論文 94-19-7-12

IP용 서비스 요청 처리방식의 성능분석

正會員 崔 高 峰* 正會員 尹 鍾 浩** 正會員 權 奇 浩***

Performance Analysis of Request Handling Schemes for Intelligent Peripherals

Go Bong Choi*, Chong Ho Yoon**, Key Ho Kwon*** *Regular Members*

要 約

본 논문에서는 고도 지능망내의 한 물리적 실체로서 통신서비스 이용자와 망간의 각종 정보교환을 지원하는 Intellingent Peripheral(IP)에서의 서비스(특수지원) 요청들을 처리하는 방식들을 제안하고 각 방식들의 성능을 분석, 비교하였다. 여기서 IP시스템은 우선순위가 있는 서비스 요청과 일반순위의 서비스 요청을 blocked-call-delayed 방식으로 처리하는 큐잉시스템으로 모델링되었고, 서비스가 지연되는 요청들은 유한 크기의 버퍼에 저장된다고 가정하였다. 방식I은 우선순위가 있는 요청만이 저장될 수 있으며, 우선순위가 있는 요청을 위한 별도의 서버를 가지고 있다. 다른 3가지 방식(방식 II, III, IV)들은 우선순위가 있는 요청용 서버를 별도로 두지 않으며, 또한 모든 호들을 저장할 수 있도록 허용하는 대신에 버퍼가 차면 일반요청은 축출될 수 있도록 한 것이다. 이러한 네 가지 방식에 대하여 각각의 블록킹확률과 지연시간 분포를 수치적으로 구해본 결과, 별도의 서버를 두지 않는 방식들이 별도의 서버를 두는 방식 보다 블록킹확률면에서 유리함을 알 수 있었다. 또한, 별도의 서버를 두지 않는 방식들의 성능을 비교한 결과, 우선순위가 있는 요청은 first-in, first-out으로 일반요청은 last-in, first-out 방식으로 처리하는 것이 가장 유리함을 알 수 있었다.

ABSTRACT

This paper presents the service handling schemes of an intelligent peripheral(IP) which provides the service resource function as a physical entity in the intelligent network. Four service request handling schemes are compared for an IP which can handle both ordinary requests and prioritized requests on the blocked-call-delayed basis. Delayed requests are assumed to be stored in a finite storage buffer. Scheme-I exclusively allows prioritized requests to be stored and to use a fixed number of reserved servers. The other three schemes without reserved servers(Scheme-II, III,

* 한국전자통신연구소
Electronics Telecommunication Research Institute
** 한국항공대학교 항공통신정보공학과
Dept. of Telecommunications Eng., Hankuk Aviation Univ.
*** 성균관대학교 전자공학과
Dept. of Electronics, Sungkyunkwan Univ.

論文番號 : 9483
接受日字 : 1994年 3月 16日

and IV) allow both types of requests to be stored and prioritized requests pushout ordinary requests if the buffer is full. For these four schemes, the blocking probabilities and delay distributions of both types of requests are numerically obtained. From the numericall results, the schemes without reserved servers reduce the blocking probability of ordinary requests without a severe penalty on proritized requests. For three schemes without reserved servers, it is noted that prioritized requests should be served on the first in, first-out basis, and ordinary requests should be served on the last-in, first-out basis.

Ⅰ. 서 론

고도지능망의 여러가지 요소중 지능형 정보제공 시스템인 IP(Intelligent Peripheral)는 각종 지능망 서비스에 필요하지만 교환기등에서 제공할 수 없는 특수한 자원들을 제공한다. 물론, 이러한 자원들을 망에 제공할 수 있도록 교환기등을 구현할 수도 있겠지만 다양한 서비스자원과 교환기능의 결합성을 배제하므로써 서비스의 진화 및 창출에 따른 서비스 자원들의 변경 및 도입을 신속하고 융통성있게 실현할 수 있으며, 이것은 바로 고도 지능망의 개념에 부합된다.[1-5]

IP가 제공할 수 있는 대표적인 통신 서비스 자원으로서는 녹음안내장치 구동, 전화번호 수집 등의 단순한 기능에서부터 "문장-음성", "음성-문장"변환 기능, 화자 식별, 번호확인, 음성메뉴 등의 음성정보 처리는 물론, 화상정보 처리, 실시간 통역, 자연어 처리 등을 들 수 있다. 고도 지능망에서는 다양한 종류의 서비스가 출현 할 것이고, 일부 서비스는 다른 서비스에 비해 우선적으로 처리되어야 할 경우가 발생하고 각 서비스마다 사용되는 자원도 각각 다르게 나타난다. 이러한 복잡한 시스템에서 고려되어야 할 사항은 제한된 수의 이질적인 자원들을 요청하는 서비스 호들을 효과적으로 처리하여 시스템의 전체의 성능을 향상시키는 것이다. 따라서 본 논문에서는 우선순위가 있는호와 일반호에 대한 여러가지의 서비스 요청 처리 방식들을 제안하고 각각의 성능을 분석, 비교하였다.

본 논문의 구성을 살펴보면, 1장 서론에 이어 2장에서는 지능망서비스를 위한 자원의 요청절차 및 자원할당을 모델링하기 위해 IP 시스템의 기능과 구조를 간략하게 기술하고, 3장에서는 IP 시스템내의 요청처리 관련 모듈들을 M/M/C/K 큐잉시스템으로 모델링하며, 4장에서는 이 모듈들에 대한 효율적인 4가지의 서비스 요청처리방식들을 제안하고 성능을 분석한다. 분석결과에 대한 검토는 5장에 기술되며, 마지막으로 6장에 결론을 맺는다.

Ⅱ. IP 시스템의 기능 및 구조

2.1 고도 지능망 내에서의 IP 기능

IP는 음성합성, 음성안내, 화자검증, 디지트수집 및 검증등의 자원들을 제어, 관리함으로써 고도 지능망에서 다양한 서비스들을 제공하기 위한 SCP(Service Control Point)의 서비스 로직(service logic) 수행을 지원한다.[6] 즉, SCP의 서비스 로직에서 지능망 서비스를 위해 자원의 지원이 필요하다고 판단되면 해당 IP로 자원의 지원요청을 보내게 되고, 해당 IP는 SCP의 요청대로 SSP(Service Switching Point)의 전달망 지원하에 사용자와의 상호작용에 의해 자신의 자원을 이용해 서비스기능을 수행한 후 그 결과를 SCP에 보내 준다. 따라서, IP에 의해 수행될 여러 기능들은 다양한 지능망 서비스의 제공에 중요한 역할을 하게 될 것이다. 대개의 경우에 IP는 어떤 서비스를 상품화 하기 위해서 사용상 필요한 사용자와의 인터페이스 기능을 제공한다. 초기 고도지능망에서 예상되는 IP의 기능은 아래와 같이 크게 3가지 형태로 나눌 수 있다.

- 음성채널을 통한 음성, DTMF(Dial-Tone Multi-Frequency), 변조된 음성 등의 데이타 수신
- 미리 녹음된 안내 방송, 저장된 음성, 호출 신호음, 변조된 음성 등을 이용자에게 송출
- 음성 정보의 녹음, 문장-음성 전환, DTMF 디지트의 판별 및 수집, 수집된 정보를 SCP에 송출

이러한 3가지 형태의 기본 기능을 바탕으로하여 다양한 기능들이 제공될 수 있으며, 계속적으로 새로운 기능들이 추가되게 될 것이다.[7]

IP에는 위와 같은 서비스 측면 기능 이외에도 여러가지의 기본적인 기능이 필요하다. 즉, IP가 하나의 시스템으로 동작하기 위해서는 망 내의 타 요소들과

의 통신이 필요하고, 이를 위해서는 각종 프로토콜 처리 기능이 필요하다. 예를 들면, SSP와의 연결을 위해 ISDN 가입자-망 프로토콜, SCP와의 통신을 위해 공통선신호 프로토콜, 망 유지 보수를 위한 TMN 프로토콜등의 처리기능이 필요하다. 또한, IP는 하나의 독립 시스템으로 갖추어야 할 일반적인 기능, 즉 운영 체제, 운용 관리 및 유지 보수기능 등을 보유해야 하고, 여러가지 자원들을 효율적으로 관리할 수 있는 데이타베이스 관리 시스템 기능과 사용자로부터의 자원 사용 요청시 필요한 자원을 사용자에게 연결시켜 줄 수 있는 스위칭 기능 등이 요구된다. 현재 통신망에서 IP와 유사한 상용제품의 예를 들면, 음성정보처리를 위한 Digicom사의 DIREX와 같은 음성정보시스템[8][9]은 다수개의 음성정보채널을 동시에 제공해 줄 수 있다.

2.2 IP 시스템 구조

초기의 지능망에서는 IP가 SSP내에 포함될 경우도 있겠지만 본 논문에서는 고도 지능망에서 IP가 독립 시스템으로 존재하는 경우만을 취급한다. IP의 기능에 대해서도, 하나의 IP내에 여러 응용 분야(음성인식/합성, 화상정보 처리, 음성-문장 변환등)가 동시에 수용될 수 있고 각각의 IP가 하나의 응용 분야로 전문화 될 수도 있다. 이러한 다양한 요구 사항을 만족시키기 위해서는 여러 가지 형태의 IP 시스템이 공통적으로 가져야 될 특징을 추출하여 플랫폼 형태로 시스템을 설계하면 개념적으로 그림 1과 같은 구조가 된다. 이 그림에서 통신망정합 모듈은 IP를 지능망, 이동통신망, 데이타망 및 통신관리망 등에 접속시키기 위한 정합장치이고, 서비스제어 모듈의 역할은 통신망정합 모듈로 부터의 서비스 제공 요청을 접수하여 서비스 종류를 확인하고 그 서비스에 필요한 자원의 사용이 가능한가를 판단하여 서비스 제공이 불가능할 경우 통신망정합 모듈을 통해 SSP나 SCP로 서비스가 불가함을 알린다. 만약 서비스 제공이 가능할 경우 자원관리 및 서비스기능 모듈에게 통지하여 요청된 자원기능(녹음 안내, 문장-음성 변환등)이 수행되도록 한다. 이때 서비스 성격에 따라 자원관리 및 서비스 기능내의 여러 장치들을 반복적으로 이용할 경우도 발생한다.

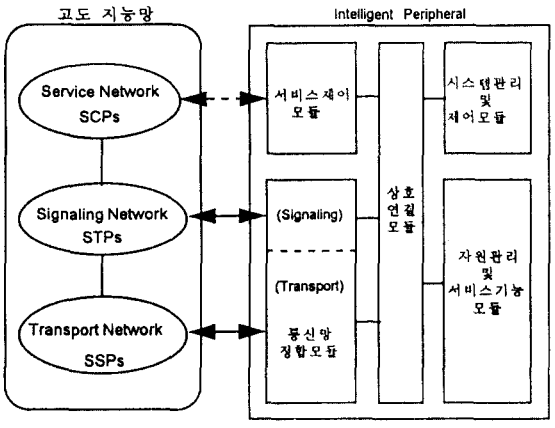

그림 1. IP 시스템 개념적 구조.
Fig 1. Conceptual Structure of IP System.

자원관리 및 서비스기능 모듈은 자원관리의 기능과 서비스 기능을 가지며, 우선 자원관리의 기능으로는 IP내의 각종 자원을 효율적으로 관리하기 위해 각 자원별로 필요한 데이타 베이스를 구축하고 외부의 서비스 요청시 관련 정보를 저장하거나 검색하고, 경우에 따라서 정보를 다른 형태나 압축된 형태로 가공하는 등의 기능을 수행한다. 또한, 서비스기능으로는 서비스제어 모듈로부터의 자원요청을 접수하여 필요한 장치를 구동한다. 즉, 요청된 정보들을 내부의 버퍼에 저장하고 정해진 서비스방식 및 우선 순위에 따라 처리한다. 이때, 서비스 요청을 처리하는 방식으로 FIFO(First-In, First-Out)방식과 LIFO(Last-In, First-Out)방식등을 생각할 수 있으며 이들 서비스방식과 성능분석에 대해서는 3장 및 4장에서 상세히 기술된다. 시스템 관리 및 제어모듈은 시스템 장애의 즉각적인 검출과 격리, 복구 등을 수행하고 과부하 등에 적절히 대응하기 위해 존재한다.

Ⅲ. IP 시스템의 모델링

IP 시스템은 SCP 혹은 SSP로 부터의 자원요청(IP 입장에서는 서비스요청)을 서비스제어 모듈에서 접수하여, 이후 자원관리 및 서비스기능 모듈에 있는 음성인식장치, 음성합성장치, DTMF 장치, 데이타정보처리장치, 화상정보처리장치 등을 구동한다. 이때 지능망서비스의 종류에 따라서 요구되는 자원이 각각 다르고, 하나의 지능망 서비스를 위해 SSP에서 IP로 특정 서비스(자원)을 중복적으로 요구하는 경우도 발생한다. 즉 IP 시스템이 어떤 지능망서비스를 위해 자원을 제공해준뒤 동일 지능망서비스를 위해 또 다른 자원의 요청을 받을 경우 이미 수행중인 지능망서비스가 중단되지 않으려면 새로운 지능망서비

스를 위해 도착하는 자원요청 보다 우선적으로 처리하여야 한다. 따라서 각 자원에 대한 요청은 일반요청과 우선요청으로 나누어져 다른 등급으로 처리되어야 할 필요가 있다. 또한, 보유하고 있는 특정자원보다 많은 자원이 동시에 요구되는 경우에 대응하기 위하여 IP의 자원관리 및 서비스 기능 모듈에는 자원요청을 일시 저장할 수 있도록 유한 크기의 버퍼를 내장하여, SSP로부터 도착한 자원요청들의 blocking 확률을 감소시킬 수 있도록 설계되어야 한다. 이때, 유한크기의 버퍼를 사용하는 이유는 무한버퍼인 경우에 발생할 수 있는 장시간의 지연시간을 방지하기 위함이다. 위의 사실을 종합하면 IP의 서비스제어 모듈과 자원관리 및 서비스기능 모듈은 우선순위가 있는 blocked-call-delayed 방식으로 요청을 처리하는 하나의 큐잉시스템으로 모델링할 수 있다.

또한 IP 시스템의 요청처리 방식에 따른 성능분석을 위한 본 논문에서 가정한 내용은 다음과 같다. 자원관리 및 서비스기능 모듈은 각 자원처리장치 마다 고정된 C개의 서버(server, 또는 채널)로 모델링 되며, 서비스제어 모듈은 C개의 서버가 모두 사용중인 경우 정보요청을 최대 $K-C$, $(K \geq C)$, 개 저장할 수 있는 버퍼로 모델링할 수 있다. 또한, 우선요청과 일반요청이 도착하는 분포는 각각 평균도착율이 λ_1과 λ_2인 Poisson분포에 따르며 모두 같은 평균처리시간인 $1/\mu$인 지수분포를 가진다. 그리고, P_h를 우선요청이 발생될 확률이라고 하면, 다음과 같이 λ_1은 λ_2에 비례하는 관계를 가진다고 가정한다.

$$\lambda_1 = P_h \cdot \lambda_2 \tag{1}$$

그러므로, 자원관리 및 서비스기능 모듈의 특정 자원처리장치에 요구되는 요청들의 총도착율 λ_T와 시스템에 부가되는 총 트래픽 부하 ρ_T는 각각 다음과 같이 주어진다.

$$\lambda_T = \lambda_1 + \lambda_2 \tag{2}$$

$$\rho_T = \lambda_T / C\mu \tag{3}$$

이러한 가정과 식에서 IP 시스템내에서의 특정자원에 대한 요청처리과정을 우선순위가 있는 하나의 $M/M/C/K$ 큐잉시스템으로 그림 2와 같이 모델링하여 해석할 수 있다.

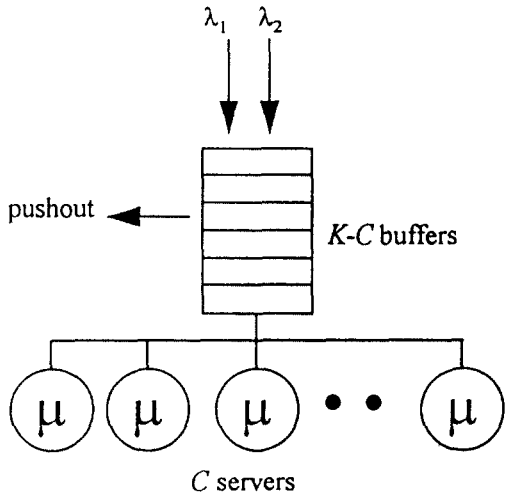

그림 2. IP 시스템의 큐잉 모델.
Fig 2. Queueing Model of IP System.

Ⅳ. 요청처리방식의 성능분석

4.1 우선요청용서버를 고정적으로 할당하는 방식(방식I)

이 방식하에서는, 우선요청은 Cg, $(Cg \leq C)$, 개의 우선요청용서버를 독점적으로 사용할 수 있다. 나머지$(C - Cg)$개의 서버는 모든 등급의 요청들이 공유할

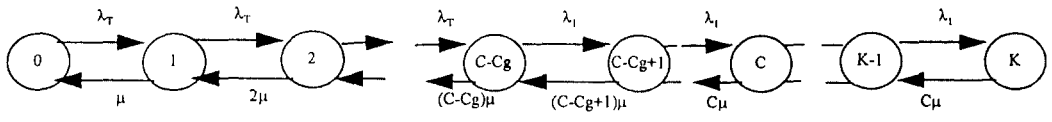

그림 3. 방식I의 상태천이도
Fig 3. State-transition-rate diagram for Scheme-I.

수 있다. 그러나 일반 요청은 현재 사용가능한 빈 서버의 갯수가 Cg개 이하이면, 차단된다. 만약 C개의 모든 서버가 사용중이면 $K-C$개의 우선요청은 요청 저장버퍼에 저장될 수 있으나, 일반요청은 저장되지 않는다. Hong과 Rapparport는 요청저장버퍼의 크기가 무한대인 경우에 이러한 방식을 해석하였다[10]. 그러므로, 본 연구에서는 그들의 결과를 유한크기의 버퍼에 대한 방식으로 수정하여 해석한다. 그림 3에 이 방식에 대한 상태천이도를 도시하였다. P_j를 이 시스템에 총 j개의 요청이 있을 확률이라고 하면, 상태천이도를 사용하면, P_j를 다음과 같이 구할 수 있다.

$$P_j=\begin{cases}\dfrac{(\lambda_1+\lambda_2)^j}{j!\mu^j}P_0 & \text{for } 1\le j\le C-C_g\\ \dfrac{(\lambda_1+\lambda_2)^{(C-C_g)}\lambda_1^{(j-C+C_g)}}{j!\mu^j}P_0 & \text{for } C-C_g+1\le j\le C\\ \dfrac{(\lambda_1+\lambda_2)^{(C-C_g)}\lambda_1^{(j-C+C_g)}}{C!\mu^C(C\mu)^{j-C}}P_0 & \text{for } C+1\le j\le C\end{cases} \quad (4)$$

여기서 P_0는 다음과 같이 주어진다.

$$P_0=\left[1+\sum_{j=1}^{C-C_g}\frac{(\lambda_1+\lambda_2)^j}{j!\mu^j}+\sum_{j=C-C_{g-1}}^{C}\frac{(\lambda_1+\lambda_2)^{(C-C_g)}\lambda_1^{(j-C+C_g)}}{j!\mu^j}+\sum_{j=C+1}^{K}\frac{(\lambda_1+\lambda_2)^{(C-C_g)}\lambda_1^{(j-C+C_g)}}{C!\mu^C(C\mu)^{j-C}}\right]^{-1} \quad (5)$$

도착하는 우선요청은 버퍼가 가득찬 경우에만 blocking되므로, 이것의 blocking확률 P_{B1}은 다음과 같다.

$$P_{B1}=P_K \quad (6)$$

또한, 일반요청의 blocking확률 P_{B2}는 시스템내의 요청갯수가 $C-Cg$ 이상인 확률의 합이므로 다음과 같이 구할 수 있다.

$$P_{B2}=\sum_{j=C-C_g}^{K}P_j \quad (7)$$

그리고, Little공식을 사용하면 성공적으로 처리되는 우선요청에 대한 평균대기시간은 다음과 같이 구할 수 있다.

$$E[W_1]=\sum_{j=C}^{K}\frac{(j-C)P_j}{\lambda_1(1-P_{B1})} \quad (8)$$

4.2 우선요청용 서버가 없는 개선된 요청처리방식 (방식II, III, IV)

본 연구에서 제안된 이 세가지 방식들(방식II, III, IV)은 우선요청용 서버를 별도로 두지 않으므로, 모든 요청은 빈 서버가 있으면 제한없이 처리된다. 그러나, 모든 서버가 사용중이면, 이들은 다음과 같이 제안된 처리방식에 따라 요청대기버퍼에 저장되거나 폐기된다.

4.2.1 처리방식들의 운용

(1) 방식II : 모든 서버가 사용중일때, 새로운 요청이 도착하면 이는 요청대기버퍼에 있는 요청중에서 가장 최근에 도착한 우선요청 뒤에 저장되거나, 버퍼내에 우선요청이 없으면 버퍼의 첫번째에 위치한다. 만약 일반요청이 시스템의 K번째(즉, 가득찬 버퍼의 제일 마지막 위치)에 있을 때, 새로운 요청이 도착하면, 이 일반요청은 버퍼에서 축출되고, 도착한 새로운 요청이 버퍼내에서 가장 최근의 우선요청 뒤에 저장된다. 시스템이 가득차고, 버퍼내에 모두 우선요청만 저장되어 있는 경우, 도착하는 모든 종류의 요청은 차단된다. 따라서, 우선 요청은 head-of-line 우선순위방식의 first-in, first-out(FIFO)으로 처리되며, 일반요청은 축출되는(pushout) LIFO 방식으로 처리되는 혼성처리방식이다. 이러한 방식과 비슷하지만 우선순위가 없거나 버퍼가 없는 경우에 대한 연구는 Doshi 및 Heyman에 의해 수행되었다[11][12].

(2) 방식III : 이 방식에서는 모든 서버가 사용중이면, 우선요청은 방식II처럼 처리된다. 그러나, 일반요청은 버퍼내의 가장 최근의 일반요청 뒤에 위치하거나, 버퍼가 비어있으면 버퍼의 첫번째에 위치한다. 시스템이 가득차 있을때 도착하는 일반요청은 차단된다. 그리고, 만약 현재의 시스템 크기가 K이고 일반요청이 시스템의 K번째(즉, 가득찬 버퍼의 제일 마지막 위치)에 있을 때, 우선요청이 도착하면, 이 요청은 버퍼에서 축출되고, 새로 도착한 우선요청이 버퍼내에서 가장 최근의 우선요청뒤에 저장된다. 시스템이 가득차고, 버퍼내에 모두 우선요청만 저장되어 있는 경우, 도착하는 모든 종류의 요청은 차단된다. 따라서, 우선요청은 방식II처럼 head-of-line 우선순위방식의 first-in, first-out(FIFO)으로 처리되지만, 일반요청은 FIFO 방식으로 처리되는 혼성처리방식이다.

(3) 방식IV : 이 방식에서는 방식 II와 III에서 처럼 모든 요청은 빈 서버가 있으면 제한없이 처리된다. 그러나, 모든 서버가 사용중이면, 우선요청은 요청대기버퍼에 요청중에서 가장 최근에 도착한 우선요청 앞에 저장되거나, 버퍼내에 우선요청이 없으면 버퍼의 첫번째에 위치한다. 또한, 일반요청은, 모든 서버가 사용중이면, 버퍼내의 가장 최근의 일반요청 뒤에 위치하거나, 버퍼가 비어있으면 버퍼의 첫번째에 위치한다. 시스템이 가득차 있을때 도착하는 일반요청은 차단된다. 그리고, 만약 현재의 시스템 크기가 K이고 일반요청이 시스템의 K번째(즉, 가득찬 버퍼의 제일 마지막 위치)에 있을 때, 우선요청이 도착하면, 이 요청은 버퍼에서 축출되고, 새로 도착한 우선요청이 버퍼내에서 가장 최근의 우선요청앞(즉, $C+1$번째 위치)에 저장된다. 시스템이 가득차고, 버퍼내에 모두 우선요청만 저장되어 있는 경우, 도착하는 모든 종류의 요청은 차단된다. 따라서, 우선요청은 head-of-line 우선순위방식의 last-in, first-out(LIFO)으로 처리되며, 일반요청은 축출되는(pushout) FIFO 방식으로 처리되는 혼성처리방식이다. 표 1에는 방식 II, III, IV에 대한 처리방식을 요약하였다.

4.2.2 분석

4.2.2.1 Blocking 확률

P_j, $(0 \le j \le K)$를 기본적인 $M/M/C/K$큐잉시스템에 j개의 요청이 있을 확률이고, $P_{j,k}$를 $j(j=0, 1,\ldots,K)$개의 요청이 시스템에 있으며 이들 중 $k(k=0, 1,\ldots,j-C)$개의 일반요청이 버퍼에 있을 확률이라고 정의하면, 제안된 방식 II, III, IV에 대한 상태천이도를 그림 4와 같이 얻을 수 있다. 각 상태는 $\delta(\cdot)$를 Dirac delta 함수라 했을 때, 다음과 같은 식으로 표현할 수 있다.

$$(\lambda_T + C\mu)P_{j,k} = \lambda_2 P_{j-1,k-1} + \lambda_1 P_{j-1,k} + C\mu P_{j+1,k} + C\mu P_{j+1,k+1} \cdot \delta(j-k-C), \text{ for } C < j < K,\ k \le j-C$$

$$C\mu P_{K,0} = \lambda_1 P_{K-1,0} + \lambda_2 P_{K,1} \quad (9)$$

$$(\lambda_1 + C\mu)P_{K,k} = \lambda_2 P_{K-1,k-1} + \lambda_1 P_{K-1,k} + \lambda_1 P_{K,k+1},$$

for $1 \le k \le K - C$

윗 식에서 $j-k<1$ 또는 $k<0$인 부분은 모두 0이며, $0 \le j \le C$이고 $k=0$인 부분은 $P_{j,0} = \rho_T^j/j!$이다.

하나의 우선요청은 시스템이 가득차고 버퍼에 일반요청이 없는 경우에만 차단되므로, 이들의 차단확률은 다음과 같이 얻을 수 있다.

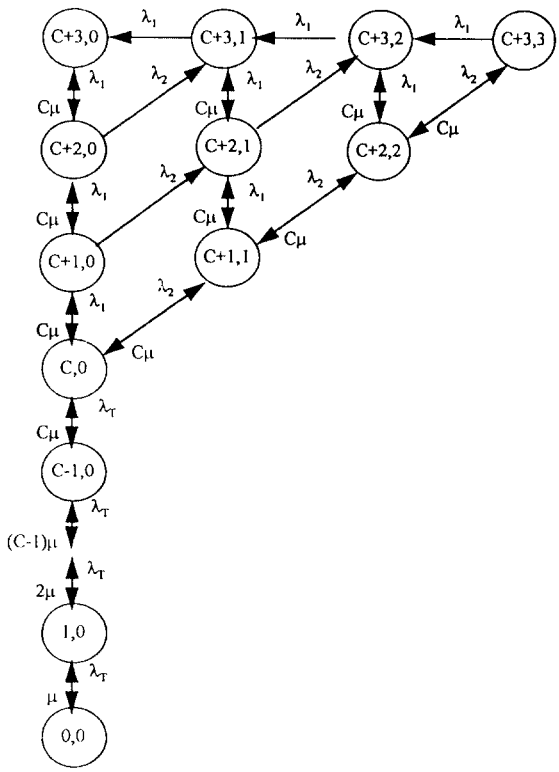

그림 4. 방식II에 의한 상태 천이도($K=C+4$인 경우).
Fig 4. State-transition-rate diagram for Scheme-II (for $K=C+3$).

표 1. 우선요청용 서버가 없는 처리 방식의 비교
Table 1. Comparison of Request Handing Schemes(Scheme-II, III, IV)

방식	우선순위가 있는 요청	일반 요청
II	FIFO with head-of-line priority basis	LIFO with pushout basis
II	FIFO with head-of-line priority basis	FIFO with pushout basis
IV	LIFO with head-of line priority basis	FIFO with pushout basis

$$P_{B1}=P_{K,0} \tag{10}$$

그리고, 일반요청의 차단확률, P_{B2}는 도착시의 차단율과 다음에 도착하는 요청으로 인한 버퍼내에서의 차단확률과의 합이다. 이것은 $\lambda_T P_K=\lambda_1 P_{B1}+\lambda_2 P_{B2}$라는 관계식으로 부터 다음과 같이 구할 수 있다.

$$P_{B2}=\frac{(\lambda_T P_K-\lambda_1 P_{B1})}{\lambda_2} \tag{11}$$

4.2.2.2 대기시간

성공적인 서비스를 받게되는 각 종류의 요청들의 대기시간확률분포함수를 구한다.

(1) 방식 II

우선요청은 $C\mu$의 처리율을 가지는 pure-death 프로세스 방식으로 처리되므로, 우선요청의 대기시간 확률분포함수는 다음과 같이 구할 수 있다.

$$W_1(0)=\sum_{j=0}^{C-1} P_j/(1-P_{B1})$$

$$W_1(t)=W_1(0)+\Big[\sum_{j=C}^{K-1}\sum_{k=0}^{j-C} P_{j,k}\int_{0^+}^{t} \frac{C\mu(C\mu x)^{j-C-k}}{(j-C-k)!}\,e^{-C\mu x}dx \tag{12}$$

$$+\sum_{k=1}^{K-C} P_{K,k}\int_{0^+}^{t}\frac{C\mu(C\mu x)^{K-C-k}}{(K-C-k)!}\,e^{-C\mu x}dx\Big]/(1-P_{B1}),\ t>0$$

식 (12)의 두번째 항은 가득차지 않은 시스템의 j번째에 우선요청이 도착하여 $j-C-k+1(C\le j<K)$번째 버퍼위치에 저장되는 경우에 대한 것으로서, 결과적으로 k개의 일반요청의 위치는 도착한 우선요청 뒤로 재조정된다. 세번째 항은 가득찬 시스템에 우선요청이 도착하여, $K-C-k+1,\ k>0$, 인 버퍼위치에 저장되는 경우에 대한 것으로서, 결과적으로 $k-1$개의 일반요청은 자신의 위치가 도착한 우선요청 뒤로 재조정되며, 제일 끝에 있던 일반요청은 시스템에서 축출(pushout)된다.

식 (12)는 다음과 같은 식을 사용하면, 보다 간편하게 표현될 수 있다. 즉,

$$\int_t^{\infty}\frac{m(mx)^k}{k!}\,e^{-mx}\,dx=\sum_{i=0}^{k}\frac{(mt)^i}{i!}\,e^{-mt} \tag{13}$$

이므로,

$$W_1(t)=W_1(0)+$$

$$\Big[\sum_{j=C}^{K-1}\sum_{k=0}^{j-C} P_{j,k}\Big(1-\int_t^{\infty}\frac{C\mu(C\mu x)^{j-C-k}}{(j-C-k)!}\,e^{-C\mu x}\,dx\Big)$$

$$+\sum_{k=1}^{K-C}\Big(1-\int_t^{\infty}\frac{C\mu(C\mu x)^{K-C-k}}{(K-C-k)!}\,e^{-C\mu x}\,dx\Big)\Big]/(1-P_{B1})$$

$$=1-\Big[\sum_{j=C}^{K-1}\sum_{k=0}^{j-C} P_{j,k}\sum_{i=0}^{j-C-k}\frac{(C\mu t)^i}{i!}\,e^{-C\mu t} \tag{14}$$

$$+\sum_{k=1}^{K-C} P_{K,k}\sum_{i=0}^{K-C-k}\frac{(C\mu t)^i}{i!}\,e^{-C\mu t}\Big]/(1-P_{B1}),\ t>0$$

이다.

성공적으로 처리되는 일반요청의 대기시간확률분포함수를 구하기 위하여, i번째 도착한 일반요청을 생각해 본다. 이 일반요청은 분명히 $C\mu/(\lambda_T+C\mu)$의 확률로 $(i-1)$번째의 새로운 위치로 이동되거나, $\lambda_T/(\lambda_T+C\mu)$의 확률로 $(i+1)$번째의 새로운 위치로 이동된다. 몇번의 이동후에 C개의 서버중 한 서버에서 성공적으로 처리되거나, $K+1$번째로 이동하여 차단될 수 있다. 일반요청이 i번째 위치에 도착하여 n번의 이동후 결국 성공적으로 처리되는 확률을 $u_{i,n}$이라고 하자. 이것은 C와 $K+1$에 흡수경계가 있는 랜덤워크(random walk)해석방법을 이용하면 다음과 같이 구할 수 있다[13].

$$u_{i,n}=\sum_{k=0}^{\infty}\Big(\frac{q}{r}\Big)^{ka}\cdot w_{2ka+(i-C),n}$$

$$-\sum_{k=1}^{\infty}\Big(\frac{q}{r}\Big)^{ka-(i-C)}\cdot w_{2ka-(i-C),n} \tag{15}$$

여기서,

$$q=\lambda_T/(\lambda_T+C\mu),$$

$$r=1-q,$$

$$a=K+1-C, \tag{16}$$

$$w_{i,n}=\frac{i}{n}\cdot\binom{n}{(n+i)/2}\cdot q^{(n-i)/2}\cdot r^{(n+i)/2},\ n\ge i$$

이고, $w_{i,n}$의 i와 n은 반드시 같은 parity를 가져야 한다.

또한, $g_{i,t}$를 i번째 위치에 도착한 일반요청이 t시간 이후에 성공적으로 처리될 확률이라고 하자. 버퍼내에서 n번의 이동후에 성공적으로 처리될 시간은 $(n-1)$

차수의 감마분포를 가지므로 다음과 같이 구해질 수 있다.

$$g_{i,t}=\sum_{n=1}^{x} u_{i,n}\cdot\frac{(\lambda_T+C\mu)^n t^{n-1}}{(n-1)!}\,e^{-(\lambda_T+C\mu)t},\ C<i\le K \tag{17}$$

시스템의 크기가 $j(j<K)$이고 $k(k\ge 0)$개의 일반요청이 버퍼에 있을 때, 새로 도착하는 일반요청은 $j-k+1$에 위치한다. 만약 현재의 시스템 크기가 K이고 $k(k\ge 0)$개의 일반요청이 버퍼에 저장되어 있을 때, 새로 도착하는 일반요청은 버퍼에서 가장 오래된 일반요청을 축출시키고 자신은 $K-k+1$에 위치한다. 그러므로, 성공적으로 처리되는 일반요청의 대기시간확률분포함수는 다음과 같이 구할 수 있다.

$$W_2(t)=W_2(0)+$$

$$\frac{\sum_{j=C}^{K-1}\sum_{k=0}^{j-C}P_{j,k}\int_{0^+}^{t}g_{j-k+1}dx+\sum_{k=1}^{K-C}P_{K,k}\int_{0^+}^{t}g_{K-k+1,x}dx}{1-P_{B2}},$$

$$t>0 \tag{18}$$

여기서, $W_2(0)=\sum_{j=0}^{C-1}P_{j,0}/(1-P_{B2})$이다.

식 (13)을 이용하면, 식 (18)은 다음과 같이 간략화 된다.

$$W_2(t)=1-$$

$$\Big[\sum_{j=C}^{K-1}\sum_{k=0}^{j-C}P_{j,k}\sum_{n=1}^{x}u_{j-k+1,n}\sum_{i=0}^{n-1}\frac{((\lambda_I+C\mu)t)^i}{i!}\cdot e^{-(\lambda_T+C\mu)t}$$

$$+\sum_{k=1}^{K-C}P_{K,k}\sum_{n=1}^{x}u_{K-k+1,n}\sum_{i=0}^{n-1}\frac{((\lambda_I+C\mu)t)^i}{i!}\cdot e^{-(\lambda_T+C\mu)t}\Big] \tag{19}$$

$$/(1-P_{B2}),\ t>0$$

(2) 방식 III

우선요청은 방식II와 같은 pure-death 프로세스 방식으로 처리되고, 우선요청의 대기시간은 일반요청의 처리방식과 무관하므로, 우선요청의 대기시간 확률분포는 식 (12)와 같다. 그러나, 일반요청의 대기시간확률분포는 다음에 구해지는 것처럼 방식II의 것과 다르다. 성공적으로 처리되는 일반요청의 대기시간확률분포함수를 구하기 위하여, i번째 도착한 일반요청을 생각해 본다. 새로 도착하는 우선요청만이 i번째 있는 일반요청의 위치를 $i+1$로 천이 시킬 수 있으므로, 이 일반요청은 분명히 $C\mu/(\lambda_1+C\mu)$의 확률로 $(i-1)$번째의 새로운 위치로 이동되거나, $\lambda_1/(\lambda_1+C\mu)$의 확률로 $(i+1)$번째의 새로운 위치로 이동된다. 몇번의 이동후에 C개의 서버중 한 서버에서 성공적으로 처리되거나, $K+1$번째로 이동하여 결국 차단(pushout)될 수 있다. 일반요청이 i번째 위치에 도착하여 n번의 이동후 결국 성공적으로 처리되는 확률을 $u_{i,n}$이라고 하였으므로, 방식 II의 경우와 마찬가지로 C와 $K+1$에 흡수경계가 있는 랜덤워크(random walk)해석방법을 이용하면 다음과 같이 구할 수 있다.

$$u_{i,n}=\sum_{k=0}^{x}\left(\frac{q}{r}\right)^{ka}\cdot w_{2ka+(i-C),n}$$

$$-\sum_{k=1}^{x}\left(\frac{q}{r}\right)^{ka-(i-C)}\cdot w_{2ka-(i-C),n} \tag{20}$$

여기서,

$$q=\lambda_1/(\lambda_1+C\mu),$$

$$r=1-q,$$

$$a=K+1-C, \tag{21}$$

$$w_{i,n}=\frac{i}{n}\cdot\binom{n}{(n+i)/2}\cdot q^{(n-i)/2}\cdot r^{(n+i)/2},\ n\ge i$$

이고, $w_{i,n}$의 i와 n은 반드시 같은 parity를 가져야 한다.

즉, 방식 II와 다른 점은 새로 도착하는 우선요청만이 i번째 있는 일반요청의 위치를 $i+1$로 천이시킬 수 있으므로, 일반요청이 i번째 위치에 도착하여 n번의 이동후 결국 성공적으로 처리되는 확률 $u_{i,n}$는 식 (15)의 q 항만 수정하면 구해질 수 있다.

또한, i번째 위치에 도착한 일반요청이 t시간 이후에 성공적으로 처리될 확률이 $g_{i,t}$을 구해보자. 버퍼내에서의 천이횟수는 우선요청의 도착과 처리율에만 관계되고 버퍼내에서 n번의 이동후에 성공적으로 처리될 시간은 $(n-1)$차수의 감마분포를 가지므로, 식 (17)을 이용하면, 다음과 같이 구해진다.

$$g_{i,t}=\sum_{n=1}^{x}u_{i,n}\cdot\frac{(\lambda_1+C\mu)^n t^{n-1}}{(n-1)!}\,e^{-(\lambda_1+C\mu)t},\ C<i\le K \tag{22}$$

시스템의 크기가 j, $j<K$, 일때 도착한 일반요청은 방식 II와 달리, 현재 버퍼에 있는 일반요청의 갯수에 무관하게 $j+1$의 위치에 저장된다. 그러나 시스템이 가득차 있는 경우는 물론 차단된다. 그러므로, 성공적으로 처리되는 일반요청의 대기시간확률분포함수는 식(18)을 사용하여 다음과 같이 구할 수 있다.

$$W_2(t)=W_2(0)+\frac{\sum_{j=C}^{K-1}\sum_{k=0}^{j-C}P_{j,k}\int_{0+}^{t}g_{j+1,x}\,dx}{1-P_{B2}},\quad t>0 \tag{23}$$

여기서, $W_2(0)=\sum_{j=0}^{C-1}P_{j,0}/(1-P_{B2})$이다.

식 (13)을 이용하면, 식 (23)은 다음과 같이 간략화 된다.

$$W_2(t)=1-\left[\sum_{j=C}^{K-1}\sum_{k=0}^{j-C}P_{j,k}\sum_{n=1}^{\infty}u_{j+1,n}\sum_{i=0}^{n-1}\frac{((\lambda_1+C\mu)t)^i}{i!}\cdot e^{(\lambda_1+C\mu)t}\right]/(1-P_{B2}),\ t>0 \tag{24}$$

(3)방식IV

성공적인 서비스를 받게되는 각 종류의 요청들의 대기시간확률분포함수를 구한다. 우선요청은 방식 II와 III에서의 FIFO방식이 아닌 LIFO방식으로 처리되지만, 이러한 LIFO방식은 방식 II와 III에서 일반요청이 처리되는 방식과 유사하다. 서버가 모두 사용중인 경우에 도착하는 우선요청은 반드시 $C+1$의 위치에 도착하고, 현재의 위치에 있던 다른 요청들의 위치를 이동시킨다. $C+1$위치에 도착한 우선요청의 이동은 자신 보다 이후에 도착하는 우선요청만에 의해서 뒤로 밀려난다. 즉, i번째에 저장된 우선요청은 분명히 $C\mu/(\lambda_1+C\mu)$의 확률로 $(i-1)$번째의 새로운 위치로 이동되거나, $\lambda_1/(\lambda_1+C\mu)$의 확률로 $(i+1)$번째의 새로운 위치로 이동된다. 몇번의 이동후에 C개의 서버중 한 서버에서 성공적으로 처리되거나, $K+1$번째로 이동하여 결국 차단(pushout)될 수 있다. 우선요청이 $C+1$번째 위치에 도착하여 n번의 이동후 결국 성공적으로 처리되는 확률을 $u_{C+1,n}$이라고 하면, 방식 II와 III의 일반요청이 처리되는 경우에 대한 해석방법과 같이 마찬가지로 C와 $K+1$에 흡수경계가 있는 랜덤워크(random walk)해석방법을 이용하면 다음과 같이 구할 수 있다.

$$uc_{+1,n}=\sum_{k=0}^{\infty}\left(\frac{q}{r}\right)^{ka}\cdot w_{2ka+1,n}-\sum_{k=1}^{\infty}\left(\frac{q}{r}\right)^{ka-1}\cdot w_{2ka-1,n} \tag{25}$$

여기서,

$$q=\lambda_1/(\lambda_1+C\mu),\quad r=1-q,\quad a=K+1-C, \tag{26}$$

$$w_{i,n}=\frac{i}{n}\cdot\binom{n}{(n+i)/2}\cdot q^{(n-i)/2}\cdot r^{(n+i)/2},\ n\geq i$$

이고, $w_{i,n}$의 i와 n은 반드시 같은 parity를 가져야 한다.

또한, $C+1$번째 위치에 도착한 우선요청이 t시간 이후에 성공적으로 처리될 확률이 $g_{C+1,t}$를 구해보자. 버퍼내에서의 천이횟수는 우선요청의 도착과 처리율에만 관계되고 버퍼내에서 n번의 이동후에 성공적으로 처리될 시간은 $(n-1)$차수의 감마분포를 가지므로, 식 (17)을 이용하면, 다음과 같이 구해진다.

$$g_{C+1,t}=\sum_{n=1}^{\infty}u_{C+1,n}\cdot\frac{(\lambda_1+C\mu)^n t^{n-1}}{(n-1)!}e^{-(\lambda_1+C\mu)t},\ C<i\leq K \tag{27}$$

시스템의 크기가 j, $j<K$ 일 때, 도착한 우선요청은 방식 II와 III과 달리, 현재 버퍼에 있는 요청의 갯수에 무관하게 $C+1$의 위치에 저장된다. 그러나 버퍼가 우선요청으로 가득차 있는 경우는 물론 차단된다. 그러므로, 성공적으로 처리되는 우선요청의 대기시간확률분포함수는 식 (18)을 사용하여 다음과 같이 구할 수 있다.

$$W_1(t)=W_1(0)+\frac{(1-\sum_{j=0}^{C-1}P_{j,0})\int_{0^+}^{t}g_{C+1,x}\,dx}{1-P_{B1}},\quad t>0 \tag{28}$$

여기서, $W_1(0)=\sum_{j=0}^{C-1}P_{j,0}/(1-P_{B1})$이다. 식 (13)을 이용하면, 식(28)은 다음과 같이 간략화 된다. 즉, $1-P_{B1}=\sum_{j=0}^{C-1}P_j+(1-\sum_{j=0}^{C-1}P_j)\sum_{n=0}^{\infty}u_{C+1,n}$이므로,

$$W_1(t)=\frac{\sum_{j=0}^{C-1}P_j}{1-P_{B1}}$$

$$+\frac{(1-\sum_{j=0}^{C-1}P_j)\int_0^t\sum_{n=1}^{\infty}u_{C+1,n}\cdot\frac{(\lambda_1+C\mu)^n}{(n-1)!}x^{n-1}\cdot e^{-(\lambda_1+C\mu)x}\,dx}{1-P_{B1}}$$

$$=\frac{\sum_{j=0}^{C-1}P_j}{1-P_{B1}}$$

$$+\frac{(1-\sum_{j=0}^{C-1}P_j)\sum_{n=1}^{\infty}u_{C+1,n}\cdot(1-\sum_{i=0}^{n-1}\frac{((\lambda_1+C\mu)t)^i}{i!}\cdot e^{-(\lambda_1+C\mu)t})}{1-P_{B1}} \quad (29)$$

$$=1-\frac{(1-\sum_{j=0}^{C-1}P_j)\sum_{n=1}^{\infty}u_{C+1,n}\cdot\sum_{i=0}^{n-1}\frac{((\lambda_1+C\mu)t)^i}{i!}\cdot e^{-(\lambda_1+C\mu)t}}{1-P_{B1}}$$

성공적으로 처리되는 일반요청의 대기시간확률분포함수를 구하기 위하여, i번째 도착한 일반요청을 생각해 본다. 새로 도착하는 우선요청만이 i번째 있는 일반요청의 위치를 $i+1$로 천이시킬 수 있으므로, 이 일반요청은 분명히 $C\mu/(\lambda_1+C\mu)$의 확률로 $(i-1)$번째의 새로운 위치로 이동되거나, $\lambda_1/(\lambda_1+C\mu)$의 확률로 $(i+1)$번째의 새로운 위치로 이동된다. 몇번의 이동후에 C개의 서버중 한 서버에서 성공적으로 처리되거나, $K+1$번째로 이동하여 결국 차단(pushout)될 수 있다. 그러므로, 성공적으로 처리되는 일반요청의 대기시간 확률분포함수는 방식III의 식 (24)와 같다.

V. 결 과

본 논문에서는 제안된 요청처리방식들에 대한 수치적인 비교를 위하여 다음과 같은 성능변수를 설정하였다. $K=30$, $C=20$, 그리고 시간은 한 요청의 평균지속시간$(1/\mu)$으로 일반화 시켰다.

그림 5와 6에는 방식I과 II, III, IV에 대한 일반요청의 도착율(λ_2) 대한 일반요청과 우선요청의 차단율을 $P_h=0.1$과 0.3에 대하여 각각 구한 결과를 도시하였다. 우선요청용 서버가 없는 방식II, III, IV에서는

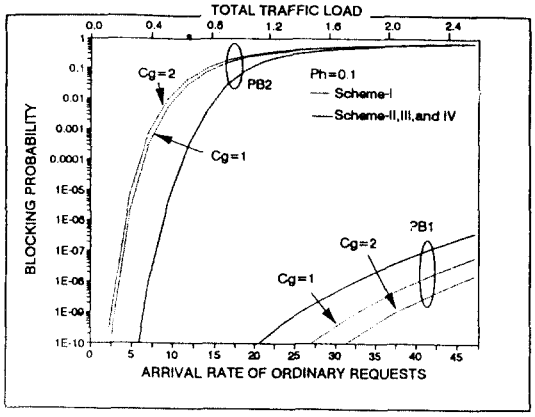

그림 5. Ph = 0.1일 때 4가지 요청처리방식의 blocking확률 비교.

Fig 5. Comparison of blocking probability of four request handing schemes for Ph = 0.1.

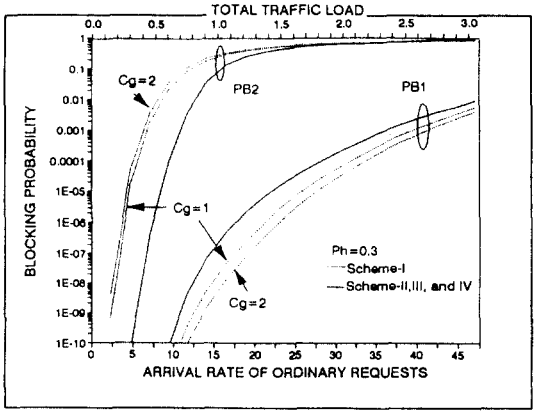

그림 6. Ph = 0.3일 때 4가지 요청처리방식의 blocking확률 비교.

Fig 6. Comparison of blocking probability of four request handing schemes for Ph = 0.3.

우선요청용 서버가 있는 방식I에 비하여 일반요청의 차단확률이 상당히 많이 감소된다. 대신에 우선요청에 대한 결과는 이와 반대이다. 이러한 장단점은 우선 요청용 서버의 갯수(Cg)가 증가되면 더 분명해 진다. 일반요청과 우선요청의 차단율에 대한 중요도는 서비스제공자의 정책에 좌우되는 문제이므로 한마디로 어떤방식이 좋다라고 단언하기는 어렵지만 일반

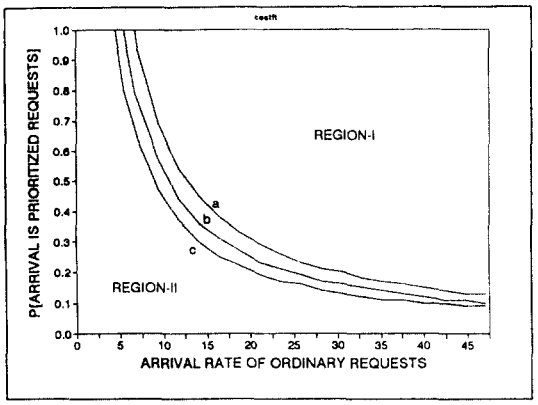

그림 7. 우선요청용 서버가 1개 있는 방식 I과 우선요청용 서버가 없는 다른 방식들간의 선택을 위한 경계곡선. (a) $1-\alpha=10^{-5}$ (b) $1-\alpha=10^{-6}$ (c) $1-\alpha=10^{-7}$

Fig 7. Boundaries between Scheme-I with a single reserved and a scheme without reserved servers.

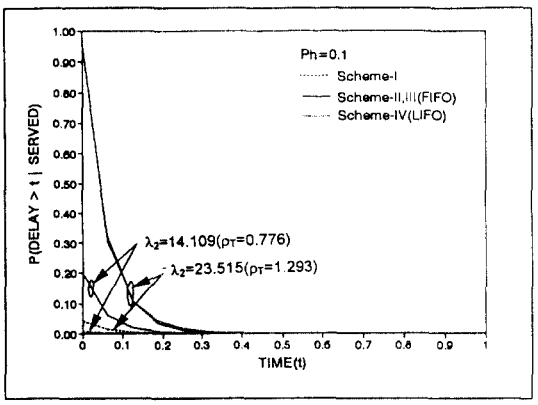

그림 8. Ph=0.1일 때, 4가지 방식에 대한 우선요청의 대기시간분포 비교.

Fig 8. Comparison of waiting time distribution of PRs for Ph=0.1 under Schemes I, II, III, IV.

적으로 공중망에서의 불통율(차단율)이 1%가 초과되지않도록 설계한다고 볼때 그림 5와 6에서 방식 II, III, IV에서의 일반요청의 일반적인 차단율이 1% 이하로 유지되는 부분에서 우선요청이 차단이 전혀 없거나 아주 낮은 값을 나타나므로 우수한 방식으로 평가될 수 있다.

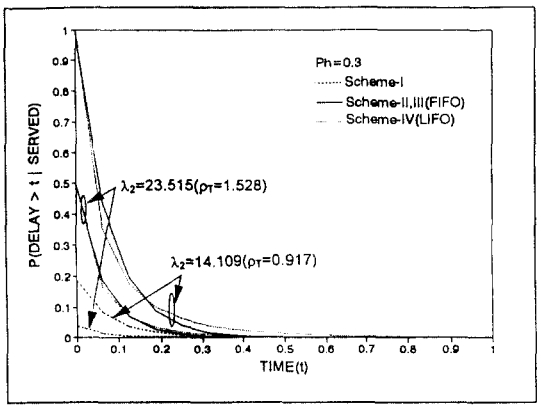

그림 9. Ph=0.3일 때, 4가지 방식에 대한 우선요청의 대기시간분포 비교.

Fig 9. Comparison of waiting time distribution of PRs for Ph=0.3 under Schemes I, II, III, IV.

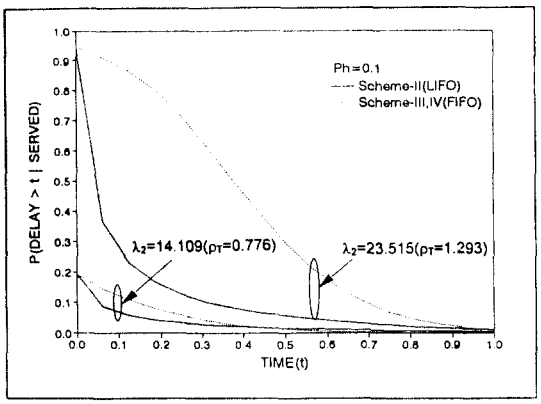

그림 10. Ph=0.1일 때, 방식II, III, IV하에서 일반요청의 대기시간분포 비교.

Fig 10. Comparison of waiting time distribution of OCs for Ph=0.1 under Schemes II, III, IV.

일반요청의 도착율이나 우선순위요청의 비율에 따른 가장 우수한 방식에 대한 선택은 다음과 같은 비용함수(Cost Function)를 도입하면 좀더 객관화 될 수 있을 것으로 판단된다. 요청의 대기시간을 고려하지 않는 경우, 요청처리방식을 비교하는데 필요한 비용함수, $CF=\alpha P_{B1}+(1-\alpha)P_{B2}$, $0\le\alpha\le1$를 사용할 수 있다. 여기서 α는 우선순위가 있는 요청과 일반요청의 차단율의 상대적인 가중치를 의미한다. 즉, 주

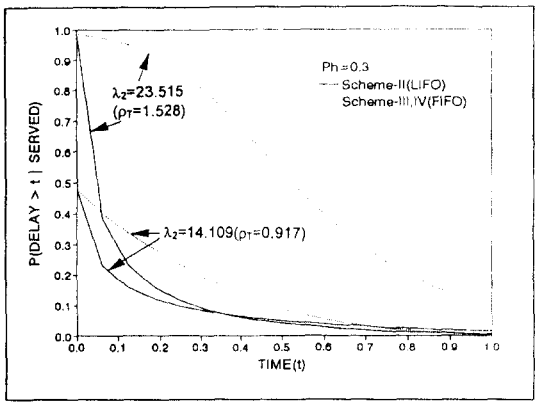

그림 11. Ph=0.3일 때, 방식II, III, IV하에서 일반요청의 대기시간분포 비교.

Fig 11. Comparison of waiting time distribution of OCs for Ph=0.3 under Schemes II, III, IV.

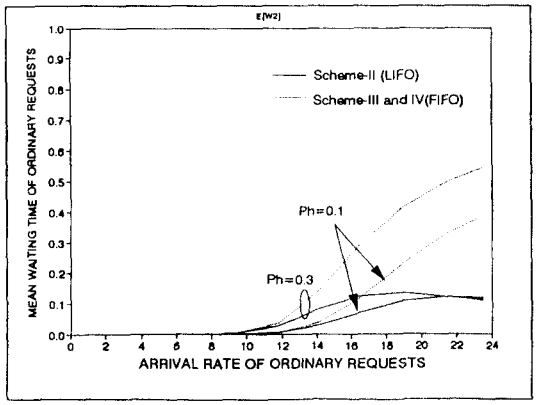

그림 12. 일반요청의 평균지연시간의 비교.

Fig 12. Comparison of mean waiting time of ordinary requests.

어진 각 차단율에 대하여 계산한 방식I의 CF값이 나머지 방식의 CF값보다 작은 경우에는 방식 I이 유리하다고 판정한다. 그림 7에는 1개의 우선요청용 서버가 있는 방식 I과 나머지 방식들에 대한 비용함수를 비교하여 구한 경계곡선을 도시한 것이다. 주어진 P_h, λ_2 및 α에 대하여, 영역-I은 방식 I이 선호되는 영역이고, 영역-II는 나머지 방식이 선호되는 영역이다. 그러므로, IP와 같이 대부분의 정상동작 영역에서 P_h와 λ_2 값이 작은 경우에는 영역-II 부근이므로 우선요청용 서버가 있는 방식 I 보다 방식 II, III, IV가 유리함을 알 수 있다.

다음에는 우선요청과 일반요청의 대기확률함수와 평균대기시간에 대한 결과를 도시한다. 그림 8과 9는 트래픽 부하가 1보다 낮은 경우($\lambda_2=14.109$)와 1보다 높은 부하($\lambda_2=23.515$)에서의 4가지 요청처리방식에 대한 우선요청의 대기시간확률함수를 $P_h=0.1$과 $P_h=0.3$에 대하여 각각 도시한 것이다. 이것과 대응되는 평균대기시간은 표 2에 나타나 있다. 우선요청용 서버를 별도로 갖고 있는 방식 I을 제외하고 FIFO방식과 LIFO방식을 비교해 보면, LIFO방식이 대기시간 분포면에서 긴 tail을 가지고 있는 반면, 평균대기시간 면에서는 거의 유사함을 알 수 있다.

그림 10과 11은 방식 II, III, IV에 대한 일반요청의 대기시간확률함수를 $P_h=0.1$과 $P_h=0.3$에 대하여 각각 도시한 것이다. 이것과 대응되는 평균대기시간은 그림 12와 표 2에 있다. FIFO 방식과 LIFO방식을 비교해 보면, 역시 일반요청의 경우에도 LIFO방식이 대기시간 분포면에서 긴 tail을 가지고 있지만, 평균대기시간 면에서도 유리함을 알 수 있다. 그림 12에서 LIFO방식인 경우 일반요청의 도착율이 높아질 때 평균대기시간이 감소하는 경향을 나타내는 것은 도착율이 증가함에 따라 신속히 서비스 받지 못한 요청들은 차단될 확율이 높아져 일단 성공적으로 처리되는 경우는 짧은 대기시간을 갖게 되기 때문이다. 이러한 현상은 Doshi가 분석한 우선순위가 없는 방식에 대한 결과에서도 언급되었다.[11]

대기시간에 대한 결과를 종합하면 우선요청인 경우 LIFO 방식에서의 평균지연시간은 FIFO 방식보다 큰 이득은 없고 대신에 편차가 크므로, 공평성이 높은 FIFO방식이 좋다고 할 수 있다. 그러나, 일반요청의 경우에는 LIFO방식의 경우 FIFO에 비하여 대기시간의 편차가 크나, 그림 10, 11에서 보는 바와 같이 tagged request가 아주 오래 대기할 확율은 매주 낮음을 알 수 있다. 대신에, 평균지연시간에서의 이득은 LIFO의 경우가 우수함을 알 수 있다.

IP는 차세대 지능망에서 교환기등이 제공할 수 없는 특수한 자원을 제공하는 특수 시스템이기 때문에 그 중요성에 비추어 충분한 수의 자원을 보유하도록 설계될 것이 분명하므로 과부하 상태가 아닌 정상적인 경우에는 요청호의 도착율과 우선순위호의 비율이 모두 낮은 상태를 유지할 것이다. 이러한 상황에서는 위에서 나타난 차단율과 지연시간에 대한 결과에서 알 수 있는 바와 같이 우선요청용 서버를 별도로 두지않고 우선요청은 FIFO, 일반요청은 LIFO방

표 2. 4가지의 요청처리 방식에 대한 요청의 평균대기시간의 비교
Table 2. Comparison of Average Waiting Times for Various Request Handing Schemes

Ph	λ2	ρ_T	E[W1]				E[W2]	
			Scheme-I		Scheme-II Scheme-III (FIFO)	Scheme-IV (LIFO)	Scheme-II (LIFO)	Scheme-III Scheme-IV (FIFO)
			Cg=1	Cg=2				
0.1	14.109	0.78	3.05E-4	2.99E-5	1.17E-2	1.17E-2	2.75E-2	3.75E-2
	23.515	1.29	2.39E-3	3.21E-4	6.03E-2	5.97E-2	1.16E-1	3.88E-1
0.3	14.109	0.92	2.38E-3	6.36E-4	3.45E-2	3.40E-2	8.13E-2	1.40E-1
	23.515	1.53	1.43E-2	5.56E-3	8.25E-2	7.68E-2	1.09E-1	5.46E-1

식으로 처리하는 방식 II가 가장 유리한 방식임을 알 수 있다.

Ⅵ. 결 론

본 논문에서는 Bellcore의 AIN개념과 ITU-T의 지능망 개념에 따른 IP의 개념적 구조와 서비스요청 처리방식들을 제안하고 각 처리방식에 대한 성능평가를 실시하여 IP 시스템에 가장 적합한 처리방식을 제시하였다. 이를 위해 IP의 서비스제어 모듈과 자원관리 및 서비스기능 모듈에 대한 하나의 큐잉모델을 선택하여 이 모델에 적합한 요청처리방식 4가지를 제안하고, 큐잉이론을 사용하여 성능을 비교, 분석하였다. 여기서 제안된 방식은 각각 별도의 우선요청용 서버를 고정적으로 할당하는 방식 1가지와 별도의 우선요청용 서버 없이 우선요청과 일반요청을 각각 FIFO /LIFO, FIFO/FIFO, LIFO/FIFO방식으로 처리하는 3가지로서, 이 중에서 우선요청용 서버를 별도로 두지않고 우선요청과 일반요청을 각각 FIFO, LIFO 방식으로 처리하는 것이 요청호의 도착율과 우선순위호의 비율이 모두 낮은 정상적인 동작범위에서는 가장 유리한 방식임을 알 수 있었다. 그러므로, 본 연구의 결과들을 IP시스템 구현시 활용하면 보다 고성능의 시스템을 구현하는데 도움이 될 수 있을 것이다.

참 고 문 헌

1. Andrew Batten, et. al, "ISDN and IN," Proceedings of ICIN '92, pp. 23-27, Mar. 1992.
2. ITU-T Rec. Q1218(draft), "Intelligent Network Interface Recommedations," Mar. 1992.
3. B. Brunner, et. al, "Implementing B-ISDN IPs and SCPs for the Intelligent Network," ICIN '92, Mar. 1992.
4. 차세대 지능망 개념서, 한국전자통신연구소, 1991.
5. G. B. Choi, et. al, "A Service Switching Point for Intelligent Network," Proceedings ICCC Conference on Intelligent Networks, pp. 268-274, May 1992.
6. Robert W. Keltgen, "Intelligent Peripherals: Interfacing Subscribers to the Advanced Intelligent Network," ICIN '92 Mar. 1992.
7. 최고봉 외, "고도 지능망을 위한 지능형 정보제공 시스템," 전자공학회지, vol. 20, no. 2, pp. 115-126, 1993년 2월.
8. 양선회, "음성우편시스템 연구개발-Software구현," KAIST M.S. Thesis, Jan. 1986.
9. 임병근, "음성우편시스템 연구개발-Hardware구현," KAIST M.S. Thesis, Jan. 1986.
10. D. H. Hong and S. S. Rappaport, "Traffic model and performance analysis for cellular mobile radio telephone systems with priortized handoff procedures," *IEEE Trans. Vech. Tech.*, vol. VT-28, no. 3, pp. 77-92, Aug. 1986.
11. B. T. Doshi and H. Heffes, "Overload performance of several processor queueing disciplines for the M/M/1 queue," *IEEE Trans. on Commun*, vol. COM-34, pp. 538-546, June 1986.
12. D. P. Heyman, "The pushout-priority queue discipline," *Oper. Res.*, vol. 33, no. 2, pp. 397-403, 1985.

13. W. Feller, An Introduction to Probability Theory and Its Applications, vol. 1, New York : Wiley, pp. 349-370, 1968.

崔 高 峰(Go Bong Choi) 정회원
1957年 8月 23日生
1980年 2月 : 경북대학교 공과대 전자공학과 졸업
1982年 2月 : 경북대학교 대학원 전자공학과 졸업(석사)
1991年 ~ 현재 : 성균관대학교 전자공학과(박사과정)
1982年 : 국방과학연구소 연구원
1987年 ~ 1989年 : Bell Telephone/Alcatel(벨지움) 방문연구원
1983年 ~ 현재 : 한국전자통신연구소 개발환경연구실장
※주관심분야 : 전자교환기, 지능망교환기, 지능형 정보제공시스템, 성능분석

權 奇 浩(Key Ho Kwon) 정회원
1953年 2月 5日生
1975年 : 성균관대학교 공과대 전자공학과 졸업
1978年 : 서울대학교 대학원 전자공학과 졸업(석사)
1988年 : 서울대학교 대학원 전자공학과 졸업(박사)
1978年 ~ 1980年 : 한국전기통신연구소 연구원
1983年 ~ 1985年 : 수원대학교 전자계산학과 전임강사
1985年 ~ 1989年 : 명지대학교 전자계산학과 조교수
1989년 ~ 현재 : 성균관대학교 전자공학과 부교수
※주관심분야 : 인공지능, 자동제어, 로보트 공학

尹 鍾 浩(Chong Ho Yoon) 정회원
1957년 12월 18일생
1977년 3월 ~ 1984년 2월 : 한양대학교 공과대학 전자공학과(공학사)
1984년 3월 ~ 1986년 2월 : 한국과학기술원 전기 및 전자공학과(공학석사)
1986년 3월 ~ 1990년 8월 : 한국과학기술원 전기 및 전자공학과(공학박사)
1991년 9월 ~ 현재 : 한국항공대학교 항공통신정보공학과 조교수
※주관심분야 : 고속통신망, 연동장치개발, 멀티미디어, 성능분석