논문 2008-45IE-1-3

차세대 네트워크에서의 절대적 지연 차별화 기능 구현

(Implementation of Absolute Delay Differentiation Scheme in Next-Generation Networks)

백 정 훈*, 김 대 업**, 주 범 순**

(Jung Hoon Paik, Dae Ub Kim, and Bheom Soon Joo)

Abstract

In this paper, an algorithm that provisions absolute differentiation of packet delays is proposed, simulated, and implemented with VHDL on XPC 860 CPU based test board 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 both simulation and the real traffic test on the board that the algorithm meets the quantitative delay bounds and shows superiority to the traffic fluctuation in comparison with the conventional non-adaptive mechanism

요 약

본 논문에서는 차세대 패킷 네트워크에서의 서비스 품질 기능 고도화를 목적으로 절대적 지연 차별화 기능을 제공하는 알고리듬을 제시하고 시뮬레이션을 통해 성능 분석을 수행한다. 또한, 제안된 알고리듬을 XPC 860 CPU 기반의 시험 보드상에서 VHDL 로 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.

Keywords : QoS, Absolute, Delay, Differentiation, DiffServ

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

* 정회원, 동아방송예술대학
(Department of Broadcasting Communication, DongAh Institute of Media and Arts)
** 정회원, 한국전자통신연구원 방송통신융합연구부분 네트워크 연구부
(ETRI)
접수일자: 2007년8월3일, 수정완료일: 2008년2월29일

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 absolute differentiation of packet delays is proposed and the implementation scheme for it is presented. 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 Ⅲ, an algorithm which provisions the quantitative differentiated services is developed. Following this, in Section Ⅳ, a set of simulation experiments to illustrate the performance of the scheme is presented. In Section Ⅴ, the contents relating to the implementation of the algorithm and the performance of it under the implementation are given. Finally, in Section Ⅵ, 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

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

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.

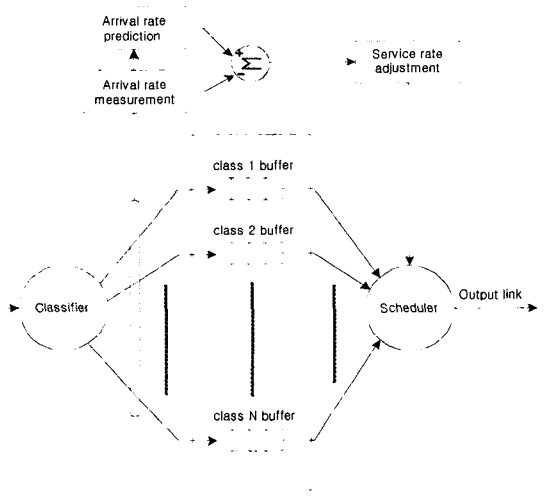


그림 1. 제안된 시스템 구조.
Fig. 1. The proposed system architecture.

3. Service Rate Adjustment

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 $\bar{\lambda}_i(n)$ of class i for the time slot n is predicted with the weighted moving average schemes like equation (3) with $\rho = 0.9$. Specifically,

그림 2. 시간 축 표기
Fig. 2. Time axis notation.

predicted values are indicated by a tilde(~).

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

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 (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.

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

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

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

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

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

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

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

Since the backlog is always positive, the service rate is constrained by

$$\gamma_i(n) \le \tilde{\lambda}_i(n) + \frac{B_i(t_{n-1})}{t - t_{n-1}} \quad (8)$$

When the variable, $t-t_{n-1}$, is T, the maximum value of $\gamma_i(n)$ is obtained for the interval $[t_{n-1}, t_n]$. Therefore, the upper bound of the service rate constrained by the backlog is given by

$$\gamma_{i,\max,backlog}(n) = \tilde{\lambda}_i(n) + \frac{B_i(t_{n-1})}{T}. \quad (9)$$

The predicted delay $\tilde{D}_i(t;t \in [t_{n-1}, t_n])$ of a class i input packet arriving to the buffer i at time t, $t \in [t_{n-1}, t_n]$, is described as

$$\tilde{D}_i(t;t \in [t_{n-1}, t_n]) = \frac{\tilde{B}_i(t;t \in [t_{n-1}, t_n])}{\gamma_i(n)} + d_i(t) = \frac{B_i(t_{n-1}) + (\tilde{\lambda}_i(n) - \gamma_i(n)) \times (t - t_{n-1})}{\gamma_i(n)} + d_i(t). \quad (10)$$

In equation (10), $d_i(t)$ is the residual service time of the packet being serviced when a class i packet is arrived to the buffer at time t and is upper bounded by L/C where L is the maximum packet length and C is the output link capacity. The maximum predicted delay at time slot n, $\tilde{D}_{i,\max}(n)$, is obtained when the variable, $t-t_{n-1}$, is T .

$$\tilde{D}_{i,\max}(n) = \frac{1}{\gamma_i(n)}\{B_i(t_{n-1}) + T[\tilde{\lambda}_i(n) - \gamma_i(n)]\} + \frac{L}{C}. \quad (11)$$

Now with $\tilde{D}_{i,\max}(n) \le D_i^*$ for the absolute delay guarantee, and in (11), the lower bound for the service rate is obtained as

$$\gamma_i(n) \ge \frac{B_i(t_{n-1}) + T\tilde{\lambda}_i(n)}{D_i^* + T - \frac{L}{C}}. \quad (12)$$

As the service rate is always positive, the condition for T is derived as

$$T > \frac{L}{C} - D_i^*. \quad (13)$$

The predicted delay in equation (10) is for the fixed service rate whose value is determined at the beginning of time slot n and the predicted input rate for time slot n. In order to make the delay less than the desired absolute delay D_i^*, the lower bound of service rate is obtained. As the same class packets require the same amount of absolute delay but the

input rate may vary, the algorithm allows service rate to be changed at every time slot. The resulting delay can be bigger than any number of time slots, but less than the desired absolute delay.

As we previously mentioned, it is a feature of our algorithm that the prediction error on the input rates over current time slot is reflected on the derivation of the service rates for the next time slot. That is, as the actual input rates can be calculated at the end of time slot n,the prediction error on delay can be derived. With the derived delay error on current time slot, the target delay on next time slot is changed to reflect the error. In order to reflect the prediction error on the derivation of the service rates, the error $\Delta\lambda_i(n)$ between the measured input rates $\lambda_i(n)$ and the predicted input rates $\tilde{\lambda}_i(n)$ is defined as

$$\Delta\lambda_i(n) = \lambda_i(n) - \tilde{\lambda}_i(n). \tag{14}$$

With the definition of (14), the delay difference $\Delta D_{i,\Delta\lambda_i}(n)$ caused by the prediction error $\Delta\lambda_i$ on input rates is derived from (11), and given by

$$\Delta D_{i,\Delta\lambda_i}(n) = T\frac{\Delta\lambda_i(n)}{\gamma_i(n)}. \tag{15}$$

The actual maximum delays $D_{i,\max}(n)$ over time slot n is adjusted with that extent of (15), and expressed as

$$\begin{aligned} D_{i,\max}(n) &= \left[\tilde{D}_{i,\max}(n) + \Delta D_{i,\Delta\lambda}(n)\right]^+ \\ &= \left[\frac{1}{\gamma_i(n)}\left\{B_i(t_{n-1}) + T\left[\tilde{\lambda}_i(n) + \Delta\lambda_i(n) - \gamma_i(n)\right]\right\} + \frac{L}{C}\right]^+ \end{aligned} \tag{16}$$

where $[z]^+ = \max\{z,0\}$. As one of our objective is to find the appropriate value of service rate γ_i so that $D_{i,\max}(n) \le D_i^*$, the service rates for the time slot n+1 is derived from (16), and it is as follows:

$$\gamma_i(n+1) \ge \frac{B_i(t_n) + T\left\{\tilde{\lambda}_i(n+1) + \Delta\lambda_i(n)\right\}}{D_i^* + T - \frac{L}{C}}. \tag{17}$$

The service rate that can be allocated to class i is upper bounded by the output link capacity minus the minimum service rates needed by the other classes, that is,

$$\gamma_{i,\max,capacity}(n+1) = C - \sum_{j\neq i}\gamma_{j,\min}(n+1) \tag{18}$$

Therefore, the maximum service rate is given in (19).

$$\gamma_{i,\max}(n+1) = \min\left\{\gamma_{i,\max,capacity}(n+1), \gamma_{i,\max,back\log}(n+1)\right\} \tag{19}$$

Therefore, the service rate can take any value $\gamma_i(n+1)$ with $\gamma_{i,\min}(n+1) \le \gamma_i(n+1) \le \gamma_{i,\max}(n+1)$ subject to the constraint $\sum_i \gamma_i(n+1) \le C$.

IV. Simulation

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. 3. 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

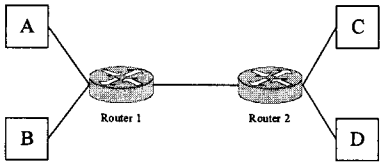


그림 3. 모의 실험용 네트워크 형상
Fig. 3. Simulated network topology.

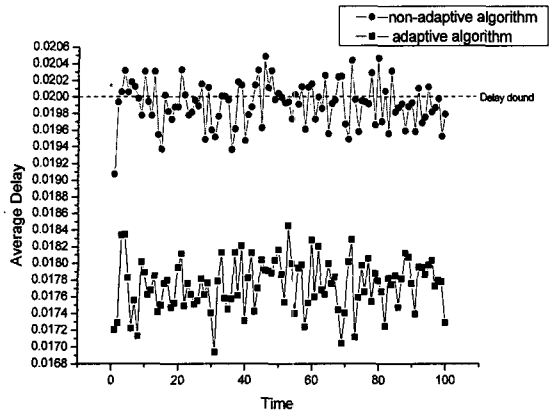


그림 4. 지수형 트래픽에 대한 클래스1 트래픽 지연
Fig. 4. Delay of class 1 with exponential traffic load.

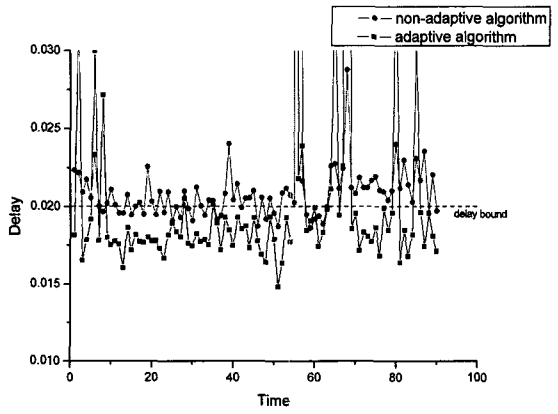


그림 5. 버스트 트래픽에 대한 절대적 지연
Fig. 5. Absolute delay to the bursty traffic.

capacity is set to 100Mbps and the link propagation delay is assumed to be negligible.

Fig. 4 shows a result of queueing delay of class 1 for an exponential traffic. It shows that the adaptive scheme clearly meets the delay bound while non-adaptive scheme frequently exceeds the delay 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.

We simulate the algorithm using the traffic and results are shown as Fig. 5. 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.

V. Implementation

The test board for implementing the suggested algorithm is shown in Fig. 6.

It is mainly composed of 10/100 Base T PHY/MAC, XPC860P CPU and Xilinx Spartan XC3S1500 FPGA. The suggested algorithm is implemented with VHDL on Xilinx Spartan XC3S1500 FPGA and the board is controlled with XPC860P CPU. The simulation is performed with ModelSim SE 5.7d.

The synthesized circuit for the suggested algorithm is shown in Fig. 7.

The signals, addr(2:0), cpu_clk, cs, rw, data(7:0), and ta_n are used in interfacing with the CPU. The CPU sets the target delay on the FPGA, starts the operation, and gets the delay results. The delay results are given with delay_sum(0:30) and delay_count(0:30).

Fig. 8 shows the internal blocks for the

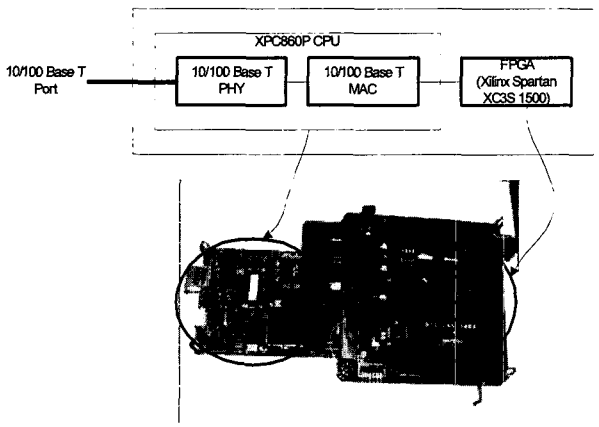


그림 6. 제안된 알고리듬을 구현한 시험 보드
Fig. 6 The test board for implementing the suggested algorithm.

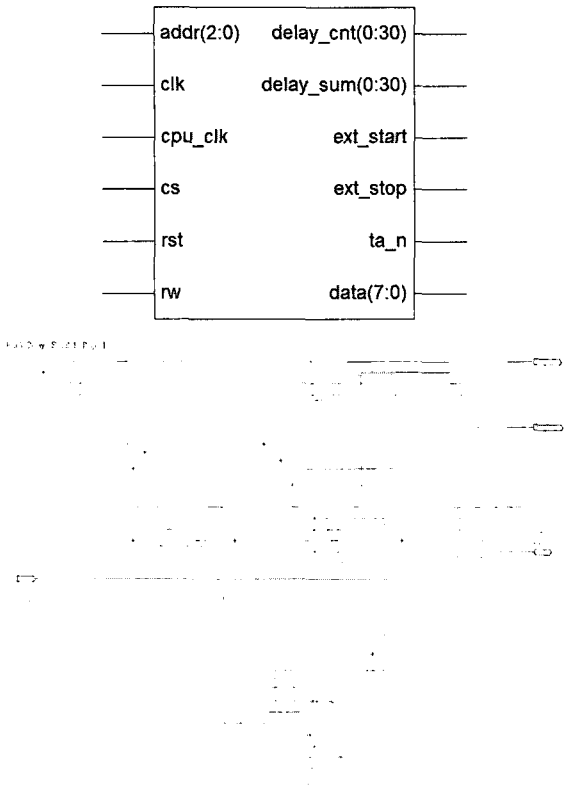


그림 7. 제안된 알고리듬의 합성 회로
Fig. 7. The synthesized circuit for the suggested algorithm.

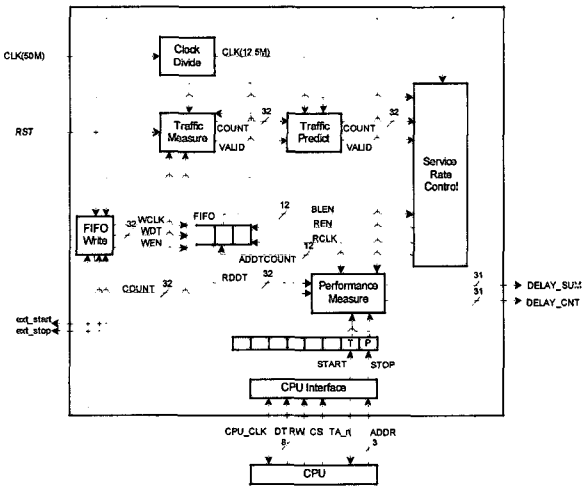


그림 8. 합성회로의 내부 블록
Fig. 8. The internal blocks for the synthesized circuit.

synthesized circuit.

Clock divide part divides 50[MHz] clock to generate 12.5[MHz] clock for accommodate the 100[Mbps] Ethernet signal with 8bits operations. FIFO Write part accounts for generating the Ethernet data and inserting it to the FIFO. Traffic Measure part constantly measures the input traffic rate and

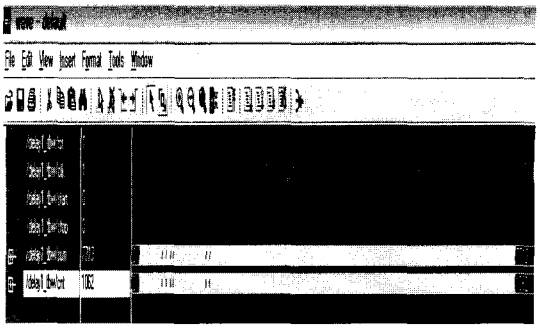


그림 9. 타이밍 시뮬레이션 결과(목표 지연:1[us])
Fig. 9. Post route simulation result(target delay:1[us]).

sends the measured traffic value to Traffic Predict part. Traffic Predict part predicts the traffic amount which will be arriving for the next time slot and the predicted one is sent to Service Rate Control part. Service Rate Control part derives the service rate to FIFOs from the delay differentiation algorithm which is suggested in chapter III. Lastly, Performance Measure part calculates the average delay from measuring the difference between the arrival time and the service time of the traffic.

DELAY_SUM and DELAY_CNT which are the outputs of Performance Measure part indicate the sum of delays for the measured packets and the number of packets during the measured time respectively. As the input packets are arriving at the rate of 12.5[MHz], the accurate average delay is derived from (DELAY_SUM / DELAY_CNT) / 12.5[us].

Fig. 9 is the post route simulation result in case that the target delay is set to 1[us]. As DELAY_SUM and DELAY_CNT are 7713 and 1062 respectively, the target delay is satisfied because the simulated average delay is $\frac{7713}{1062}\times\frac{1}{12.5[us]}=0.58[us]$.

표 1. 시험보드에서 측정한 지연
Table 1. The delay measured at the test board.

Target delay[us]	Measured delay[us]
1	0.5
7	5
15	11
23	17
87	66
255	176

The post route simulation is performed for the case of the target delay of 3 us, 5 us, and 15 us, and the resulting delays are 1.84 us, 3.26 us, and 9.54 us, respectively. All the delay constraints are satisfied.

Table 1 is the delay result measured at the test board where the implemented VHDL code is downloaded. As it indicates, the suggested algorithm guarantees the target delay.

VI. Conclusion

In this paper, a delay differentiation algorithm that achieves absolute QoS provisioning is proposed and the implementation scheme for it is presented. 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 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.

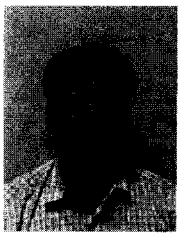
[15] Chin-Chang Li, Shiao-Li Tsao, Meng Cheng Chen, Yeali Sun, and Yueh-Min Huang, "Proportional Delay Differentiation Service Based on Weighted Fair Queueing," 2000 IEEE ICCCN, p418-423, Oct. 2000.

[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 동아방송예술대학 방송통신과 교수
<주관심분야 : QoS, 스위치, 전송>

주 범 순(정회원)
1983년 서울대학교 전자공학과 학사 졸업.
1999년 한국과학기술원 전기 및 전자공학과 석사 졸업
1984년~현재 한국전자통신연구원 네트워크 연구그룹 캐리어이더넷 연구팀장
<주관심분야 : High Speed Networks, Broadband convergence Networks, Ethernet Technology, Network Synchronization>



김 대 업(정회원)
1999년 영남대학교 전자공학과 학사 졸업.
2001년 한국정보통신대학교(ICU) 공학부 통신전공 석사 졸업.
2001년~현재 한국전자통신연구원 선임연구원
<주관심분야 : 네트워크시스템설계, 캐리어이더넷, 이더넷 OAM>