DiffServ 기반 네트워크에서의 실시간 트래픽 서비스

정회원 정 진 우*

Real-time traffic service in networks with DiffServ

Jinoo Joung* ***Regular Member***

요 약

본 연구에서는 Diffserv 기반의 네트워크, 특히 그 중에서도 트리구조를 가지는 네트워크에서의 지연시간에 대해서 연구하였다. 이러한 네트워크에서의 지연시간 최대치를 구하는 수식을 구하였고, 특히 기존에 알려진 바와는 달리 트리구조의 네트워크에서는 네트워크 utilization과는 상관없이 최대치가 항상 존재함을 보였다. 지연시간 최대치의 특성은 다음과 같이 표현된다. 네트워크 utilization이 큰 경우 네트워크내의 최대 hop count의 제곱에 비례하며, 네트워크 utilization이 작은 경우는 최대 hop count에 비례한다. 이러한 결과를 바탕으로 실제 Metro Ethernet Network과 같은 대규모 네트워크의 경우에서 DiffServ를 이용하여 실시간 트래픽 전송이 가능하다는 것을 보였다.

Key Words : QoS, DiffServ, Delay bound, Tree topology, LR server

ABSTRACT

We investigate the end-to-end delay bounds in large scale networks with Differentiated services (DiffServ) architecture. It is generally understood that networks with DiffServ architectures, where packets are treated according to the class they belong, can guarantee the end-to-end delay for packets of the highest priority class, only in lightly utilized cases. We focus on tree networks, which are defined to be acyclic connected graphs. We obtain a closed formula for delay bounds for such networks. We show that, in tree networks, the delay bounds exist regardless of the level of network utilization. These bounds are quadratically proportional to the maximum hop counts in heavily utilized networks; and are linearly proportional to the maximum hop counts in lightly utilized networks. Considering that tree networks, especially the Ethernet networks are being accepted more and more for access networks as well as provider networks, we argue that based on these delay bounds DiffServ architecture is able to support real time applications even for a large network. Throughout the paper we use Latency-Rate (LR) server model, with which it has proven that FIFO and Strict Priority are LR servers to each flows in certain conditions.

Ⅰ. Introduction

DiffServ[1] is a well-known QoS architecture that has been proposed to solve the scalability problem of Integrated Services (IntServ). DiffServ classifies packets, or the flows they belong, into a number of traffic classes. The packets are marked accordingly at the edge of a network. Therefore the hard works are necessary only at the edge nodes. Classes may be assigned with strict

※ 본 연구는 상명대학교 소프트웨어 미디어연구소의 지원으로 수행되었습니다.
* 상명대학교 소프트웨어학부 (jjoung@smu.ac.kr)
논문번호 : KICS2006-11-505, 접수일자 : 2006년 11월 22일, 최종논문접수일자 : 2007년 1월 4일

priorities, or a certain amount of bandwidth is provisioned for each class, as was the case with a flow in IntServ. We consider networks with DiffServ architecture, especially the QoS characteristics of the highest priority class traffic with the strict priority scheduling scheme. We focus on the queueing and scheduling behaviors of the flows or aggregated flows, and investigate the delay characteristics of them.

QoS characteristics of the network with IntServ architecture have been well studied and understood by numerous researches in the past decade. Providing the allocated bandwidths, or service rates, or simply rates of an output link·to multiple sharing flows plays a key role in this approach. A myriad of scheduling algorithms has been proposed. The Packetized Generalized Processor Sharing (PGPS)[2] and Deficit Round Robin (DRR)[3], and many other rate-providing servers are proved to be a Latency-Rate server[4], or simply LR server. All the work-conserving servers that guarantee rates can be modeled as LR servers. The behavior of an LR server is determined by two parameters, the latency and the allocated rate. The latency of an LR server may be considered as the worst-case delay seen by the first packet of the busy period of a flow. It was shown that the maximum end-to-end delay experienced by a packet in a network of LR servers can be calculated from only the latencies of the individual servers on the path of the flow, and the traffic parameters of the flow that generated the packet. More specifically for a leaky-bucket constrained flow,

$$D_i \le \frac{\sigma_i - L_i}{\rho_i} + \sum_{j=1}^{N} \Theta_i^{S_j} \qquad (1)$$

where D_i is the delay of flow i within a network, σ_i and ρ_i are the well known leaky bucket parameters, the maximum burst size and the average rate, respectively, L_i is the maximum packet length and $\Theta_i^{S_j}$ is the latency of flow i at the server S_j.

It is proved that FIFO scheduler and strict priority scheduler used in DiffServ network is also an LR server to the individual flows within a class, under conditions that every flow conforms to leaky-bucket model and the aggregated rate is less than or equal to the link capacity[5]. We will apply this result with the LR server model to DiffServ networks for further analysis. The current belief on such networks is that only with the low enough network utilization, one can guarantee delay bounds. Otherwise it is not certain that the bounds exist, or if they do they *explode* to infinity with just sufficient number of hops or the size of the network[6]. Based on this argument, there have been a trend of aborting DiffServ for delay sensitive real-time applications such as Voice over IP (VoIP)[7][8][9][10][11]. The exploding nature of the delay bounds that has stimulated such diverse research activities, however, is the conclusion with general network topology. If we can somehow avoid the burst size accumulation due to the loops formed in a network, which is suspected as the major reason for the explosion, the delay bounds may be still useful with the simple DiffServ architecture. We investigate this possibility throughout the paper. We focus on the tree networks where the loops are avoided. The tree networks can be found in many types of Local Area Network (LAN), especially Ethernet network. Ethernet networks are gaining momentum to be extensively used more and more in access networks as well as provider networks[12]. In a large scale, these networks together form a tree topology, by means of spanning tree algorithms or manual configurations[13][14].

Ⅱ. Iterative calculation for delay bounds in networks with DiffServ architecture

Under a condition that there is no flow that violates the leaky bucket constraint specified for itself, we can guarantee the delay upper bound for each flows, even with relatively simple scheduling strategies. If there is only a single class of flows, which specifies for their required bandwidths then First-in-first-out (FIFO) servers can guarantee the delay bounds. If in the network there are best-effort traffic that does not specify its bandwidth, the minimum protection against such best-effort traffic is necessary. In this regard the Strict Priority (SP) scheduling can guarantee delay bounds for flows of the real-time traffic. A Strict Priority (SP) server is a server that maintains at least two queues. The queue with highest priority, which is for the flows with real-time constraints, transmits packets whenever it has one, right after the completion of the current packet that is being served.

In this environment, the sum of all flows that want to pass through a server is compared with the link capacity, and if it's less than the link capacity then delay upper bounds will be prescribed, as it can be calculated with the method explained in this section. We consider a network of packet level servers.

Theorem 1. A FIFO server or an SP server, under conditions that all the input flows are leaky bucket constrained and the sum of average rates is less than the link capacity, is an LR server for individual flows with latency given as the following:

$$\Theta_i^S = \frac{\sigma^S - \sigma_i^S}{r^S} + \Theta^S, \tag{2}$$

where σ^S is the sum of all the σ_i^S within the server S, γ^S is the link capacity of S, and

$$\Theta^S = \begin{cases} L/r^S \text{ when } S \text{ is FIFO} \\ 2L/r^S \text{ when } S \text{ is SP.} \end{cases}$$

Proof. See the proof in section 3 of [5].

Corollary 1. The output traffic of flow i from a FIFO server or an SP server S conforms to the leaky bucket model with parameters $(\sigma_i^S + \rho_i \Theta_i^S, \rho_i)$, where σ_i^S is the maximum burst size of flow i into the server S.

The end-to-end delay of a network with DiffServ architecture with FIFO servers and/or Strict Priority servers can be obtained by the following sets of equations:

$$D_i \le \frac{\sigma_i - L_i}{\rho_i} + \sum_{n=1}^{N} \Theta_i^{In},$$

$$\Theta_i^{In} = \frac{\sigma^{In} - \sigma_i^{In}}{r^{In}} + \Theta^{In},$$

$$\sigma_i^{In+1} = \sigma_i^{In} + \rho_i \Theta_i^{In}, \text{ for } n \ge 1, \tag{3}$$

where I_n is the n_{th} server of the network for i, N is the number of servers that i traverses in the network; and $\sigma_i^{I1} = \sigma_i$.

Ⅲ. Closed-form delay bound in tree network

While (3) gives a tight bound through iterative computation of the latencies and the maximum burst sizes of each server, we still have to make assumptions about the burst sizes of other flows. With a reasonable restriction on the network topology, the other flows' burst sizes can be inferred, and the delay bound for a whole network can be obtained. In this section we consider delay bounds in tree networks, which are defined to be acyclic connected graphs. Tree networks appear in a broad range of networks. For example in Ethernet, both in LAN and in wide area networks, the logical tree-based network topology is achieved by running Spanning Tree Protocol (STP) or static configuration (e.g. configuring Virtual LANs)[13][14].

Let us define *hop* by the server and the accompanying link through which packets are queued and serviced and then transmitted. First we start by observing an important property of tree networks. Let i be the flow under observation.

Lemma 1. Consider a tree network with the flow under observation, i. Assume that the flow i traverses the path of the maximum possible hop counts. Then at any server in the i's path, other flows that are confronted by i have traversed less number of hops than i just have.

Proof. Let us denote the i's path by $(I_1, I_2, \cdots, I_H)$ where I_n is the n_{th} server in the i's path and H is the maximum number of hops possible in the given tree network. Similarly let us denote another flow j's path by $(J_1, J_2, \cdots, J_m)$, where $m \le H$. Let $I_k = J_l$, for some k and l where $1 \le k \le H$ and $1 \le l \le m$, that is the flow i and j confront each other at a server in the path. A path is defined to be a sequence of nodes, with no repeated nodes, in which each adjacent node pair is linked. We will show that $k \ge l$ for any case, by contradiction. Assume $k < l$.

Case 1: $\{J_1, J_2, \cdots, J_{l-1}\}$ is disjoint with $\{I_{k+1}, \cdots, I_H\}$. This is to say that there is no server that takes part of both the remaining path of i and the traveled path of j. Then the path $(J_1, J_2, \cdots, J_{l-1}, I_k, I_{k+1}, \cdots, J_p)$, exists that has more hops than H. This contradicts to the assumption that H is the maximum possible hop counts.

Case 2: There is at least one server in the remaining path of i, $(I_{k+1}, \cdots, I_H)$, that is also a server in the traveled path of j, $(J_1, \cdots, J_{l-1})$. Let us call this server J_p. Then $(J_p, J_{p+1}, \cdots, J_{l-1} I_k, I_{k+1}, \cdots, J_p)$ forms a cycle, which contradicts the assumption that the network is a tree.

In both cases the statement $k < l$ contradicts the assumption. Therefore $k \ge l$ for any case and the lemma follows.

Lemma 2. lets us infer about the maximum burst sizes of the confronted flows in the path. Therefore the end-to-end delay bound can be obtained from a few network parameters.

Now consider a server I_n in the path of i. Let the set of flows F_{In}, including i, are competing for service in I_n. For any flow j, $j \in F_{In}$, which has traveled $(n-m-1)$ hops until reaching I_n, imagine a corresponding flow j' with m more hops from the starting node of the flow j. Moreover j' has entered the network with the same parameter with j. Further imagine that for additional hops for each flows, not the existing nodes but new nodes are attached to the starting nodes of each flows, so that the numbers of flows in the upstream nodes of I_n are intact. Now we have constructed an imaginary network, in which at I_n, the flows in F_{In} all have traveled exactly $(n-1)$ hops until reaching I_n. We claim the following.

Lemma 3. The maximum burst size of any flow at the entrance of I_n in the constructed network is always greater than or equal to that of the original network. That is,

$$\sigma_j^{In} \le \sigma_{j'}^{In}, \tag{4}$$

for all $j \in F_{In}$.

Proof. It is enough to show that $\sigma_j^{J1} \le \sigma_{j'}^{J1}$ for any j, $j \in F_{In}$. Since σ_j^{Jn} is a nondecreasing function of n,

$$\sigma_{j'}^{J1} \ge \sigma_{j'}^{J(1-m)} = \sigma_j = \sigma_j^{J1}. \tag{5}$$

The lemma follows.

We argue the main result of the paper as the following.

Theorem 2. The end-to-end delay of a tree network with DiffServ architecture with FIFO servers is bounded by

$$D_i \le \frac{\sigma_i - L_i}{\rho_i} + \tau\frac{(1+\alpha)^H - 1}{\alpha}, \tag{6}$$

where τ and α are defined as

$$\sum_{j \in F_s} \sigma_j \le \tau r^S, \sum_{j \in F_s} \rho_j \le \alpha r^S, \tag{7}$$

for any server S in the network, in which there is a set of flows, F_S.

The parameters τ and α is similarly defined in [7]. We will call τ the *burst allowance level* measured in time for their transmission, and α the *network utilization*. Note that $0 < \alpha < 1$ in our network configurations.

Proof. First, we will show that the following inequalities hold for any I_n, $n \le H$.

$$\Theta_i^{In} \le \tau(1+\alpha)^{n-1},$$

$$\sigma_i^{In} \le \sigma_i + \tau\rho_i\frac{(1+\alpha)^{n-1} - 1}{\alpha}. \tag{8}$$

Let us first assume that (8) is true for I_n. We will show that it holds for I_{n+1} as well, and then it holds for I_1, therefore it holds for any I_n. If

(8) is true for I_n, from (3) we get

$$\sigma_i^{In+1} = \sigma_i^{In} + \rho_i \Theta_i^{In} \le \sigma_i + \tau\rho_i((1+\alpha)^n - 1)/\alpha. \quad (9)$$

From (9), and from lemma 3, the sum of the maximum burst sizes of all the incoming priority flows in I_{n+1},

$$\sigma^{In+1} = \sum_{j\in F_{In}} \sigma_j^{In+1} \le \sum_{j'\in F_{In}} \sigma_{j'}^{In+1} \le \sum_{j'} \{\sigma_j + \tau\rho_j((1+\alpha)^n - 1)/\alpha\}, \quad (10)$$

since any j, $j\in F_{In}$, has traveled $(n-1)$ hops, therefore $\sigma_j^{In+1} = \sigma_j + \tau\rho_j((1+\alpha)^n - 1)/\alpha$ by the assumption at the beginning of the proof. We obtain

$$\sigma^{In+1} \le \tau r^{In+1}(1+\alpha)^n \quad (11)$$

from (7). Equation (3) yields

$$\Theta_i^{In+1} = \frac{\sigma^{In+1} - \sigma_i^{In+1}}{r^{In+1}} + \Theta^{In+1} \quad (12)$$

Note that Θ^{In+1} is L/r^{In+1} for a FIFO server. The maximum burst size of a flow, by definition, is always greater than or equal to the maximum packet length, that is $\sigma_i^{In+1} \ge L$. Therefore, and from (11),

$$\Theta_i^{In+1} \le \frac{\sigma^{In+1}}{r^{In+1}} \le \tau(1+\alpha)^n. \quad (13)$$

With (9) and (13), we have shown that (8) holds for I_{n+1}. Now we'll consider the case for I_1. For I_1, $\sigma^{I1} = \sum_{j\in F_{I1}} \sigma_j \le \tau r^{I1}$, and

$$\Theta_i^{I1} \le \tau - \frac{\sigma_i}{r^{I1}} + \frac{L}{r^{I1}} \le \tau, \quad (14)$$

which shows that (8) holds for I_1 as well. From (3),

$$D_i \le \frac{\sigma_i - L_i}{\rho_i} + \sum_{n=1}^{H} \tau(1+\alpha)^{n-1} \le \frac{\sigma_i - L_i}{\rho_i} + \tau\sum_{n=0}^{H-1}(1+\alpha)^n \le \frac{\sigma_i - L_i}{\rho_i} + \tau\frac{(1+\alpha)^H - 1}{\alpha}, \quad (15)$$

for $H \ge 1$.

Similar conclusion can be claimed for the network with SP servers.

Theorem 3. The end-to-end delay of a tree network with DiffServ architecture with SP servers is bounded by

$$D_i \le \frac{\sigma_i - L_i}{\rho_i} + \tau'\frac{(1+\alpha)^H - 1}{\alpha}, \quad (16)$$

where τ' and $\alpha$are defined as

$$L + \sum_{j\in F_S} \sigma_j \le \tau' r^S, \ \sum_{j\in F_S} \rho_j \le \alpha r^S, \quad (17)$$

for any server S in the network, in which there s a set of flows, F_S.

Proof. The proof of this theorem is exactly the same with that of the previous theorem except that Θ^{In+1} in (12) is $2L/r^{In+1}$ for an SP server, therefore (13) is still valid with the modified τ'. We omit the detail.

Ⅳ. Discussion

We first examine the extreme cases; in which $\alpha\to 0$ or $\alpha\to 1$. When $\alpha\to 0$, the delay bound becomes $(\sigma_i/\rho_i + H\tau')$. For the worst case delay we can say that it is $(\max_j(\sigma_j/\rho_j) + H\tau')$. When $\alpha\to 1$, the delay bounds becomes $(\max_j(\sigma_j/\rho_j) + (2^H - 1)\tau')$. The delay bounds increases linearly as hop count increases, when the utilization is low. When the utilization is high, however, the delay bounds increases quadratically with hop counts.

The delay bounds in a general topology network with DiffServ architecture have been obtained in the literatures [6, 15]. In [6], it was concluded that unless the link utilizations are kept under a certain level, the end-to-end delay explodes to infinity. The delay bound obtained in [6] is, only under condition that $\alpha < 1/(H-1)$,

$$D \leq \frac{H}{1-(H-1)\alpha}\tau', \tag{18}$$

for a case with infinite incoming links' capacity, which is also the case considered in this paper. Equation (18) becomes, as $\alpha \rightarrow 0$, $D \leq \tau' H$. As it was already noted in [15], however, (18) has apparently not taken the non-preemptive nature of SP servers into consideration, therefore was later corrected in [15] to, under the same condition,

$$D \leq \frac{H}{1-(H-1)\alpha}(\tau' + \max_j(L/\sum_j \rho_j)). \tag{19}$$

Equation (19) becomes, as $\alpha \rightarrow 0$, $D \leq (\tau' + \max_j(L/\sum_j \rho_j))H$. Table 1 summarizes the delay bounds obtained from this paper and from two previous works, with the suggested network parameters both in [6, 15].

Table 1. The bounds from this paper and from the related works, in milliseconds: $H=10$, $\sigma_j = 100$ bytes for all flows, ρ_i=32kbps for all flows, L=1500 bytes, r^S =149.760Mbps for all S.

α	0.04	0.08
Bound by (16)	12.97	30.13
Bound in [6]	16.88	74.29
Bound in [15]	48.18	110.06

The reason that the bounds obtained in this paper is less than the one from [15] is twofold. The first and obvious reason is that our network is strictly confined in a tree topology. This restriction inhibits the burst accumulation in loop, so that the bound still exists even in high network utilization. The second reason is that, even in the low network utilization cases, the bounds in previous works are simply the summation of nodal delays. Our bound, on the other hand, is from considering the whole network as a single virtual node, so that the correlation among consecutive nodes are taken into account. This characteristic is the primary virtue of the analysis based on the LR server[4]. The bounds in table 1, however, is meaningless since the maximum burst size for every flow, σ_j, is set to be less than the maximum packet length, which cannot be real cases. In the following table, we examine the bounds with varying H, the maximum number of hops, and with varying α, the network utilization. As table 2 suggests, even in a tree network, delay bounds with moderate to high network utilization seems quite unacceptable.

Table 2. The bounds obtained from (16), in seconds, with varying H and α: σ_j= L= 1500 bytes for all flows, ρ_i=32kbps for all flows, r^S = 149.760Mbps for all S.

α	0.1	0.3	0.5	0.7
$H=6$	0.290	1.436	3.898	8.679
$H=8$	0.430	2.686	9.240	25.792
$H=10$	0.599	4.798	21.258	75.245
$H=12$	0.804	8.368	48.301	218.175

Since in numerous standardization bodies it is suggested that the end-to-end delay bound for voice to be less than 400ms[16], only with α much less than 0.1 can only meet those standards, if network parameters in table 2 are to be used.

Consider now where the maximum burst size, one of primary contributor to delay bound increment, and the maximum packet length can be controlled to be much less than the ordinary IP packet length. This assumption on the maximum packet length, therefore on the maximum burst size, is not extravagant since if we are to transmit the MPEG-2 Transport Streams (TS) data whose lengths are fixed at 188 bytes[17], with 12 bytes RTP fixed header[18], 4 bytes RTP video-specific header[19], 8 bytes UDP header, 20 bytes IP header and finally 26 bytes Ethernet header and trailer including preamble, then the maximum packet length in this case becomes 258 bytes. Considering the extended headers fields and Ethernet inter-frame gap, the maximum packet length will be about 300 bytes. In such a network of limited packet length, the 400ms requirement can be reasonably met as table 3 suggests.

Table 3. The bounds obtained from (16), in seconds, with varying H and α: $\sigma_j = L = 300$ bytes for all flows, $\rho_i = 32$kbps for all flows, $r^S = 149.760$Mbps for all S.

α	0.05	0.1	0.2	0.3
$H=6$	0.0256	0.0580	0.149	0.287
$H=8$	0.0360	0.0859	0.248	0.537
$H=10$	0.0474	0.120	0.390	0.960
$H=12$	0.0600	0.161	0.594	1.674
$H=14$	0.0738	0.210	0.889	2.880
$H=16$	0.0891	0.270	1.313	4.919

For example, if we manage to restrict the maximum burst size at the entrance of the network), then at the 10% network utilization even a very large network with 16 hop counts can successfully support the voice applications. Considering that the applications requesting Premium Service in DiffServ are usually a small portion of the total traffic, table 3 suggests the DiffServ may indeed be useful in some cases.

V. Conclusion

Contrary to the traditional belief that without a very small network utilization delay bound goes to infinity, we have shown that in tree networks delay bounds always exist with DiffServ. We have shown also that this bound is linearly proportional to the number of hop counts when the utilization is small; and is quadratically proportional to the number of hop counts when the utilization is large. It may be argued, however, that the existing bound is not acceptable for moderate to large sized networks with moderate to large network utilization, even in tree networks. On the other hand, with a manipulation to the network configuration such as maximum packet length restriction, we have shown that the DiffServ architecture can support the real-time application even in large networks with moderate network utilization.

Although the result we have obtained in this paper is applicable only to tree networks, in which loops are strictly avoided, the insight we gained that a tighter bound can be derived by taking into consideration the topology characteristics of a network, is still valid. Hence the future work may consist of the delay bound investigation on a broader range of network topology.

References

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for Differentiated Services," IETF RFC 2475, 1998.

[2] A. K. Parekh and R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case," *IEEE/ACM Trans. Networking*, vol. 1, no. 3, June 1993.

[3] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round-robin," *IEEE/ACM Trans. Networking*, vol. 4, no.3, pp. 375--385, June 1996.

[4] D. Stiliadis and A. Varma, "Latency-Rate servers: A general model for analysis of traffic scheduling algorithms," *IEEE/ACM Trans. Networking*, vol. 6, no. 5, Oct. 1998.

[5] J. Joung, B-S. Choe, H. Jeong, and H. Ryu: Effect of Flow Aggregation on the Maximum End-to-End Delay. Lecture Notes in Computer Science, Vol. 4208. Springer-Verlag (2006) 426--435

[6] A. Charny and J.-Y. LeBoudec, "Delay bounds in a network with aggregate scheduling," in *Proc. First International Workshop of Quality of Future Internet Services (QOFIS'2000)*, 2000.

[7] F. Le Faucheur et al., "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services," IETF RFC 3270, May 2002.

[8] B. Rong, et al., "Modeling and Simulation of Traffic Aggregation Based SIP over MPLS Network Architecture," in *Proc. IEEE 38th Annual Simulation Symposium (ANSS'05)*, 2005.

[9] C. Dovrolis, D. Stiliadis, and P. Ramanathan, "Proportional Differentiated Services: Delay Differentiation and Packet Scheduling," *IEEE/ACM Trans. on Networking*, Vol. 10, No. 1, Feb. 2002.

〔10〕 A. Kortebi, S. Oueslati, and J. Roberts, ''Cross-protect: implicit service differentiation and admission control," in *Proc. IEEE HPSR*, April 2004.

〔11〕 S. Oueslati and J. Roberts, ''A new direction for quality of service: Flow-aware networking," in *Proc. Conference on Next Generation Internet Networks (NGI)*, April 2005.

〔12〕 "'Layered Network Architecture and Implementation for Ethernet Services," White Paper, Fujitsu Network Communications, Inc., 2004.

〔13〕 "'802.1ad Virtual Bridged Local Area Networks- Amendment 4: Provider Bridges," Ammedment to IEEE 802.1Q, May 2006.

〔14〕 "Metro Ethernet Service - A Technical Overview," White Paper, R. Santitoro, Metro Ethernet Forum, 2003. 〔Online〕.

〔15〕 Y. Jiang, "Delay Bounds for a Network of Guaranteed Rate Servers with FIFO," *Computer Networks*, Elsevier Science, Vol. 40, No. 6, pp. 683-694, 2002.

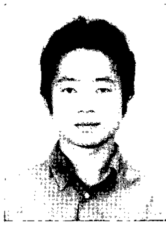
〔16〕 "Network performance objectives for IP-based services," International Telecommunication Union - Telecommunication Standadization Sector (ITU-T), Recommendation Y.1541, Feb. 2006.

〔17〕 "Information Technology -- Generic Coding of Moving Pictures and Associated Audio Information Part 1: Systems," ISO/IEC InternationalStandard IS 13818, November 1994.

〔18〕 H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," IETF RFC 3550, July 2003.

〔19〕 Hoffman, D., Fernando, G., Goyal, V. and M. Civanlar, "RTP Payload Format for MPEG1/MPEG2 Video," IETF RFC 2250, January 1998.

정 진 우 (Jinoo Joung) 정회원



1992년 KAIST 전기전자공학과 공학사

1997년 Polytechnic Univ., NY, USA, 공학박사 (Ph.D.)

1997년~2001년 삼성전자 중앙연구소

2001년~2005년 삼성종합기술원

2005~현재 상명대학교 소프트웨어학부