lambda}_i(n) \quad (9)$$

With the definition of equation (9), 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 (8) and given by equation (10).

$$\Delta D_{i,\Delta\lambda_i}(n) = \frac{T}{2} \times \frac{\Delta\lambda_i(n)}{\gamma_i(n)} \quad (10)$$

The actual averaged delays D_i^{avg} over time slot n is adjusted with that extent of equation (10) and expressed as equation (11).

$$D_i^{avg} = \widetilde{D}_i^{avg} + \Delta D_{i,\Delta\lambda_i} \quad (11)$$

$$= \frac{1}{\gamma_i(n)}\left[B_i(t_{n-1}) + \frac{T}{2}\{\widetilde{\lambda_i}(n) + \Delta\lambda_i(n) - \gamma_i(n)\}\right]$$

In proportional differentiated services, the delay ratio between two adjacent classes should be fixed such that equation (1) is satisfied. As equation (11) indicates the actual average delay for the time slot n, the actual relative ratio on average delay for the service classes i and i+1 at the time slot n is described as equation (12).

$$\frac{D_{i+1}^{avg}(n)}{D_i^{avg}(n)} = \frac{\frac{1}{\gamma_{i+1}(n)}\left\{B_{i+1}(t_{n-1}) + \frac{T}{2}[\widetilde{\lambda_{i+1}}(n) - \Delta\lambda_{i+1}(n) - \gamma_{i+1}(n)]\right\}}{\frac{1}{\gamma_i(n)}\left\{B_i(t_{n-1}) + \frac{T}{2}[\widetilde{\lambda_i}(n) - \Delta\lambda_i(n) - \gamma_i(n)]\right\}} = \alpha_i(n) \quad (12)$$

As there is the possibility that $\alpha_i(n)$ in equation (12) deviates from the target value α_i^*, the delay ratio difference $\Delta\alpha_i(n) = \alpha_i^* - \alpha_i(n)$ is applied to the updated target value at the next time slot such as equation (13) to compensate the deviation at the previous time slot n.

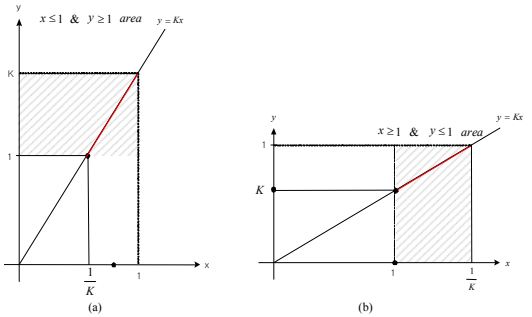
$$\alpha_i(n+1) = \alpha_i^* + \Delta\alpha_i(n) \quad (13)$$

The delay relation between time slot n and n+1 is given by equation (14) from equation (12) and equation (13).

$$\frac{D_{i+1}(n+1)}{D_i(n+1)} = K \cdot \frac{D_{i+1}^{avg}(n)}{D_{i+1}^{avg}(n)}, \quad K = \frac{2\alpha_i^*}{\alpha_i(n)} - 1 \quad (14)$$

As it is possible to convert K in equation (18) into a fraction form, $\frac{y}{x}$, equation (14) is expressed as equation (15).

$$D_{i+1}(n+1) = y \cdot D_i^{avg}(n) \quad (15)$$
$$D_i(n+1) = x \cdot D_i^{avg}(n)$$



(Figure 3) Determination of x and y for K

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 the 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, i.e., x and y should be $0 \le x \le 1$ and $1 \le y \le K$. The possible areas for x and y are illustrated in (Figure 3) (a) We pick x and y values as the corner point of the feasible area that can reduce the delay of the class whose relative delay is higher than that of the other class.

Next, the case of $\Delta\alpha_i(n) < 0$ is considered. There 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 $|\Delta\alpha_i(n)| \ge \alpha_i^*$, 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 in the case of $\Delta\alpha_i(n) > 0$, the area of x and y is $1 \le x \le 1/K$ and $0 \le y \le 1$. The values of x and y are determined to be a corner point from possible values shown in (Fi-

gure 3) (b).

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 class. Since K becomes negative in this condition, x and y are negative and positive respectively. Without violating the relative ratio, K can be recalculated as $K = \frac{x}{x-y}$, and K always satisfies the bound K < 1. The other steps are the same as in the case of $|\Delta\alpha_i(n)| < \alpha_i^*$.

Descriptions of all cases are summarized at equation (16).

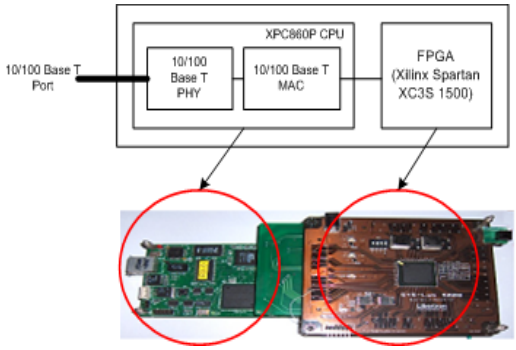
$$
\begin{aligned}
&(case\,1)\ \Delta\alpha_i(n) > 0 \\
&\qquad D_{i+1}(n+1) = D_{i+1}(n) \\
&\qquad D_i(n+1) = K \cdot D_i(n) \\
&(case\,2)\ \Delta\alpha_i(n) < 0 \\
&\quad (case\,2-1)\ |\Delta\alpha_i(n)| < \alpha_i^* \\
&\qquad D_{i+1}(n+1) = K \cdot D_{i+1}(n) \\
&\qquad D_i(n+1) = D_i(n) \\
&\quad (case\,2-2)\ |\Delta\alpha_i(n)| \geq \alpha_i^* \\
&\qquad K = \frac{x}{x-y},\ x<0,\ y>0 \\
&\qquad D_{i+1}(n+1) = K \cdot D_{i+1}(n) \\
&\qquad D_i(n+1) = D_i(n)
\end{aligned}
\tag{16}
$$

With the values $D_{i+1}(n+1)$ and $D_i(n+1)$, we can derive the service rate which is given by equation (17).

$$
\gamma_k(n+1) = \frac{B_k(t_n) + \frac{T}{2}\tilde{\lambda}_k(n+1)}{D_k(n+1) + \frac{T}{2}},\quad k = i,\ i+1 \tag{17}
$$

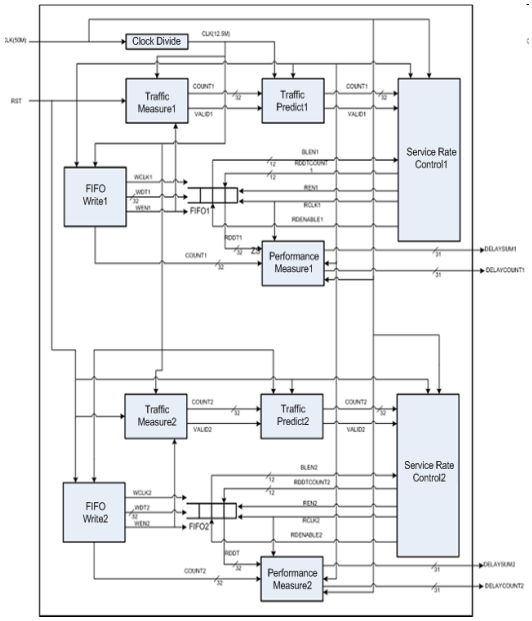
3. Implementation

The suggested algorithm is implemented with VHDL on Xilinx Spartan XC3S1500 FPGA on the test board shown in (Figure 4). It is mainly composed of 10/100 Base T PHY/MAC and XPC860P CPU.

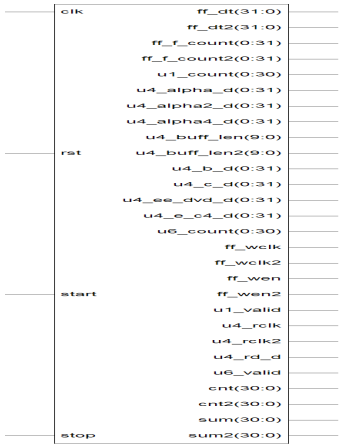


(Figure 4) The test board for implementing the suggested algorithm

(Figure 5) shows the internal blocks for the proportional delay differentiation algorithm. As it handles only two classes, there are two FIFOs. Clock Divide block divides 50 [MHz] clock to generate 12.5 [MHz] clock for accommodate the 100 [Mbps] Ethernet signal with 8bits operations. FIFO Write block accounts for generating the Ethernet data and inserting it to the FIFO. Traffic Measure block constantly measures the input traffic rate and sends the measured traffic value to Traffic Predict block. Traffic Predict block predicts the traffic amount which will be arriving for the next time slot and the predicted one is sent to Service Rate Control block. Service Rate Control block derives the service rate to FIFOs from the delay differentiation algorithm which is suggested in chapter II. Lastly, Performance Measure block calculates the average delay from measuring the difference between the arrival time and the service time of the traffic.



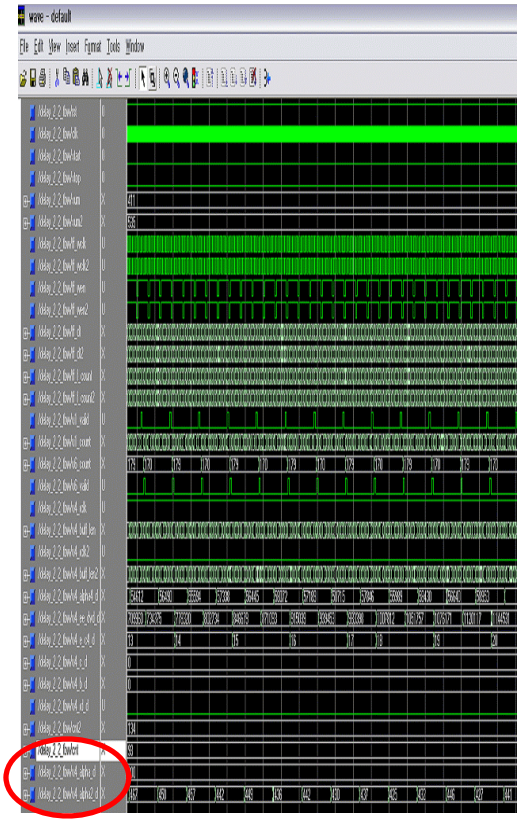
(Figure 5) The internal blocks for the synthesized circuit



(Figure 6) The synthesized circuit for the suggested algorithm

The synthesized circuit is shown in (Figure 6) where signals alpha_d and alpha_2d represent the delay of class 1 and class 2 respectively. Therefore the relative delay ratio for two classes is given as a ratio of alpha_d and alpha_2d.

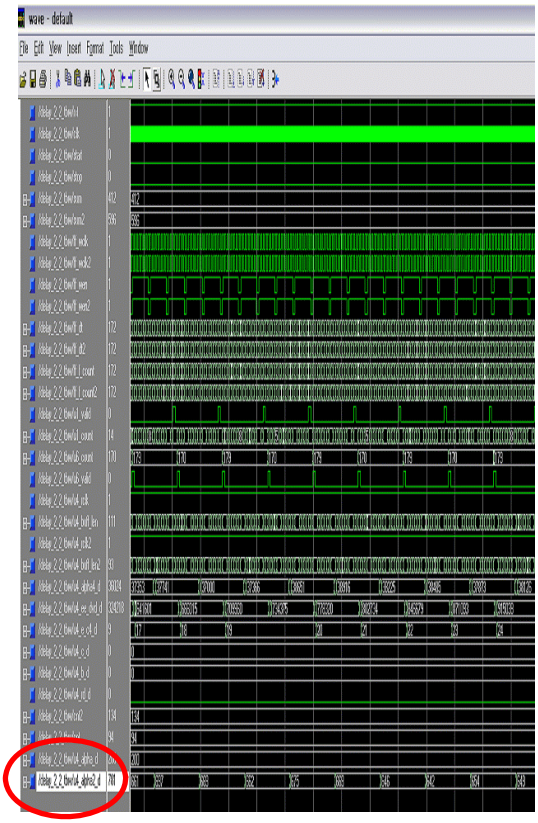
(Figure 7) and (Figure 8) are the post route simulation results for the target delay ratio of 2 and 3 respectively. Referencing two signals, alpha_d and alpha_2d, at the bottom of them show that the delay ratio is about 2.2 and 3.2 respectively. So, our proportional delay differentiation algorithm meets the target delay.



(Figure 7) Post route simulation result (target delay ratio = 2 : 1)

4. Conclusion

In this paper, a delay differentiation algorithm



(Figure 8) Post route simulation result (target delay ratio = 3 : 1)

that achieves proportional 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, Tel Aviv, Israel, Vol. 3, No. 1, pp. 1471-1480, March 2000.

[7] C. Cetinkaya, V. Kanodia, and E. Knightly, "Scalable Services via Egress Admission Control", IEEE TRANSACTION ON MULTIMED IA 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, Ischia, Italy, Vol. 9, No. 2, pp. 199-210, Sept. 1998.

[9] L. Breslau, E. Knightly, S, Schenker, I. Stoi-

ca, and H. Zhang, “Endpoint admission control : Architectural issue and performance”, ACM SIGCOMM 2000, Stockholm, Sweden, Vol. 30, No. 4, pp. 57-69, 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, Tel Aviv, Israel, Vol. 3, No. 1, pp. 1223-1232, 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.

백 정 훈

1986년 한양대학교 전자공학과 (공학사)

1988년 한양대학교 전자공학과 (공학석사)

1999년 한양대학교 전자공학과 (공학박사)

2002년~현재 동아방송예술대학 방송통신학과 교수