

# 음성 트래픽을 위한 동적우선권제어방식의 성능분석

## Performance Analysis of a Dynamic Priority Control Scheme for Delay-Sensitive Traffic

김도규\*, 김용규\*, 조석팔\*  
(Dokyuu Kim\*, Yongkyu Kim\*, Sokpal Cho\*)

\*성결대학교 컴퓨터학부

(접수일자: 2000년 10월 20일; 채택일자: 2000년 11월 14일)

음성의 특성은 지연에 매우 민감한 (delay-sensitive) 트래픽으로 시스템에서 손실에 민감한 (loss-sensitive) 다른 트래픽과 동일한 방식으로 서비스되어서는 두 트래픽 모두의 QoS (Quality of Service)를 만족시킬 수 없다. 효율적인 트래픽의 처리를 위해서는 시스템의 상태에 따라 손실에 민감한 트래픽에 영향을 적게 주면서 지연민감 트래픽의 성능도 만족하는 것이 바람직하다. 본 논문에서 도입한 동적우선권제어함수 (Dynamic Priority Control Function, DPCF)는 시스템의 상태에 따라 각 트래픽에 우선권을 동적으로 할당하여 서비스 스케줄링을 제어하는 함수  $f(\cdot)$ 로 기존의 우선권 제어방식을 더욱 일반화시킨 방식이다. 클래스 1의 손실민감 트래픽과 음성과 같은 클래스 2의 지연민감 트래픽을 수용하는 시스템에 TBPJ (Threshold-based Bernoulli Priority Jump) 방식의 DPCF를 적용하여 성능을 분석하였다. DPCF은 각 시스템에 대기하고 있는 트래픽의 양과 시스템의 가용 용량인 버퍼의 threshold에 따라 각 트래픽 클래스에 우선권을 동적으로 할당하여 서비스가 이루어지도록 한다. 성능분석을 통하여 TBPJ 제어방식이 기존의 우선권 제어 방식보다 성능 및 효율성에 있어서 우수함을 입증하였다. 즉 성능을 분석한 결과 손실민감 트래픽의 성능저하는 미미하고 지연민감 트래픽의 성능향상은 현저히 개선된 것을 확인하였다.

**핵심어:** 동적 우선권 제어방식, 임계값 기반의 베르누이 우선순위도약 (TBPJ), HOL 가상큐

**투고분야:** 음향 통신기술 분야 (6.5)

This paper considers the performance of a dynamic priority control function (DPCF) of a threshold-based Bernoulli priority jump (TBPJ) scheme. Loss-sensitive and delay-sensitive traffics are applied to a system with a TBPJ scheme that is a general state-dependent Bernoulli scheduling scheme. Loss-sensitive and delay-sensitive traffics represent sound and data, respectively. Under the TBPJ scheme, the first packet of the loss-sensitive traffic buffer goes into the delay-sensitive traffic buffer with Bernoulli probability  $p$  according to system states which represent the buffer thresholds and the number of packets waiting for scheduling. Performance analysis shows that TBPJ scheme obtains large performance build-up for the delay-sensitive traffic without performance degradation for the loss-sensitive traffic. TBPJ scheme shows also better performance than that of HOL scheme.

**Key words:** Dynamic priority control function (DPCF), Threshold-based bernoulli priority jump (TBPJ), HOL virtual queue

**Subject classification:** Acoustic communication (6.5)

### I. 서 론

반도체 및 컴퓨터 기술의 발전으로 음성정보와 데이터 정보가 하나의 단말에서 통합되어 처리되고 이를 네트워크를 통하여 전달하게 되었다. 최근 인터넷을 통한 음성 정보의 전달은 이러한 정보의 통합을 보여주는 대표적인 예이다. 그러나 음성정보를 네트워크를 통하여 전달하는

경우 지연에 매우 민감 (delay-sensitive)하여 네트워크에서 적절한 처리가 이루어져야 사용자가 원하는 수준의 음성품질을 만족시킬 수 있다. 반면에 데이터 정보는 손실에 민감 (loss-sensitive)하기 때문에 역시 적절한 처리가 이루어져야 한다. 하나의 시스템에서 이들 트래픽을 동일한 방식으로 서비스하는 경우, 가장 엄격한 QoS (Quality of Service)를 가진 트래픽이 수용될 수 있도록 시스템이 설계되어야 한다. 그러나 이 경우 시스템은 낮은 이용률 (utilization)과 대용량의 버퍼링 등이 필요하게 된다. 따라서 각 트래픽의 QoS를 만족시키면서 시스템의 이용률을

책임저자: 김도규 (dkkim@hana.sungkyul.ac.kr)  
430-742 경기도 안양시 만안구 안양8동 산 147-2  
성결대학교 컴퓨터학부  
(전화: 031-467-8108; 팩스: 031-449-0529)

향상시키기 위하여 다양한 트래픽 제어방식이 요구된다.

음성이나 데이터 등의 정보를 네트워크에서 수용하는 경우 대표적인 QoS는 손실확률 (loss probability)과 종단간의 지연시간 (end-to-end delay)이다. 그리고 트래픽은 크게 클래스 1의 손실민감 트래픽과 클래스 2의 지연민감 트래픽으로 구분할 수 있다. 손실민감 트래픽과 지연민감 트래픽을 처리하는 방식에서 우선권 제어방식 방식이 가장 간단하다. 그러나 이 우선권 제어 방식을 어느 하나의 트래픽 클래스에 적용하는 경우 다른 트래픽의 QoS는 만족시킬 수 없다. 따라서 지금까지 이를 보완할 수 있는 다양한 트래픽 제어방식이 연구되었다 [1]-[12]; 지연민감 트래픽을 중요시하는 delay-priority 방식의 경우에 head-of-line (HOL) 우선권 제어방식을 사용할 수 있는데, 이 방식은 클래스 2 트래픽에 우선권을 고정적으로 할당하는 방식으로, 클래스 2 트래픽 용량이 증가하는 경우 클래스 1 트래픽의 성능이 심각하게 저하되는 문제가 있다. 이를 보완하기 위해 Chipalkatti는 minimum laxity threshold (MLT)[1], queue length threshold (QLT)[1], Goldberg는 earliest-due-date (EDD) 방식[2], Lim은 HOL with priority jump (HOL-PJ) 방식[3] 등을 연구되었다. Chen은 preemptive priority (PP)를 사용한 two-priority input queuing 스위치의 성능을 분석하였고[4], 특히 low priority 트래픽에 대한 saturation throughput에 대하여 연구하였다. Gupta는 delay-priority input queueing 스위치에서 non-preemptive priority 방식에 대하여 연구하였다[5]. 반면에 손실에 민감 트래픽을 중시하는 loss-priority control의 경우 네트워크에서 혼잡 (congestion)이 발생하는 경우 낮은 priority 패킷을 우선적으로 discarding 해서 손실확률을 감소시키는 것이 일반적이다. HOL 우선권 방식을 사용하는 경우는, 역시 high 트래픽에 대하여 다른 클래스의 트래픽에 심각한 지연을 발생시키는 문제가 있다. 따라서 이를 보완할 수 있는 push-out, partial buffer sharing 방식과 separate 버퍼를 사용하여 제어하는 방식이 연구되었다[6]-[7]. Lee와 Sengupta는 threshold-based delay-priority 방식을 제안하여, 이를 실시간 트래픽에 적용하여 적절한 QoS를 제공할 수 있도록 연구하였다[8].

지금까지 연구된 트래픽 제어방식을 요약하면 우선권 제어방식과 버퍼 제어방식으로 요약되는데, 이들 방식을 각각 이용하여 손실민감 트래픽과 지연민감 트래픽 모두에게 적절한 QoS를 제공하기는 힘들 것으로 판단되고 따라서 복합적인 제어방식이 적절하다고 생각된다. 즉 시스템의 상태에 따라 각 트래픽에 서비스 우선권을 적절히 할당하여 패킷의 스케줄링을 제어하는 방식이 효율적인 것으로 판단된다. 이를 위하여 본 논문에서는 동적우선권 제어함수(Dynamic Priority Control Function, DPCF)  $f(\cdot)$ 의 개념을 도입하여 각 트래픽 클래스에 대한 스케줄링을 제어하도록 하였다. 다양한 스케줄링 방식을 DPCF에 적용하여 기존의 우선권 제어방식 등을 포괄적으로 해석할 수 있다.

본 논문에서는 음성과 같은 지연민감 트래픽의 성능을 개선하기 위하여 Threshold-based Bernoulli Priority Jump

(TBPJ) 방식의 DPCF를 제안하고, 이를 클래스 1의 손실민감 트래픽과 클래스 2의 지연민감 트래픽을 수용하는 스위칭속도(switching capacity)가  $m$ 인  $N \times N$  용량의 스위칭 시스템의 성능을 분석하였다. TBPJ를 적용한 시스템은 유한한 크기의 분리된 입력 버퍼를 가지고 무한크기의 출력 버퍼로 구성되어 있다. TBPJ 방식의 DPCF은 각 입력버퍼에 대기하고 있는 현재 패킷의 수와 입력버퍼의 무턱값(threshold)에 따라 각 트래픽 클래스에 우선권을 동적으로 할당하여 서비스가 이루어진다. TBPJ 방식의 성능을 분석한 결과 손실민감 트래픽의 성능저하는 미미하고 지연민감 트래픽의 성능향상은 현저히 개선된 것을 확인하였다.

서론에 이어 2장에서는 고속 패킷스위치의 시스템 모델, 트래픽 모델, TBPJ 방식에 대하여 기술하고, 3장에서는 패킷스위치 입력 버퍼의 큐잉모델 (queuing model)의 분석에 대하여 간략히 기술하고 TBPJ 방식의 성능이 우수함을 분석하였다. 마지막 4장에서 결론을 맺었다.

## II. 시스템 모델과 TBPJ 방식의 DPCF

### 2.1. 트래픽 Model과 고속 패킷스위치의 시스템 모델

TBPJ를 적용하는 시스템의 성능분석에 적용한 트래픽 모델은 다음과 같다; (i) 클래스 1의 손실민감 트래픽과 클래스 2의 지연민감 트래픽으로 분류한다. (ii) 패킷의 길이는 고정되어 있고, 매 슬롯에 대하여 도착률  $\lambda$ 는 Bernoulli 분포를 가지고 도착한다. 도착한 패킷이 클래스 1인 확률은  $r$  이고 클래스 2인 확률은  $(1-r)$ 이다. 즉 클래스 1 트래픽의 도착률은  $\lambda_1 = r\lambda$  이고 클래스 2 트래픽의 도착률은  $\lambda_2 = (1-r)\lambda$ 이다. (iii) 시스템의 모든 입력포트에 도착하는 패킷의 도착률은 동일하고 (즉 uniform 트래픽), 입력포트에 도착한 각 패킷의 목적지는 독립적(independent)이고 균등 (uniform)하게 분포한다고 가정한다.

본 논문에서 채택한 입출력 큐잉이 되고 스위칭속도 (switching capacity)가  $m$ 인  $N \times N$  용량의 고속 패킷스위치의 시스템 모델은 다음과 같다; (i) 스위치의 용량은  $N \times N$  이고 내부적으로 nonblocking 스위치이다. 또한 스위치는 슬롯 단위로 동기적으로 동작하고, 슬롯 시간은 하나의 패킷을 송신하는데 필요한 시간이다. (ii) 스위치의 스위칭속도  $m$ 은 하나의 슬롯 시간 내에 입력포트에서 동시에 출력포트로 전달되는 패킷의 수이다 (iii) 클래스  $i$  ( $i=1, 2$ ) 트래픽을 위한 입력버퍼는 분리되어 있고 크기는  $N_i$ 이다. 그리고 출력버퍼의 크기는 무한대이다. (iv) 각 입력포트에는 스케줄링된 패킷이 대기하는 head-of-line (HOL) 큐가 존재하고 하나의 패킷만 저장된다. (v) 입력포트의 HOL에서 대기하고 있는 패킷 가운데 동일한 목적지 포트에 향하는  $m$ 개를 임의로 선택하여 출력포트로 전송하고 그렇지 못한 패킷은 다음 슬롯에서 다시 출력포트로 이동하기 위하여 다른 입력포트의 패킷과 경쟁을 하게 된다. HOL에 대기하고 있는 패킷 가운데

데 클래스 1 패킷이 우선 순위가 높다 (vi) TBPJ 방식을 적용하기 위한 클래스  $i$  입력버퍼의 threshold(Queue Length Threshold, QLT)는  $T_i$  이다.

## 2.2. Threshold-based Bernoulli Priority Jump

방식의 Dynamic Priority Control Function

$i$ 개의 트래픽 클래스를 수용하는 시스템에서 하나의 패킷을 서비스 한 후 다음에 어느 클래스의 패킷에 우선권을 부여하여 서비스를 할 것인가를 결정하기 위한 스케줄링 함수를 DPCF  $f(\cdot) = f_i(X_1, X_2, \dots, X_i)$  라고 하였다. 여기에서  $X_j$ 는 트래픽 클래스  $j$ 에 대한 시스템의 상태(state)를 나타낸다 ( $j = 1, \dots, i$ ). 즉 DPCF  $f(\cdot)$ 는 시스템의 상태(state)에 따라 특정 트래픽 클래스에 우선권을 동적으로 할당하여 우선적으로 스케줄링 되도록 결정하는 상태의존 스케줄링 (state-dependent scheduling) 함수이다.

본 논문에서는 2개의 트래픽 클래스에 대해서 DPCF를 적용하기로 하고 Threshold-based Bernoulli Priority Jump (TBPJ)의 스케줄링 방식을 제안한다. TBPJ의 스케줄링은 다음과 같이 이루어진다. 클래스  $i$  ( $i = 1, 2$ ) 트래픽을 저장하는 크기가  $N_i$ 인 버퍼에 대하여 버퍼 threshold  $T_i$ 를 적용하여 DPCF 함수  $f(\cdot) = f_i(x, y)$ 를 다음과 같이 정의된다.

$$f_1(x, y, p, T_1, T_2) = \begin{cases} 1, & \text{if } x \neq 0 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$f_2(x, y, p, T_1, T_2) = \begin{cases} p, & \text{if } x \leq T_1, y \geq T_2, \\ & \text{where } p \text{ is BPJ probability} \\ 1, & \text{if } x = 0, y \neq 0 \\ 0, & \text{if } x \neq 0 \end{cases}$$

여기에서  $f_i$ 는 클래스  $i$ 의 DPCF이고,  $(x, y)$ 는 입력버퍼에 대기하고 있는 클래스 1, 2의 패킷 수이다. 위 스케줄링 함수  $f(\cdot)$ 의 동작은 다음과 같다. (i)  $x \neq 0$ 이면 (즉 클래스 1의 버퍼가 비어 있지 않으면) 기본적으로 클래스 1의 패킷에 우선권을 할당하여 HOL 큐에 우선적으로 스케줄링이 되도록 한다. 즉  $x \neq 0$ 인 경우 클래스 1 패킷의 스케줄링 확률은 1이다. (ii) 그러나 입력버퍼에 대기하고 있는 패킷의 수가  $x \leq T_1, y \geq T_2$ 인 경우는 클래스 2 입력버퍼의 가장 선두에 있는 패킷은 Bernoulli 확률  $p$ 로 priority jump (Bernoulli Priority Jump, BPJ) 하여 클래스 1 버퍼의 마지막으로 이동하여 클래스 1 패킷과 같이 HOL 큐로 우선하여 스케줄링 된다. (iii) 클래스 1 버퍼가 비어 있으면 (즉  $x = 0$ ), 클래스 2 패킷이 HOL 큐로 스케줄링 된다.

$f(\cdot) = f_i(x, y, p, T_1, T_2)$ 의 파라미터를 적절히 조정하면 다음의 제어방식이 된다.

(i) Head of line (HOL) 우선 스케줄링 방식 :

$$f(\cdot) = f_i(x, y, 0, 0, 0)$$

(ii) Bernoulli 스케줄링 방식 :

$f(\cdot) = f_i(x, y, p_i, T_1, T_2)$ 에서 priority jump는 없다.

$$f_1(x, y, p_1, T_1, T_2) = \begin{cases} p_1, & \text{if } x > 0, y \geq T_2 \\ 1, & \text{if } x > 0, y \leq T_2 \end{cases}$$

$$f_2(x, y, p_2, T_1, T_2) = \begin{cases} p_2, & \text{if } y > 0, x \geq T_1 \\ 1, & \text{if } y > 0, x \leq T_1 \end{cases}$$

(iii) Queue length threshold (QLT) 스케줄링 방식 :

$$f(\cdot) = f_i(x, y, 1, T_1, T_2)$$

## III. 성능분석 및 해석

TBPJ 방식의 DPCF에 의한 고속 패킷스위치의 성능을 분석하기 위한 단계는 다음과 같다. (i) 클래스  $i$  버퍼의 최초 패킷이 TBPJ 방식에 의하여 스케줄링 되어 HOL 큐로 이동하는데 필요한 시간 (슬롯시간)의 분포를 구한다. (ii) HOL 큐에 있는 클래스  $i$  패킷의 정상상태 (steady-state) 서비스 시간에 대한 확률분포를 구한다. (iii) 시스템의 동작을 HOL 큐에서 패킷이 서비스되는 시점을 기준으로 하여 embedded Markov chain을 구성하고, 이에 대한 정상상태의 확률 분포를 구한다. (iv) 임의의 슬롯 시간에 대한 정상상태의 확률분포를 구한다. (v) Bernoulli 입력 분포를 갖는 랜덤프로세스 (random process)는 임의의 패킷이 도착 시점에서의 시스템 상태가 정상상태에서의 시스템 상태와 동일하다는 BASTA (Bernoulli Arrivals See Time Average) [15] 성질을 이용하여 손실확률을 구한다. (vi) Little's law를 이용하여 평균대기시간을 구한다.

우선 HOL큐의 동작을 기술하기 위하여 패킷의 목적지가 특정한 출력포트  $j$ 인 경우에 대하여 큐잉시스템을 분석한다. 스위치의 용량이 매우 큰 경우 출력포트  $j$ 로 향하는 입력프로세스는 Poisson 프로세스로 모델링 된다 [14]. 유한한 입력 버퍼이므로 패킷의 도착시점에 버퍼가 차있는 경우 손실된다. 따라서 유효 (effective)한 도착률은 다음과 같다.

$$\bar{\lambda}_i = (1 - P_{Li}) \lambda_i \quad (2)$$

여기에서  $P_{Li}$  ( $i = 1, 2$ )는 클래스  $i$  트래픽의 손실 확률이다. 따라서 HOL큐에서 출력포트  $j$ 로 향하는 시스템은 무작위순서 (random order) 서비스방식을 가진 discrete-time  $M_1, M_2 / D / m$  큐잉 시스템으로 모델링 된다.

슬롯 경제에서 HOL 가상 큐에서의 state-transition 확률  $G = \{g(i, j, k, l)\}$ 는 다음의 2차원 Markov chain을 구성한다.  $(i, j)$ 에서  $i$ 는 클래스 1 패킷의 수,  $j$ 는 클래스 2 패킷의 수이다.

(i)  $i+j < m$  인 경우

$$g_{(i,j)(k,l)} = \begin{cases} \sum_{x=0}^{m-(i+j)} a_x \sum_{y=0}^{m-(i+j+x)} b_y, & \text{for } k=l=0 \\ \sum_{x=0}^{m-(i+j)+i} a_x b_{l-j+(m-(i+x))}, & \text{for } k=0, l \leq j \\ \sum_{x=0}^{m-i} a_x b_{l-j+(m-(i+x))}, & \text{for } k=0, l > j \\ a_{m+k-j} b_{l-j}, & \text{for } k \geq 1, l \geq j \\ 0, & \text{for } k \geq 1, l < j \end{cases}$$

(ii)  $i+j \geq m, i < m$  인 경우

$$g_{(i,j)(k,l)} = \begin{cases} 0, & \text{for } k=0, l < j-(m-i) \text{ or } k \geq 1, l < j \\ \sum_{x=0}^{l-(j-(m-i))} a_x b_{l-(j-(m-i))-x}, & \text{for } k=0, j-(m-i) \leq l < j \\ \sum_{x=0}^{m-i} a_x b_{l-j+(m-(i+x))}, & \text{for } k=0, l \geq j \\ a_{m-i+k} b_{l-j}, & \text{for } k \geq 1, l \geq j \end{cases}$$

(iii)  $i \geq m$  인 경우

$$g_{(i,j)(k,l)} = \begin{cases} a_{k+m-i} b_{l-j}, & \text{for } k \geq i-m, l \geq j \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

그리고 시스템에  $i$ 개의 클래스 1 패킷과  $j$ 개의 클래스 2 패킷이 있을 확률을  $\phi_{ij}$  라고 하면, 정상상태에서의 확률  $\Phi_i = (\phi_{i0}, \phi_{i1}, \phi_{i2}, \dots)$ ,  $\Phi = (\Phi_0, \Phi_1, \Phi_2, \dots)$  는 다음과 같이 구한다.

$$\Phi = \Phi \cdot G, \quad \Phi \cdot e = 1, \quad (4)$$

여기에서  $e = (1, 1, \dots)^T$  이다.

[13]에서 사용한 방법을 확장하여, HOL 가상 큐의 패킷이 특정한 출력포트로 송신되는데 소요되는 시간인 경쟁 (contention) 시간은 바로 HOL 가상 큐에 있는 패킷의 서비스시간이므로 다음과 같이 구한다. HOL 가상 큐에  $y$ 개의 클래스 1 패킷이 있는 상태에서 클래스 1 tagged 패킷의 서비스 시간이  $x$  라면 tagged 패킷의 residual 서비스 시간  $R_{x,y} (x \geq 1, y \geq 1)$  는 다음과 같다.

$$R_{1,y} = \begin{cases} 1, & \text{for } 1 \leq y \leq m \\ \frac{m}{y}, & \text{for } y > m \end{cases}$$

$$R_{x,y} = \begin{cases} 0, & \text{for } 1 \leq y \leq m, x \geq 2 \\ \frac{y-m}{y} \sum_{j=0}^{\infty} R_{x-1,y+j-1} a_j, & \text{for } y > m, x \geq 2 \end{cases} \quad (5)$$

클래스 1 tagged 패킷의 경쟁시간에 대한 확률분포  $S_1$  은 다음과 같다.

$$S_1(x) = \Pr[S_1=x]$$

$$= \sum_{y=1}^{\infty} R_{x,y} \cdot \Pr(\text{tagged 패킷이 도착한 후 HOL 가상 큐에 } y \text{ 개의 클래스 1 패킷이 존재})$$

$$= \sum_{y=1}^{\infty} R_{x,y} \sum_{j=0}^{y-1} \left\{ \sum_{i=0}^{\infty} \phi_{ij} \right\} a_{y-i-1} \quad (6)$$

클래스 2 패킷의 경쟁시간에 대한 확률분포  $S_2$  는 [5]에서 사용한 방법을 확장하여 구한다.

$$S_2(x) = \Pr[S_2=x]$$

$$= \sum_{y=1}^{\infty} \sum_{z=0}^{m-x-1} D_{x,y,z} \left\{ \sum_{i=0}^y \sum_{j=0}^{z-1} \phi_{ij} a_{z-i} b_{y-j-1} \right\} \quad (7)$$

여기에서  $D_{x,y,z}$  는  $z$ 개의 클래스 1 패킷과  $y$ 개의 클래스 2 패킷이 있는 상태에서 클래스 2 tagged 패킷의 residual contention 시간이  $x$  인 것을 표시하고 다음과 같다.

$$D_{1,y,z} = \begin{cases} 1, & \text{for } 1 \leq y+z \leq m \\ \frac{m-z}{y}, & \text{for } z \leq m, y+z > m \\ 0, & \text{for } z > m, y+z > m \end{cases} \quad (8)$$

$$D_{x,y,z} = \begin{cases} 0, & \text{for } z \geq mx \\ 0, & \text{for } y+z \leq m, x \geq 2 \\ \frac{y-(m-z)}{y} \sum_{i=0}^{\infty} \sum_{j=0}^{m(x-1)-1} a_j b_i D_{x-1,y-(m-i)+i,j}, & \text{for } x \geq 2, y+z > m, z < m \\ \sum_{i=0}^{\infty} \sum_{j=0}^{m(x-1)-(x-m)-1} a_j b_i D_{x-1,y+i,x-m+j}, & \text{for } x \geq 2, y+z > m, z \geq m \end{cases}$$

입력포트에 도착하는 패킷은 Bernoulli 분포이고 입력버퍼의 크기는  $N_1 + N_2 + 1$  이다. 그리고 패킷의 서비스 시간은 HOL 가상 큐에 있는 클래스  $i$  패킷의 경쟁시간이고 이것은 일반(general) 분포이다. 따라서 TBPJ 방식을 적용한 고속 패킷 스위치는  $\text{Geom}_1, \text{Geom}_2 / G_1, G_2 / 1 / N_1 + N_2 + 1$  큐잉시스템으로 모델링 된다.

패킷이 서비스 받고 떠나는 시간을 embedded point로 하여 2차원의 Markov chain을 구성하여 시스템을 분석한다. embedded point에서  $i$ 개의 클래스 1 패킷과  $j$ 개의 클래스 2 패킷이 있는 경우 시스템의 상태를  $(i, j)$  로 표시하고,  $\pi(i, j)$  는 시스템 상태가  $(i, j)$  일 확률이고, 시스템의 state-transition 확률을  $h(i, j; k, l)$  라고 하면  $\pi(i, j)$  는 다음에서 구할 수 있다.

$$\pi = \pi H, \quad \pi e = 1 \quad (9)$$

여기에서  $\pi = (\pi_0, \dots, \pi_{N_1})$ ,  $\pi_j = (\pi(j, 0), \dots, \pi(j, N_2))$ ,  $H = (h(i, j; k, l))$ .

Embedded point에서의 semi-Markov 프로세스를 이용하여 HOL 패킷의 서비스시간 동안에 도착하는 패킷의 수를 고려하여 임의의 슬롯시간에 (즉 steady-state에서) 시스템이  $(i, j)$  에 있을 확률  $p(i, j)$  를 구할 수 있다. 정상상태 확률  $p(i, j)$  를 구하면 BESTA (Bernoulli arrivals see time average, [15]) 성질을 이용하여 클래스  $i$  트래픽의 손실확률  $P_{Li}$  를 구할 수 있다. 즉, 클래스 1패킷이 도착하였을 때  $N_1+1$  개의 패킷이 있거나  $N_1$  개의 패킷과 HOL 큐에 클래스 2의 패킷이 있는 경우 손실된다. 그리고 클래스 2패킷의 경우는  $N_2+1$  개의 클래스 2패킷이 있거나  $N_2$  개의 클래스 2패킷과 HOL 큐에 클래스 1패킷이 있는 경우 손실된다. 이때 손실확률의 계산은 closed form으로 구할 수 없고 iteration 방법에 의해서 구해야 한다.

$$P_{L1} = \sum_{j=0}^{N_1} p_{N_1+1,j} + p_{N_1+1,N_2} + \sum_{j=1}^{N_1} \left\{ \sum_{k=1}^{N_1} \sum_{l=1}^{N_1} \overline{p}_{jk} f_2(j, k, p, T_1, T_2) \right\}_{i=N_1-k}^{\infty} \overline{N}_2 + \sum_{j=1}^i \overline{p}_{0j} \sum_{l=N_1}^{\infty} \overline{N}_2 + \overline{p}_{00} (1-r) \frac{\lambda \overline{S}_2}{1+\lambda \overline{S}_2} \sum_{l=N_1}^{\infty} \overline{N}_2 \quad (10)$$

$$P_{L2} = \sum_{j=0}^{N_2} p_{i,N_2+1} + p_{N_1+1,N_2} + \sum_{j=1}^{N_2} \left\{ \sum_{k=1}^{N_2} \sum_{l=1}^{N_2} \overline{p}_{jk} f_1(j, k, p, T_1, T_2) \right\}_{i=N_2-k}^{\infty} \overline{N}_1 + \sum_{j=1}^i \overline{p}_{j0} \sum_{l=N_2}^{\infty} \overline{N}_1 + \overline{p}_{00} r \frac{\lambda \overline{S}_1}{1+\lambda \overline{S}_1} \sum_{l=N_2}^{\infty} \overline{N}_1 \quad (11)$$

여기에서  $f_i(j, k, p, T_1, T_2)$  는 TBPI의 DPCF이다; 클래스 2패킷이 BPJ 방식에 의하여 클래스 1 버퍼로 PJ 하는 확률은 클래스 2의 버퍼에 대기하고 있는 패킷의 개수가  $T_2$  이상이고 클래스 1 버퍼에 대기하고 있는 패킷의 수가  $T_1$  이하인 경우에 Bernoulli 확률  $p$  로 PJ 하게 된다.

Little의 법칙을 이용하여 클래스  $i$  트래픽의 평균 대기시간  $\overline{W}_i$  는 다음과 같이 구한다.

$$\overline{W}_i = \frac{\sum_{j=0}^{N_i+1} j M_i}{(1-P_{Li}) \lambda_i} - 1 \quad (12)$$

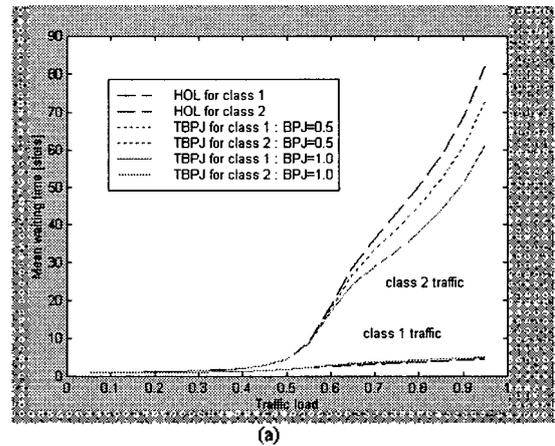
여기에서  $M_i$  는 marginal 확률로 클래스  $k(k=1, 2, \overline{k}=2, 1)$  인 경우에

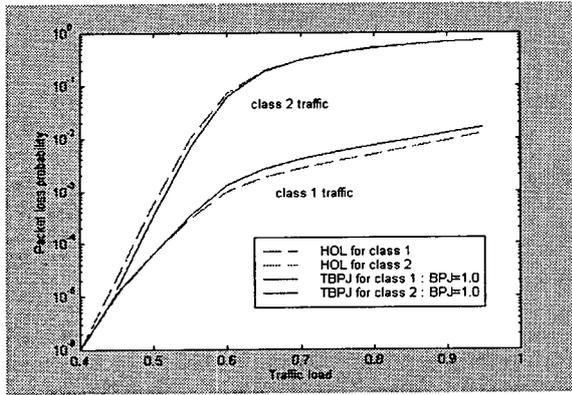
$$M_i = \begin{cases} \sum_{j=0}^{N_i+1} p_{ij}, & \text{for } 0 \leq i \leq N_{\overline{k}} \\ \sum_{j=0}^{N_i} p_{ij}, & \text{for } i = N_{\overline{k}} + 1 \end{cases} \quad (13)$$

64×64 용량의 스위치에 대하여 TBPI 방식의 성능을 확인하였다. 성능비교를 위하여 기존의 전형적인 제어방식인 HOL 방식과 성능을 비교하였다.

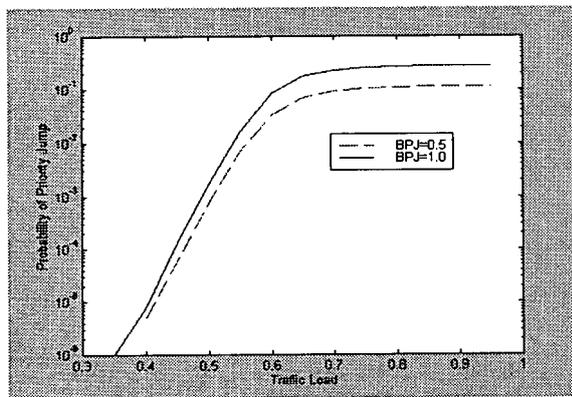
그림 1, 2, 3 에 TBPI 방식을 적용한 경우와 적용하지 않은 경우에 대하여 클래스  $i(i=1, 2)$  트래픽에 대한 손실확률, 대기시간, 클래스 2패킷의 priority jump (PJ) 확률을 비교하였다. TBPI를 적용한 경우 트래픽 용량  $\lambda$  에 대하여 버퍼 threshold  $T_i$ , 스위칭속도  $m$ , Bernoulli PJ (BPJ) 확률  $p$  에 대한 성능을 보여주고 있다. 스위치에 적용한 파라미터는 다음과 같다;  $N_1=7, N_2=10, \lambda_1=\lambda_2$ .

그림 1은  $T_1=1, T_2=7$  이고 BPJ 확률이  $p=0.5$  와  $p=1.0$  일 때 클래스  $i$  에 대한 손실확률, 대기시간, 클래스 2패킷이 클래스 1 버퍼로 PJ 하는 확률에 대한 결과를 보여주고 있다. 그림 1(b)에서 보여주는 것과 같이 클래스 1의 손실 확률은 TBPI 방식에 의하여 조금 증가하였고 클래스 2의 손실확률의 감소는 BPJ 확률이 1일 때 감소의 폭이 큰 것을 알 수 있다. 그러나 그림 1(a)에 보여주는 것과 같이 TBPI에 의해 클래스 1의 대기시간에 대한 성능저하는 미미하지만 클래스 2에 의한 성능 개선의 효과는 매우 크다는 것을 알 수 있다. 특히 이 경우에는 BPJ 확률  $p=0.5$  에서도 성능의 개선이 크게 나타나는 것을 알 수 있다. 특히 트래픽 용량  $\lambda > 0.6$  에서부터 TBPI에 의한 성능 개선이 두드러짐을 확인 할 수 있다. 그림 1(c)에 BPJ에 의해 클래스 2패킷이 클래스 1 버퍼로 PJ 되는 확률을 보여주고 있다. 낮은 트래픽 용량 ( $\lambda < 0.4$ ) 에서는 PJ가 발생되지 않고 어느 정도 트래픽이 증가하면 PJ도 증가 한다. 그리고  $\lambda > 0.6$  에서는 PJ 되는 확률이 크게 증가하여 클래스 2패킷의 대기시간이 크게 감소하는 것을 알 수 있다.





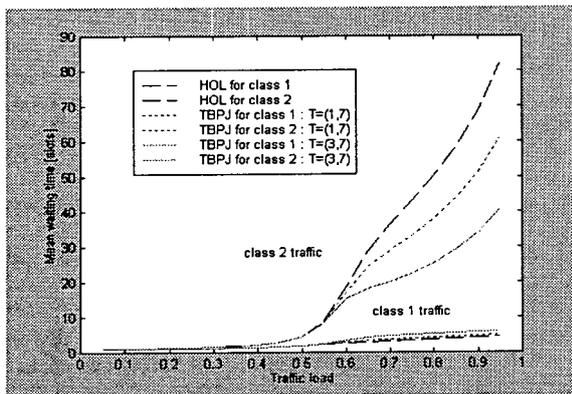
(b)



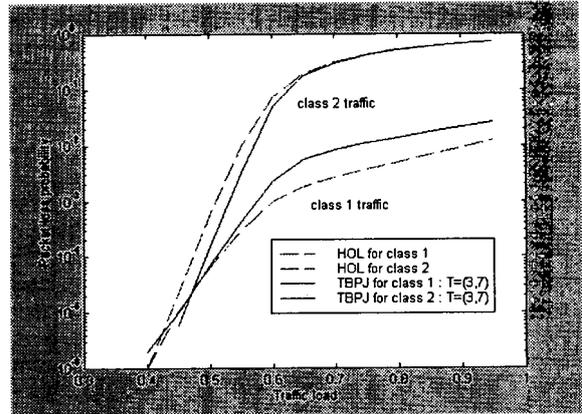
(c)

그림 1. Bernoulli 확률  $p$ 에 대한 TBPJ의 성능  
 (a) 평균대기시간 (b) 손실확률 (c) Priority jump 확률  
 Fig. 1. Performance of TBPJ for bernoulli probability  $p$ ,  
 (a) Mean waiting time, (b) Loss probability,  
 (c) Priority jump probability.

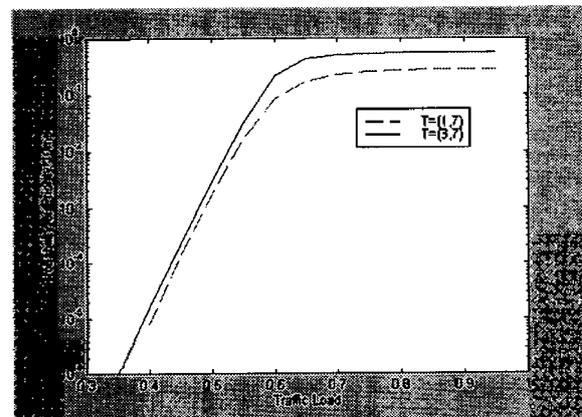
그림 2는 BPJ 확률  $p=1$  이고  $T_2=7$  인 경우에 threshold  $T_1$ 의 변화에 대한 TBPJ의 성능을 보여주고 있다. TBPJ의 성능은 전체적으로 그림 1의 결과와 유사한 것을 알 수 있다. 클래스 1버퍼의 threshold 값이  $T_1=1$ 에서  $T_1=3$ 로 증가하면 상대적으로 클래스 2패킷의 BPJ 확률이 증가되어 클래스 2패킷의 대기시간이 크게 감소됨을 알 수 있다.



(a)



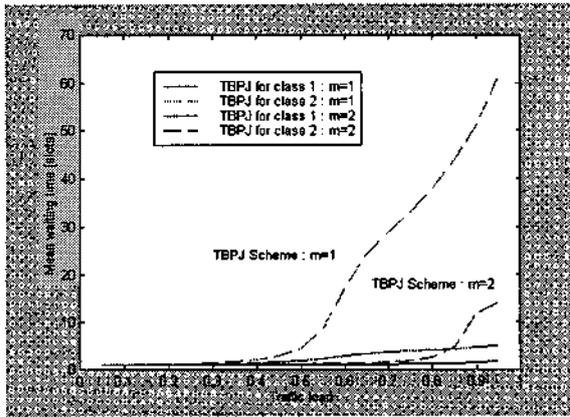
(b)



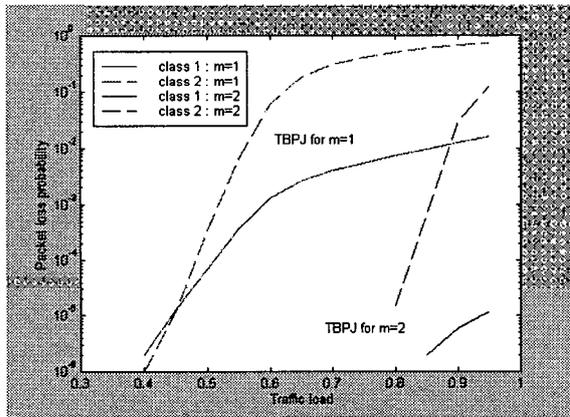
(c)

그림 2. 클래스 1 버퍼의 문턱값  $T_1$ 에 대한 TBPJ의 성능  
 (a) 평균대기시간 (b) 손실확률 (c) priority jump 확률  
 Fig. 2. Performance of TBPJ for buffer threshold  $T_1$ ,  
 (a) Mean waiting time, (b) Loss probability,  
 (c) Priority jump probability.

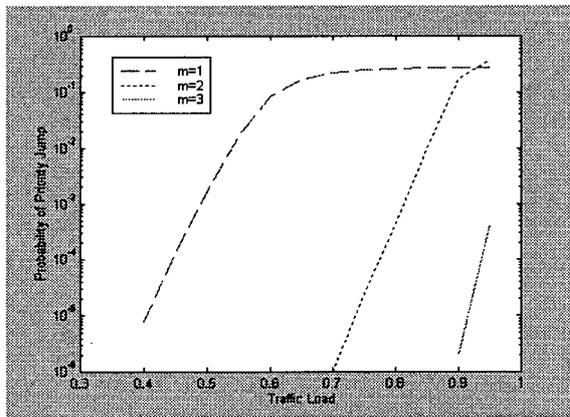
그림 3은 스위칭속도  $m$ 이 TBPJ에 미치는 영향을 보여주고 있다.  $m=2$ 인 경우에, 클래스 1트래픽은  $\lambda < 0.85$  이하에서는 손실이 없고 반면에 클래스 2트래픽에 대해서는  $\lambda < 0.8$  이하에서 손실이 없는 것을 알 수 있다.  $m=3$ 인 경우에는 패킷의 손실이 없다는 것을 확인할 수 있다. 특히  $m=3$ 인 경우에는 클래스 2패킷의 손실이 없으므로  $\lambda=0.9$ 에서부터 TBPJ에 의한 PJ가 발생된다. 따라서  $m \geq 3$ 인 경우에는 TBPJ에 의한 트래픽의 제어가 의미가 없다는 것을 알 수 있다. 또한  $m \geq 3$ 이면 패킷이 도착하고 다음 슬롯에서 바로 전송된다는 것을 알 수 있다. 즉 버퍼에서의 대기 시간이 거의 없다는 것을 알 수 있다. 이것으로부터 스위치를 구현하는 경우 스위칭속도는 2 정도가 되면 적절하다는 것을 의미한다.



(a)



(b)

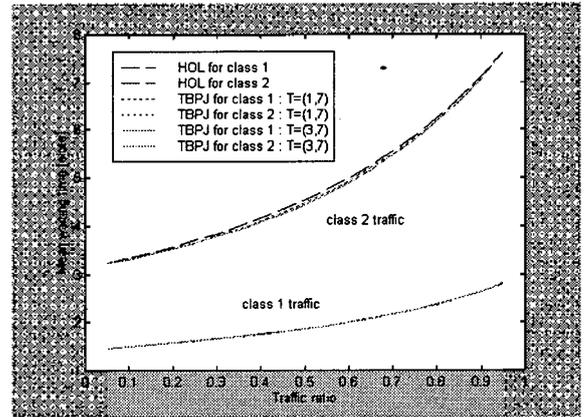


(c)

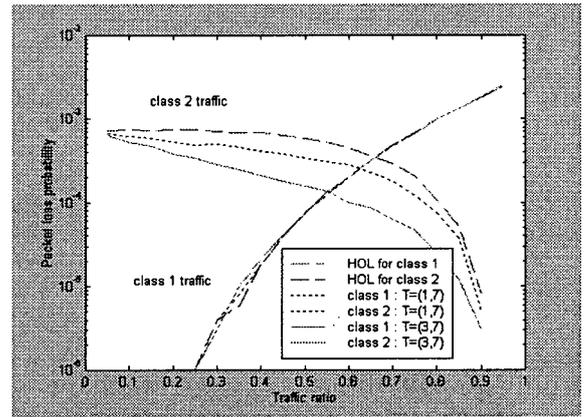
그림 3. 스위칭 속도  $m$  에 대한 TBPJ의 성능  
 (a) 평균대기시간 (b) 손실확률 (c) Priority jump 확률  
 Fig. 3. Performance of TBPJ for switching capacity  $m$ ,  
 (a) Mean waiting time, (b) Loss probability,  
 (c) Priority jump probability.

그림 4에서는  $\lambda=0.5$ 일때 트래픽 비율  $r$ 에 대한 TBPJ의 성능을 보여주고 있다.  $r$ 이 작은 경우에는 클래스 2 트래픽의 비율이 높고 그 결과 클래스 2패킷의 손실 확률은 매우 크고  $r$ 의 값이 증가함에 따라 클래스 1 트래픽의 손실확률이 증가한다. 따라서 트래픽의 비율  $r$ 에

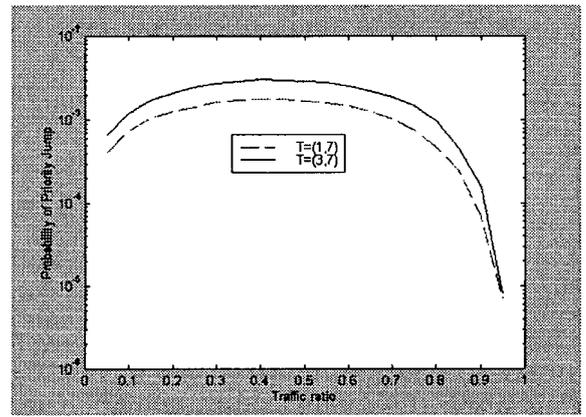
따라 적절한 크기의 버퍼가 요구되는 것을 알 수 있다.  $r$ 의 증가에 따라 대기시간 역시 증가하지만 TBPJ에 의한 클래스 1트래픽의 대기시간에는 거의 변화가 없지만 클래스 2트래픽의 경우는 조금씩 감소되는 것을 확인 할 수 있다.



(a)



(b)



(c)

그림 4. 트래픽 비율  $r$ 에 대한 TBPJ의 성능  
 (a) 평균대기시간 (b) 손실확률 (c) Priority jump 확률  
 Fig. 4. Performance of TBPJ for traffic ratio  $r$ ,  
 (a) Mean waiting time, (b) Loss probability,  
 (c) Priority jump probability.

또한  $r$ 이 증가함에 따라 클래스 2패킷이 클래스 1버퍼로 PJ하는 확률은 서서히 증가한다. 그러나  $r$ 의 증가로 인하여 클래스 1버퍼에 대기하고 있는 패킷의 수가 threshold  $T_1$  보다 많아지게 되어 결국은 PJ 되는 패킷의 수가 감소하는 것을 알 수 있다.

그림 5은 스위칭속도  $m$ 에 대하여 최대허용 트래픽용량 (maximum admissible load)  $\lambda_{max}$ 와의 관계를 보여주고 있다. 스위치에 적용한 파라미터는 다음과 같다;  $N_1=7, N_2=10, T_2=7, r=0.5, P_{L1} \leq 10^{-6}, P_{L2} \leq 10^{-11}, \overline{W}_1 \leq 1.0, \overline{W}_2 \leq 2.0$ . 그림 5에서 보여주는 것과 같이  $m=1$ 인 경우에 HOL 방식에서는  $\lambda_{max}=0.4$ 이지만 TBPI에서는  $\lambda_{max}=0.45$ 으로 향상된 것을 확인할 수 있다. 또한  $m=2$ 에서는  $\lambda_{max}=0.85$ 로 HOL 방식이나 TBPI 방식 모두 동일하다. 즉  $m \geq 2$ 인 경우에 어느 방식을 사용하든 거의 동일한 성능을 가진다는 것을 의미한다. 이것은 또한 앞의 결과에서와 같이 패킷스위치를 스위칭플랜으로 구현하는 경우 병렬로 (즉  $m=2$ ) 하면 적절하다는 것을 의미한다.

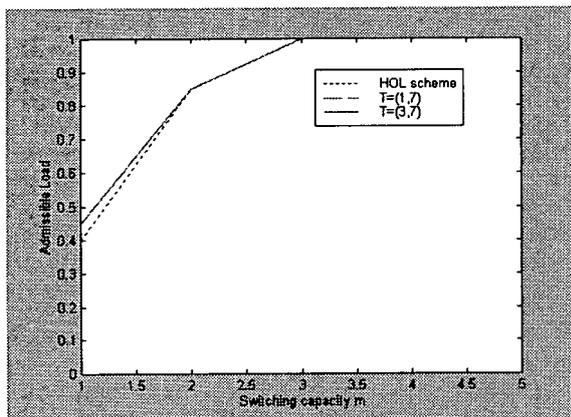


그림 5. 스위칭 속도  $m$ 에 대한 최대 허용 트래픽용량  
Fig. 5. Maximum admissible traffic load for switching capacity  $m$ .

#### IV. 결 론

본 논문에서는 시스템의 상태에 따라 음성과 같은 지연민감 트래픽의 서비스를 동적으로 하기 위한 TBPI 방식의 DPCF를 제안하였다. TBPI 방식은 기존의 우선권 제어방식을 더욱 확장한 방식으로 클래스 1의 손실민감 트래픽과 클래스 2의 지연민감 트래픽을 수용하는 시스템에 TBPI를 적용하여 성능을 분석하였다. 성능분석을 통하여 TBPI 방식이 기존의 우선권 제어 방식보다 성능 및 효율성에 있어서 우수함을 입증하였다. 즉 TBPI 방식을 적용한 결과 손실민감 트래픽의 성능저하는 미미하고 지연민감 트래픽의 성능향상은 현저히 개선된 것을 확인하였다.

DPCF를 이용한 트래픽제어방식은 인터넷을 통하여 음성정보를 서비스하는 경우에 적용할 수 있는 적절한 방식으로 생각된다. 음성과 데이터 트래픽이 균일 (uniform)한 경우에는 TBPI의 DPCF  $f_i(x, y, \rho, T_1, T_2)$ 을 모든 입력에 대해서 동일하게 적용하고, 만약 비균일 (nonuniform) 트래픽의 경우에는 각 입력에 대하여 서로 다른 스케줄링 제어함수  $f(\cdot)$ 을 적용하면 보다 더 효율적인 트래픽 제어가 가능할 것으로 기대된다.

#### 참 고 문 헌

1. R. Chipalkatti, J. F. Krose and D. Towsley, "Scheduling policies for real-time and nonreal-time traffic packet switching node," Proc. IEEE Infocom 1989, pp. 774-783, 1989.
2. H. M. Goldberg, "Analysis of the earliest due date scheduling rule in queuing systems," Math. Oper. Res., Vol. 2, pp. 145-154, 1977.
3. Y. Lim and J. E. Kobza, "Analysis of a delay-dependent priority discipline in an integrated multiclass traffic fast packet switch," IEEE Trans. Communications., Vol. 38, pp. 659-665, May 1990.
4. J. S. C. Chen and R. Guerin, "Performance study of an input queueing packet switch with two priority classes," IEEE Trans. Communications., Vol. 39, No. 1, pp. 117-126, Jan. 1991.
5. A. K. Gupta and N. D. Georganas, "Priority performance of ATM switches," Proc IEEE Infocom 1992, pp. 727-733, 1992.
6. H. Kroner, "Comparative performance study of space priority mechanisms for ATM networks," Proc IEEE Infocom 1990, pp. 1136-1143, 1990.
7. H. Kroner, G. Hebuterne, P. Boyer and A. Gravey, "Priority management in ATM switching nodes," IEEE J. Selected Areas Commun., Vol 9, pp. 418-427, April 1991.
8. D. Lee and B. Sengupta, "Queueing analysis of a threshold based priority scheme for ATM networks," IEEE/ACM Trans. Networking. Vol 1, No. 6, pp. 709-717, Dec. 1993.
9. Dokyu Kim, "Performance analysis of bandwidth allocation strategy with state-dependent Bernoulli access," KAIST Ph.D dissertation, 1993.
10. Dokyu Kim, "Performance analysis of bandwidth allocation strategy with state-dependent Bernoulli access and preemptive priority in wideband integrated networks," Telecommunication Systems. Vol 4, 1995.
11. Dokyu Kim, "Performance analysis of bandwidth allocation strategy with optimal access probabilities in wideband integrated networks," Journal of Electrical Engineering and Information Science. Vol 3, No 5, 1998.
12. J. Y. Lee, "Performance analysis of traffic control schemes in broadband ATM networks," KAIST Ph.D dissertation, 1995.
13. O. J. Boxma and W. P. Groenendijk, "Waiting times in discrete time cyclic-service system," IEEE Trans.

Commun., Vol. 36, pp. 164-170, Feb. 1988.

- 14. San-qi Li, "Nonuniform traffic analysis on a nonblocking space-division packet switch," IEEE Trans. Commun., Vol. 38, No. 7, pp. 1085-1096, July 1990.
- 15. M. J. Karol, M. G. Hluchyj and S. P. Morgan, "Input vs output queuing on a space-division packet switch," IEEE Trans. Commun., Vol. 35, pp. 1587-1597, Dec. 1988.

▲ 김도규 (Dokyu Kim)



1987년 2월 : 경북대학교 전자공학과 (학사)  
 1989년 2월 : 한국과학기술원 전기전자공학과 (석사)  
 1993년 2월 : 한국과학기술원 전기전자공학과 (박사)  
 1993년~1995년 2월 : 정보통신부 정보통신정책실 통신사무관

1995년~현재 : 성결대학교 컴퓨터학부 정보통신전공

▲ 김용규 (Yongkyu Kim)



1985년 2월 : 연세대학교 전자공학과 (학사)  
 1987년 2월 : 연세대학교 전자공학과 (석사)  
 1995년 8월 : 연세대학교 전자공학과 (박사)  
 1987년~1995년 9월 : 삼성전자

1996년~현재 : 성결대학교 컴퓨터학부 정보통신전공

▲ 조석팔 (Sokpal Cho)



1976년 2월 : 광운대학교 전자통신학과 (학사)  
 1987년 8월 : 한양대학교 전자통신학과 (석사)  
 1992년 8월 : 경희대학교 전자공학과 (박사)  
 1975년 : Control Data Corp.

Computer SE

1984년 : 삼성전자 정보통신연구소 연구실장(수석연구원)

1995년~현재 : 성결대학교 컴퓨터학부 정보통신전공