

이질적 버스트 입력 트래픽 환경에서 패킷 교환기의 연속 시간 큐잉 모델과 근사 계산 알고리즘

(Continuous-Time Queuing Model and Approximation Algorithm of a Packet Switch under Heterogeneous Bursty Traffic)

홍 석 원 †

(Sugwon Hong)

요 약 본 논문에서는 버퍼를 공유하는 패킷 교환기의 연속 시간 큐잉 모델을 제시하고 큐 길이 확률 분포를 구하는 근사 계산 알고리즘을 제안한다. N 개의 입력 프로세스는 상호 이질적인 버스트 특성을 갖는다. 입력 프로세스는 계차-2 콕시안 분포로서 모형화하며 서버의 서비스 시간은 계차- r 에랑 분포로서 모형화한다. 근사 알고리즘은 통합된 상태 변수를 사용하여 큐잉 시스템을 표현한다. 먼저 N 개의 입력 프로세스는 하나의 통합된 상태 변수로 나타내며 큐잉 시스템은 서브 시스템으로 분해하고 이것을 통합된 상태 변수로 나타낸다. 그리고 이러한 통합된 상태 변수를 사용하여 반복적인 방법에 의해서 상태 방정식의 해를 유도한다. 근사 알고리즘의 타당성은 시뮬레이션을 통해서 검증한다.

키워드: 연속 시간 큐잉 모델, 공유 버퍼 패킷 스위치, 근사 알고리즘, 버스트 트래픽, 콕시안 분포, 중단 포아송 프로세스, 에랑 분포, 통합

Abstract This paper proposes a continuous-time queuing model of a shared-buffer packet switch and an approximate algorithm. N arrival processes have heterogeneous bursty traffic characteristics. The arrival processes are modeled by Coxian distribution with order 2 that is equivalent to Interrupted Poisson Process. The service time is modeled by Erlang distribution with r stages. First the approximate algorithm performs the aggregation of N arrival processes as a single state variable. Next the algorithm decompose the queuing system into N subsystems which are represented by aggregated state variables. And the balance equations based on these aggregated state variables are solved for by iterative method. Finally the algorithm is validated by comparing the results with those of simulation.

Key words: continuous-time queuing model, shared-buffer packet switch, approximate algorithm, bursty traffic, Coxian distribution, Interrupted Poisson Process, Erlang distribution, aggregation

1. 서론

패킷망에서 교환기는 망 구성의 핵심 요소라고 할 수 있다. 이러한 이유로 ATM 교환기를 비롯한 패킷 교환

기의 성능을 평가하기 위해서 큐잉 모델 이론을 적용한 많은 연구가 이루어져 왔다. 이것은 단순히 교환기의 개발을 위한 성능 요소의 평가라는 목적도 있지만 패킷망에서의 체증(congestion) 제어를 위해서 적정 수준의 성능을 보장하기 위한 교환기의 버퍼 크기의 결정이라는 망 차원에서의 요구 사항도 배경이 되고 있다고 할 수 있다[1]. 이러한 요구 사항은 최근 인터넷의 경우도 라우터에서의 체증 제어의 필요성이 등장하면서 다양한

· 본 연구는 과학기술부 및 한국과학재단의 ERC 프로그램을 통한 지원으로 이루어졌음

† 종신회원 : 명지대학교 컴퓨터소프트웨어학과 교수
swhong@mju.ac.kr

논문접수 : 2002년 7월 10일
심사완료 : 2003년 2월 27일

트래픽 환경에서의 버퍼 길이의 분포, 블로킹에 의한 패킷의 손실을, 지연 시간 등의 다양한 평가 요소가 관심의 대상으로 등장하게 되었다[2].

이러한 패킷 교환기에 대해서 큐잉 모델을 적용하여 해를 구하는 과제에 어려움은 마코프 체인(Markov Chain)의 상태의 수가 너무 많기 때문에 일반적인 수학적 방법으로 풀기에는 너무 복잡한 문제라는 점이다. 이러한 이유로 이 분야의 많은 연구가 모든 입력 프로세스를 동일한 포아송 분포(Poisson Distribution)나 베르누리 분포(bernoulli Distribution)를 갖는 단순한 입력 프로세스를 가정하였다. 하지만 이러한 모형은 실제 트래픽 환경을 지나치게 단순화하는 문제점이 있다. 최근 트래픽의 버스티(bursty) 특성을 고려하여 중단 포아송 과정(Interrupted Bernoulli Process)이나 중단 베르누리 과정(Interrupted Bernoulli Process) 혹은 마코프 변조 포아송 과정(Markov Modulated Poisson Process) 등을 사용하고 있다[3]. 하지만 이 경우 마코프 체인의 상태 변수를 줄이기 위해서 N개의 입력 프로세스(Input process)들은 모두 동일한 확률 분포를 갖는 것으로 가정하고 있다[4].

본 논문에서는 이질적 특성을 갖는 N 개의 입력 프로세스 환경에서 버퍼를 공유하는 패킷 교환기의 연속 시간 큐잉 모형(continuous-time queuing model)을 제시하고 교환기의 버퍼 즉 큐(queue) 길이 확률 분포를 구하는 근사 계산 알고리즘을 제시한다. 본 논문에서 제시하는 알고리즘은 전체 시스템의 상태 수를 줄이기 위해서 서브 시스템(subsystem)으로 분해하고 전체 상태를 통합(aggregation)하는 접근 방법을 사용하고 있다. 이러한 통합 방법은 다른 응용에서도 적용되어 왔다[5,6,7]. 그리고 이산 시간 큐잉 모형(discrete-time queuing model)을 사용하여 동일한 통합 방법을 적용한 경우가 있다[8,9].

본 논문에서는 연속 시간 모형으로서 2차-콕시안 분포(Coxian distribution with order 2)를 사용하여 N개의 입력 프로세스를 나타내고 이를 통합된 상태(aggregated state)로 표현하는 방법을 제시한다. 그리고 서브 시스템으로 분해하여 이를 통합된 상태 변수로 나타내는 방법을 제시하고 이러한 통합된 상태 변수들을 통해서 만들어지는 평형 방정식에서 근사적으로 해를 구하는 방법을 제시한다. 그리고 이러한 근사 계산 알고리즘의 결과를 시뮬레이션을 통해 구한 결과와 비교하여 그 타당성을 검증해 본다.

본 논문의 구성은 다음과 같다. 2절에서는 패킷 교환

기의 연속 시간 큐잉 모델을 제시하고, 3절에서는 이것을 근사적으로 해를 구하는 알고리즘을 설명한다. 4절에서는 알고리즘을 통해 구한 결과와 시뮬레이션을 통해 구한 결과를 비교하여 타당성을 검증하고 5절에서 결론을 맺는다.

2. 연속 시간 큐잉 모델

본 논문에서는 버퍼를 공유하는 패킷 교환기의 연속 시간 큐잉 모델은 그림 1과 같이 모형화 하였다. 이 모델은 N개의 단일 서버 큐(queue)로서 구성된다. 여기서 N은 교환기의 포트 수에 해당하며 각 서버는 교환기의 출력 포트에 해당한다. 포트 i로 입력된 패킷은 확률 ρ_{ij} 로 출력 포트 j로 스위칭된다. 모든 서버의 큐에 있는 패킷의 수는 스위치의 총 버퍼 크기인 M을 넘지 않는다. 스위치의 버퍼가 다 차있는 순간에 도착한 패킷은 손실된다.

그림 1의 모델에서 서버에서의 서비스 시간은 계차-r 열랑 분포(Erlang distribution with r stages: E_r)로 모형화하였다. 여기서 μ 는 각 지수 단계(exponential stage)에서의 서비스율을 나타낸다. 패킷을 처리하는 서비스 시간 분포는 패킷 길이의 분포에 의해 결정된다. E_r 분포의 분산은 $1/k\mu^2$ 로서 지수 분포의 경우보다 적은 분산값을 갖는다. 따라서 패킷의 길이가 지수 분포를 갖는다는 일반적인 가정에 비해서 특정 길이에 집중되어 있는 적은 분산값의 패킷 길이를 가정하고 있다. 각 입력 큐에서의 도착 과정(arrival process)은 버스티 트래픽(bursty traffic)의 특성을 나타내기 위해서 중단 포아송 프로세스(Interrupted Poisson Process: IPP)로 모형화하였다[3].

IPP에서는 두 개의 별도의 지수 분포 구간, 즉 패킷이 도착하는 활동 구간(active period)과 패킷의 도착이 없는 비활동 구간(idle period)의 두 구간이 반복해서 발생한다. 활동 구간에서 패킷의 도착은 포아송 분포의 형태로 발생한다. IPP에서 두 패킷의 연속적인 도착 시간은 계차-2 초지수 분포(hyperexponential distribution with order 2: H_2)로 변형할 수 있으며, 이것은 다시 계차-2 콕시안 분포(Coxian distribution with order 2: C_2)로 변형할 수 있다[10]. 그림 2는 IPP, H_2 , 그리고 C_2 간의 관계를 보여 주고 있다. 그림 2의 IPP에서 $1/a$, $1/\beta$ 그리고 λ 는 각각 비활동 구간의 평균 시간, 활동 구간의 평균 시간, 그리고 활동 구간에서의 패킷의 도착율을 나타내고 있다. IPP는 λ_1 , λ_2 , 그리고 단계(stage) 1에서 단계 2로 이행할 확률 a로 이루어진 콕시안 분포로 변형할 수 있다. 따라서 스위치의 i번째 입력 포트의

입력 프로세스는 $\lambda_{i,1}$, $\lambda_{i,2}$, 그리고 $a_i(i=1,2, \dots, n)$ 로 이루어진 C^2 분포로 모형화할 수 있다.

본 논문에서는 입력 트래픽의 버스티 특성을 나타내는 파라미터로서 변동 계수의 제곱(Squared Coefficient of Variation) C^2 으로 나타내었다. IPP에서 평균 도착 간격 시간(inter-arrival time), 분산, 그리고 C^2 는 다음과 같이 구할 수 있다[10].

$$E(t) = (\alpha + \beta) / \alpha \lambda \tag{1}$$

$$\text{var}(t) = (2\lambda\beta + (\alpha + \beta)^2) / (\alpha\lambda)^2 \tag{2}$$

$$C^2 = 1 + 2\lambda\beta / (\alpha + \beta)^2 \tag{3}$$

H_2 분포의 파라미터 θ_1 , θ_2 , P_1 , P_2 와 IPP의 α , β , λ 의 관계는 식 (4), (5)와 (6)과 같이 나타낼 수 있다.

$$\theta_1 = \frac{1}{2}((\lambda + \alpha + \beta) + \sqrt{(\lambda + \alpha + \beta)^2 - 4\alpha\lambda}) \tag{4}$$

$$\theta_2 = \frac{1}{2}((\lambda + \alpha + \beta) - \sqrt{(\lambda + \alpha + \beta)^2 - 4\alpha\lambda}) \tag{5}$$

$$P_1 = (\lambda - \theta_2) / (\theta_1 - \theta_2), \text{ and } P_2 = 1 - P_1 \tag{6}$$

또한 C_2 의 λ_1 , λ_2 , 그리고 a 는 식 (7)로 구할 수 있다. $\lambda_1 = \theta_1$, $\lambda_2 = \theta_2$, and $a = p_2(\lambda_1 - \lambda_2) / \lambda_1$ (7)

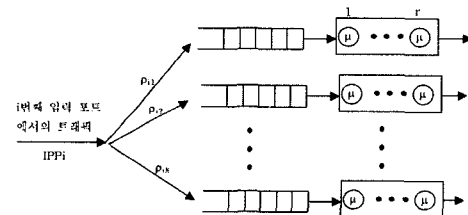
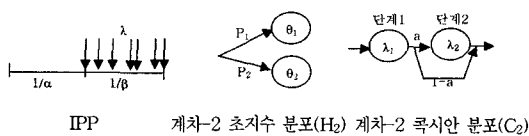


그림 1 공유 버퍼 패킷 교환기의 연속 시간 큐잉 모형



IPP 계차-2 초지수 분포(H_2) 계차-2 복사안 분포(C_2)

그림 2 세 가지 분포의 상관성

3. 근사 계산 알고리즘

3.1 전체 평형 방정식(global balance equation)

그림 1과 같은 큐잉 모델의 완전한 상태(state)는 벡터 $(\underline{w}, \underline{m}) = (w_1, \dots, w_N; m_1, \dots, m_N)$ 으로 주어진다. 여기서 $w_i(i=1,2, \dots, N, w_i=1$ 혹은 $2)$ 는 i 번째 입력 포트에 도착하는 패킷의 입력 프로세스가 그림 2에서의 같은 C_2 분포의 어느 단계에 현재 위치하고 있는지를 나타내며, $m_j(j=1,2, \dots, N)$ 는 j 번째 큐에 있는 패킷들이 서비스를 종료할 때까지 아직 남아있는 E_r 분포의 총 단계 수(total

number of stages)를 나타낸다[11].

K 는 현재 스위치에 존재하는 모든 패킷의 수를 나타내고, k_i 는 i 번째 큐에서의 패킷의 수라고 하면, K 와 k_i 는 다음과 같은 관계식을 갖는다.

$$K = \sum_{i=1}^N k_i = \sum_{i=1}^N \left\lceil \frac{m_i}{r} \right\rceil$$

여기서 $\lceil a \rceil$ 는 a 보다 크거나 같은 정수 중에서 가장 작은 정수를 의미한다.

이 모델의 전체 평형 방정식(global balance equation)을 구하는데 표현을 단순화하기 위해서 다음과 같은 유사 도착율과 유사 서비스율을 정의한다.

$$\lambda_{i,1}(K) = \begin{cases} \lambda_{i,1} & 0 \leq K \leq M-1 \\ 0, & K = M \end{cases}$$

$$\lambda_{i,2}(K) = \begin{cases} \lambda_{i,2} & K = M \\ 0, & 0 \leq K \leq M-1 \end{cases}$$

$$\mu(m_j) = \begin{cases} \mu, & 1 \leq m_j \leq rM \\ 0, & m_j = 0 \end{cases}$$

상태 $(\underline{w}, \underline{m})$ 에서의 안정 상태 확률(steady state probability)을 $P(\underline{w}, \underline{m})$ 으로 표시한다. 그리고 주어진 \underline{w} 에 대해서 현재의 입력 프로세스가 그림 2에서와 같은 C_2 분포의 단계 1에 존재하는 입력포트의 집합을 W_1 그리고 C_2 분포의 단계2에 존재하는 입력포트의 집합을 W_2 으로 정의한다. 그러면 $(\underline{w}, \underline{m})$ 에서의 전체 평형 방정식은 상태 $(\underline{w}, \underline{m})$ 로부터 이웃 상태로의 전이와 상태 $(\underline{w}, \underline{m})$ 의 이웃 상태에서 상태 $(\underline{w}, \underline{m})$ 로의 전이에 의해 다음과 같이 구할 수 있다.

$$\begin{aligned} & \left[\sum_{i \in W_1} (a\lambda_{i,1} + (1-a_i)\lambda_{i,1}(K)) + \sum_{i \in W_2} \lambda_{i,2} + \sum_{j=1}^N \mu_j(m_j) \right] P(\underline{w}, \underline{m}) \\ &= \sum_{i \in W_2} a_i \lambda_{i,1} P(\underline{w} - \underline{e}_i^n, \underline{m}) \\ &+ \sum_{i \in W_1} \sum_{j=1}^N b_{i,j} (1-a_i) \lambda_{i,1} P(\underline{w}, \underline{m} - r \underline{e}_j^n) \\ &+ \sum_{i \in W_1} \sum_{j=1}^N b_{i,j} \lambda_{i,2} P(\underline{w} + \underline{e}_j^n, \underline{m} - r \underline{e}_i^n) \\ &+ \sum_{i \in W_1} \lambda_{i,2}(K) P(\underline{w} + \underline{e}_i^n, \underline{m}) \\ &+ \sum_{j=1}^N \mu_j P(\underline{w}, \underline{m} + r \underline{e}_j^n) \end{aligned} \tag{8}$$

이 식에서 \underline{e}_i^n 은 i 번째 요소가 1인 n 차원 단위 벡터이다. 식 (8)의 좌변은 입력 프로세스가 C_2 의 1단계에서 2단계로 이행하는 전이($a_i \lambda_{i,1}$), C_2 의 1단계에서 직접 출력포트의 큐로 스위칭 되는 전이($(1-a_i) \lambda_{i,1}(K)$), C_2 의 2단계에서 출력 포트의 큐로 스위칭 되는 전이($\lambda_{i,2}$), 그리고 E_r 의 서비스 단계에서의 전이($\mu_j(m_j)$)로 구성되어 있다. 식 (8)의 우변의 각 항은 상태 $(\underline{w}, \underline{m})$ 의 이웃 상태를 나타낸다. 첫번째 항은 입력 프로세스가 C_2 의 1단계에서 2단계로 이행하는 전이이고, 두번째 항은 C_2 의 1단계에서 직접 출력포트의 큐로 스위칭 되는 전이, 세번째

항은 C_2 의 2단계에서 출력포트의 큐로 스위칭 되는 전이, 네번째 항은 스위치의 큐가 가득 찼을 때($K=M$) 도착하는 패킷이 손실되는 경우의 전이를 나타낸다. 그리고 마지막 항은 서비스 단계에서의 전이를 나타낸다.

3.2 상태 변수의 통합(aggregation)

식 (8)과 같은 전체 평형 방정식의 상태의 수는 $2^N(M+1)$ 이다. 따라서 이 방정식의 해를 구하는 것은 $O(2^NM)$ 의 복잡도(complexity)를 갖는다. 따라서 이 문제를 수식적으로(Numerically) 접근 가능한 문제가 되도록 하기 위해서는 상태의 수를 감소시킬 필요가 있다.

본 논문에서는 각 큐를 나머지 다른 큐들과 분리하고 나머지 다른 큐들을 하나의 통합된 큐로서 구성된 서브 시스템(subsystem)을 대상으로 하는 방법을 사용한다. 현재 분석의 대상이 되고 있는 큐를 Q_j 라고 하면 j 번째 큐를 제외한 나머지 큐들을 $N-Q_j$ 로 표시한다. 그림 3은 이와 같이 구성된 서브 시스템 j 를 보여 주고 있다. 결국 그림 1의 큐잉 모델은 각각의 큐에 대해서 N 개의 서브 시스템으로 분해된다.

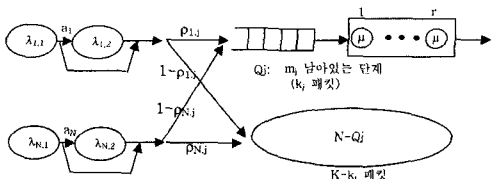


그림 3 서브 시스템의 큐잉 구조(N 입력 프로세스를 통합하기 전)

이 서브 시스템은 n 개의 입력 프로세스의 버스트 특성을 나타내기 위해서 C_2 로 표현하였기 때문에 각 서브 시스템에서 상태의 수는 대단히 많다. C_2 분포를 갖는 N 개의 입력 프로세스의 상태의 수를 줄이기 위해서 N 개의 입력 프로세스를 하나의 도착 프로세스로 통합한다. 통합된 입력 프로세스는 변수 x 를 갖는데 x 는 도착 프로세스들 중에서 C_2 분포의 단계 2에 존재하는 입력 프로세스의 수를 나타낸다. 이와 같이 하여 서브 시스템은 세 개의 상태 변수, 즉 단계 2에 존재하는 입력 프로세스의 수(x), j 번째 큐에 있는 패킷들이 아직 완료해야 하는 단계들의 수(m_j), 그리고 전체 스위치에서의 전체 패킷의 수(K)로 표현할 수 있다.

S_x 는 상태 $\underline{w}=(w_1, \dots, w_N)$ 중에서 정확하게 x 개의 프로세스가 단계2에 있는 상태들의 집합을 나타낸다. 서브 시스템 j 를 이와 같이 세 개의 변수로 나타내기 위해서는 3.1절에서 설명한 원래의 상태 $(\underline{w}, \underline{m})$ 을 상태 $(x, m_j,$

$K)$ 로 통합하여야 한다. $P_j(x, m_j, K)$ 를 상태 (x, m_j, K) 의 정상 결합 확률 분포(stationary joint probability distribution)라고 하면 $P_j(x, m_j, K)$ 는 $P(\underline{w}, \underline{m})$ 의 모든 상태 중에서 현재 출력 포트 큐 Q_j 의 패킷들이 아직 완료해야 하는 단계들의 수가 m_j 이면서 패킷의 총 수가 K 인 상태와 동시에 현재 x 개의 입력 프로세스가 C_2 의 2단계에 존재하는 상태들의 확률의 합으로서 다음과 같이 구할 수 있다.

$$P_j(x, m_j, K) = \sum_{\substack{\underline{w} \\ \sum_{i=1}^N m_i = K, m_j}} P(\underline{w}, \underline{m}). \tag{9}$$

이것을 식 (8)에 적용하면 통합된 상태에 대한 평형 방정식은 식 (10)과 같이 구할 수 있다.

$$\begin{aligned} & \left[\sum_{\underline{w} \in S_x} \text{Prob}\{\underline{w} | x, m_j, K\} \right. \\ & \left. \left\{ \sum_{i=1}^N (a_i \lambda_{i,1} + (1-a_i) \lambda_{i,1}(K)) + \sum_{i=2}^N \lambda_{i,2} \right\} \right. \\ & \left. + \mu(m_j) + \mu \sum_{i=1, i \neq j}^N \sum_{r=1}^{K - \sum_{i=1}^N m_i - 1} \text{Prob}\{m_i = rt + 1 | x, m_j, K\} \right] \\ P_j(x, m_j, K) &= \sum_{\underline{w} \in S_{x-1}} \text{Prob}\{\underline{w} | x-1, m_j, K\} \\ & \sum_{i=1}^N a_i \lambda_{i,1} P_j(x-1, m_j, K) + \sum_{\underline{w} \in S_x} \text{Prob}\{\underline{w} | x, m_j - r, K-1\} \\ & \sum_{i=1}^N P_{i,1} (1-a_i) \lambda_{i,1} P_j(x, m_j - r, K-1) \\ & + \sum_{\underline{w} \in S_x} \text{Prob}\{\underline{w} | x, m_j, K-1\} \\ & \sum_{i=1}^N \sum_{i=1, i \neq j}^N P_{i,1} (1-a_i) \lambda_{i,1} P_j(x, m_j, K-1) \\ & + \sum_{\underline{w} \in S_{x+1}} \text{Prob}\{\underline{w} | x+1, m_j - r, K-1\} \\ & \sum_{i=1}^N P_{i,1} \lambda_{i,2} P_j(x+1, m_j - r, K-1) \\ & + \sum_{\underline{w} \in S_{x+1}} \text{Prob}\{\underline{w} | x+1, m_j, K-1\} \\ & \sum_{i=1}^N \sum_{i=1, i \neq j}^N P_{i,1} \lambda_{i,2} P_j(x+1, m_j, K-1) \\ & + \sum_{\underline{w} \in S_{x+1}} \text{Prob}\{\underline{w} | x+1, m_j, K\} \sum_{i=2}^N \lambda_{i,2}(K) P_j(x+1, m_j, K) \\ & + \mu P_j(x, m_j + 1, K + \nu_{m_j+1}) \\ & + \mu \sum_{i=1, i \neq j}^N \sum_{r=1}^{K - \sum_{i=1}^N m_i - 1} \text{Prob}\{m_i = rt + 1 | x, m_j, K+1\} P_j(x, m_j, K+1), \end{aligned} \tag{10}$$

여기서 $x=0,1,\dots,N, m_j=0,1,\dots,rM, K=0,1,\dots,M$, 그리고 ν_{m_j+1} 은 다음과 같이 정의된다.

$$\nu_{m_j+1} = \begin{cases} 1, & \left\lfloor \frac{m_j}{r} \right\rfloor = \frac{m_j}{r} \\ 0, & \left\lfloor \frac{m_j}{r} \right\rfloor > \frac{m_j}{r} \end{cases}$$

식 (10)은 두 개의 미지의 조건 확률, 즉 $\text{Prob}\{\underline{w} | x, m_j, K\}$ 와 $\text{Prob}\{m_i = rt + 1 | x, m_j, K\}$ 을 포함하고 있다. 이 두 개의 조건 확률의 계산은 다음과 같다.

입력 도착 프로세스의 상태는 j 번째 큐에 존재하는

패킷들이 서비스를 완료하기 전에 남아 있는 C₂ 분포의 단계의 수 혹은 전체 교환기에 존재하는 패킷의 수에 의존하지 않는다고 가정한다. 그러면

$$\begin{aligned} \text{Prob}\{w | x, m_j, k\} &\approx \text{Prob}\{w | x\} \\ &= \frac{P(w)}{P(Sx)}, \end{aligned} \quad (11)$$

$$P(Sx) = \sum_{w \in Sx} P(w)$$

와 같이 된다. 그리고 식 (10)에 의하면

$$P(w) = \left[\prod_{i=1}^m \frac{\lambda_{i,2}}{a_i \lambda_{i,1} + \lambda_{i,2}} \right] \left[\prod_{i=1}^m \frac{\lambda_{i,1}}{a_i \lambda_{i,1} + \lambda_{i,2}} \right]$$

그리고 Prob{m₁=rt+1|x,m₁,K}를 계산하기 위해서 큐 l(l≠j)에서 현재 서비스 중인 패킷은 r개의 일량 단계(r Erlang stages) 중 현재 각 단계에 존재할 확률은 동일하다고 가정한다. 큐 l(l≠j)이 비어 있을 확률은 N-Q_j에 있는 패킷의 수에 의존한다. 따라서 Prob{m₁=rt+1|x,m₁,K}는 다음의 식과 같이 계산할 수 있다.

$$\begin{aligned} &\sum_{l=1, l \neq j}^N \binom{K - \sum_{i=1}^m m_i}{\sum_{i=1}^m m_i} - 1 \text{ Prob}\{m_1 = rt + 1 | x, m_j, K\} \quad (12) \\ &\cong \sum_{l=1, l \neq j}^N \frac{1 - \text{Prob}\{k_l = 0 | x, m_j, K\}}{r} \\ &\cong \sum_{l=1, l \neq j}^N \frac{1 - \text{Prob}\{k_l = 0 | K - k_j\}}{r} \\ &\cong f_j(K - k_j) / r \end{aligned}$$

식 (12)에서 f_j(K-k_j)는 서브 시스템 j의 N-Q_j에 K-k_j개의 패킷이 존재할 때 패킷이 하나라도 존재하는 즉, 비어있지 않은 큐의 수의 평균값을 의미한다.

3.3 f_j(K-k_j)의 근사 계산

식 (10)의 평형 방정식을 통해서 각 상태의 해를 구할 때 마지막으로 남은 과제는 f_j(K-k_j)를 구하는 것이다. 그런데 N 입력 프로세스를 통합된 상태로 표현할 경우 f_j(K-k_j)의 정확한 값은 계산할 수 없다. 따라서 이 값의 계산을 위하여 근사적인 방법을 찾아보아야 한다.

α=K-k_j라고 하고 α개의 패킷이 N-Q_j에 존재할 때 f_j(α)는 다음과 같은 속성을 갖는다. 먼저, 패킷이 한 개 존재할 때 비어있지 않은 큐의 수는 1, 즉 f_j(1)=1이 되며, 패킷이 무한개 존재한다면 비어있지 않은 큐의 수는 N-1, 즉 f_j(∞)=N-1이 된다. 이 두 값은 f_j(α)의 양 극값이 된다. 또한 패킷의 수가 증가할수록 비어있지 않은 큐의 수는 증가한다. 또한 N-Q_j에 패킷의 수가 증가할수록 이미 패킷으로 채워진 큐의 수가 증가하므로 새로운 패킷이 비어있는 큐로 갈 수 있는 확률은 감소한다. 이상의 속성을 갖는 f_j(α)는 그림 4와 같은 형태를 갖는다.

g_j(α)는 각 큐에서 서비스 시간이 무한대라고 가정하고 α개의 패킷이 N-Q_j에 도착한 후에 적어도 하나의

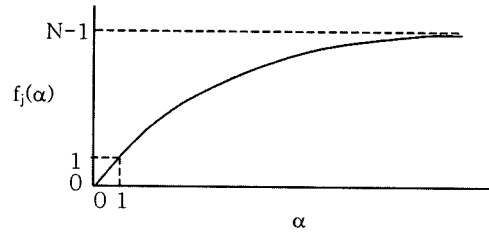


그림 4 f_j(α)의 그래프

패킷으로 채워지게 되는 큐의 수의 평균값이라고 정의한다. 그리고 각 입력포트에서 패킷의 도착율을 λ_i (i=1,2,...,N)라고 한다. 그러면 g_j(α)는 다음과 같은 식으로 나타낼 수 있다.

$$g_j(\alpha) = \sum_{i=1, i \neq j}^N \left[1 - \left(\frac{\sum_{l=1, l \neq j}^N \lambda_l}{\sum_{l=1, l \neq j}^N \lambda_l} \right)^\alpha \right] \quad (13)$$

g_j(α)는 서비스 시간이 무한대라고 가정했으므로 N-Q_j에서 비어있지 않은 큐의 수의 실제 평균값 f_j(α)보다는 작을 수 없다. 따라서 f_j(α)는 1과 g_j(α)를 양 극값(limits)으로 사용하여 다음과 같은 근사값으로 나타낼 수 있다.

$$f_j(\alpha) = 1 + \beta [g_j(\alpha) - 1] \quad 0 \leq \beta \leq 1 \quad (14)$$

g_j(α)는 f_j(α)와 마찬가지로 그림 4와 같은 그래프를 갖는다. 따라서 식 (13)과 (14)를 사용하여 β의 값을 조정하여 주어진 수렴 조건(convergence criterion)을 만족할 때까지 반복 수행함으로써 f_j(α)를 구할 수 있다.

수렴 조건으로는 여러 가지 방법을 사용할 수 있으나 본 논문에서는 계산의 편의를 위해서 교환기에서의 패킷의 평균수를 사용하였다. 교환기에서 패킷의 평균수는 다음과 같이 두 가지 식에 의해서 구할 수 있다.

$$E_1 = \sum_{j=1}^N \sum_{k_j=1}^{M_j} K_j P_j(k_j) \quad (15)$$

$$E_2 = \frac{1}{N} \sum_{j=1}^N \sum_{k=1}^{M_j} K P_j(K) \quad (16)$$

여기서 P_j(k_j)는 Q_j와 N-Q_j로 이루어진 서브 시스템에서 Q_j에 존재하는 패킷의 수가 k_j일 확률을 의미하며, P_j(K)는 동일한 서브 시스템에서 전체 시스템에 존재하는 패킷의 수가 K일 확률을 의미한다. 이 두 값은 다음과 같이 구할 수 있다.

$$P_j(k_j) = \sum_{x=0}^N \sum_{m_j=r_0}^{r_0} \sum_{r=r+1}^{M_j} P_j(x, m_j, K)$$

$$P_j(K) = \sum_{x=0}^N \sum_{m_j=0}^{M_j} P_j(x, m_j, K)$$

E₁과 E₂는 동일해야 하며 E₁-E₂는 f_j(α)와 β의 단순 증가 함수(monotonically increasing function)이다. 따라서 두 평균값의 크기에 따라 f_j(α)가 과대 계산되었

는지 혹은 과소 계산되었는지 판단할 수 있다.

3.4 알고리즘

지금까지 설명한 내용을 알고리즘으로 정리하면 다음과 같다.

Step0: 초기값 $\beta_L^0=0, \beta_H^0=1, S=0$

Step1: $S=S+1, \beta_M^S=(\beta_L^{S-1}+\beta_H^{S-1})/2$

식 (13)과 식 (14)를 사용하여 $f_j^S(a, \beta_M^S)$ 를 계산한다. ($a=1, \dots, M$ 그리고 $j=1, \dots, N$)

Step2: 식 (11)과 식 (12)를 사용하여 $\text{Prob}\{w|x, m_j, K\}$ 와 $\text{Prob}\{m_j=rt+1|x, m_j, K\}$ 을 계산한다.

식 (10)을 사용하여 $P_j^S(x, m_j, K)$ 를 구한다.

Step3: 식 (15)와 식 (16)을 사용하여 E_1^S 와 E_2^S 를 구한다.

Step4: 만약 $|E_1^S - E_2^S| \leq \epsilon$ 이면, 종료

만약 $E_1^S - E_2^S > \epsilon$ 이면, $\beta_H^S = \beta_M^S, \beta_L^S = \beta_L^{S-1}$ 그리고 Step 1로 간다.

만약 $E_1^S - E_2^S < -\epsilon$ 이면, $\beta_H^S = \beta_H^{S-1}, \beta_L^S = \beta_M^S$ 그리고 Step 1로 간다.

4. 검증

근사 계산 알고리즘을 $N=16$ 의 교환기에 대해 적용하여 구한 결과를 시뮬레이션을 사용하여 구한 결과와 비교하여 타당성을 검증하였다. 근사 계산 알고리즘과 시뮬레이션을 사용하여 전체 큐 길이 분포(global queue-length distribution) $P(K)$ ($K=0, 1, \dots, M$)와 첫번째 큐에서의 큐 길이 분포 $P_1(k_1)$ ($k_1=0, 1, \dots, M$)을 구하였다. 첫번째 큐는 가장 트래픽이 집중되는 큐가 되도록 하였다. 이와 함께 평균 큐 길이를 구하여 비교하였다.

각 입력 프로세스는 평균 도착률 0.5, 최대 도착률 (peak arrival rate) 1.0으로 하였으며 각각의 프로세스의 버스티 특성을 모두 다르게 하였다. 각 프로세스의 변동 계수의 제곱 C_2 값은 표 1에 주어져 있다.

입력 트래픽이 출력포트로 교환되는 확률(branching probabilities)은 $\rho_{i,1}=0.08, \rho_{i,2}=0.066, \rho_{i,3}=0.065, \rho_{i,4}=0.064, \rho_{i,5}=0.064, \rho_{i,6}=0.064, \rho_{i,7}=0.063, \rho_{i,8}=0.063, \rho_{i,9}=0.062, \rho_{i,10}=0.061, \rho_{i,11}=0.061, \rho_{i,12}=0.061, \rho_{i,13}=0.060, \rho_{i,14}=0.059, \rho_{i,15}=0.059, \rho_{i,16}=0.048$ ($i=1, \dots, N$)로 하였다. 이 값에서 보는 바와 같이 큐1에 가까워질수록 트래픽이 집중되도록 하였다. 서비스 시간은 E_3 로 가정하였다.

표 2, 3은 근사 계산 알고리즘과 시뮬레이션을 사용하여 서비스율이 1.3과 2.5일 때의 $P(K), P_1(k_1)$, 그리고 평균 큐 길이를 구한 결과이며 그림 5와 6은 이 결과를 그래프로 보여주고 있다. 표에서 시뮬레이션 결과는

95% 신뢰도 구간을 보여 주고 있다. 이 표에서 보는 바와 같이 근사 계산의 결과는 시뮬레이션의 결과와 매우 비슷한 것을 볼 수 있다. 각 큐에서의 확률 분포 $P_1(k_1)$ 의 결과가 전체 큐 길이 분포 $P(K)$ 의 결과보다 더 우수한 결과를 보여 준다. 이것은 근사 계산 알고리즘이 N 입력 프로세스를 통합하는 과정에서 잃게 되는 정보에서 비롯된다고 볼 수 있다. 결국 이 알고리즘의 정확도는 통합된 상태로 인하여 각 프로세스와 각 큐의 개별적인 정보가 얼마나 보존되느냐에 달려있다고 할 수 있다.

표 1 입력 프로세스의 파라미터 값($N=16$)

입력 프로세스	$\lambda_{1,1}$	$\lambda_{1,2}$	a_i	C^2
1	1.0025	0.0025	0.0025	200
2	1.0026	0.0026	0.0026	195
3	1.0027	0.0026	0.0026	190
4	1.0027	0.0027	0.0027	185
5	1.0028	0.0028	0.0028	180
6	1.0029	0.0029	0.0029	175
7	1.0030	0.0029	0.0030	170
8	1.0032	0.0031	0.0031	160
9	1.0034	0.0033	0.0034	150
10	1.0036	0.0036	0.0036	140
11	1.0039	0.0039	0.0039	130
12	1.0040	0.0040	0.0040	125
13	1.0042	0.0042	0.0042	120
14	1.0044	0.0044	0.0044	115
15	1.0046	0.0046	0.0046	110
16	1.0051	0.0050	0.0051	100

표 2 근사 계산 알고리즘과 시뮬레이션 결과(서비스율=1.3)

K	P(K)		k_1	$P_1(k_1)$	
	근사계산	시뮬레이션		근사계산	시뮬레이션
0	.0445	.0478±.0025	0	.2719	.2719±.0055
1	.0704	.0792±.0026	1	.2033	.1954±.0021
2	.0879	.0944±.0020	2	.1389	.1291±.0011
3	.0956	.0955±.0013	3	.0995	.0905±.0013
4	.0956	.0892±.0008	4	.0752	.0677±.0012
5	.0906	.0799±.0008	5	.0587	.0538±.0012
6	.0827	.0705±.0009	6	.0461	.0444±.0011
7	.0737	.0620±.0009	7	.0357	.0372±.0008
8	.0647	.0548±.0008	8	.0267	.0308±.0009
9	.0563	.0490±.0008	9	.0188	.0256±.0009
10	.0488	.0447±.0007	10	.0123	.0201±.0009
11	.0423	.0418±.0008	11	.0071	.0149±.0008
12	.0368	.0403±.0007	12	.0036	.0097±.0007
13	.0322	.0396±.0008	13	.0015	.0055±.0004
14	.0284	.0393±.0010	14	.0005	.0025±.0002
15	.0256	.0380±.0011	15	.0001	.0008±.0001
16	.0238	.0338±.0012	16	.0	.0001±.0
평균 큐길이	6.42		6.62		

표 3 근사 계산 알고리즘과 시뮬레이션 결과
(서비스 율=2.5)

K	P(K)		k ₁	P ₁ (k ₁)	
	근사계산	시뮬레이션		근사계산	시뮬레이션
0	.0496	.0442±.0026	0	.7404	.7455±.0033
1	.1167	.1139±.0043	1	.2036	.2025±.0019
2	.1618	.1667±.0035	2	.0441	.0418±.0011
3	.1702	.1802±.0017	3	.0093	.0081±.0003
4	.1503	.1603±.0014	4	.0020	.0017±.0001
5	.1178	.1238±.0022	5	.0005	.0003±.0
6	.0848	.0858±.0024	6	.0001	.0001±.0
7	.0573	.0547±.0021	7	.0	.0±.0
8	.0369	.0325±.0016	8	.0	.0±.0
9	.0229	.0183±.0011	9	.0	.0±.0
10	.0138	.0098±.0007	10	.0	.0±.0
11	.0081	.0050±.0004	11	.0	.0±.0
12	.0046	.0025±.0002	12	.0	.0±.0
13	.0026	.0012±.0001	13	.0	.0±.0
14	.0014	.0006±.0001	14	.0	.0±.0
15	.0008	.0003±.0	15	.0	.0±.0
16	.0004	.0001±.0	16	.0	.0±.0
평균 큐길이	3.90	3.78			

5. 결론

본 논문에서는 버퍼를 공유하는 패킷 교환기를 연속 시간 큐잉 모델로 모형화하고 큐 길이 분포를 계산하기 위한 근사 계산 알고리즘을 제시하였다. 이 근사 계산 알고리즘은 분석 대상의 큐와 나머지 큐들을 분리한 N개의 서브 시스템으로 분해하고 나머지 큐들을 통합된 상태 변수로 나타내었다. 또한 N개의 2차 코시안 분포를 갖는 입력 프로세스의 상태 수를 줄이기 위해서 N개의 입력 프로세스를 하나의 통합된 상태 변수로서 표현하였다. 이와 같이 하여 원래의 $O(2^NM)$ 의 상태 변수를 $O(NM^2)$ 로 감소하였다.

본 논문에서 제시한 근사 계산 알고리즘의 결과를 시뮬레이션의 결과와 비교하여 타당성을 검증하였다. 결과는 C^2 가 200에서 100의 값을 갖는 이질적인 입력 트래픽 상황과 특정 포트에 트래픽이 집중되는 극한 상황에서 비교적 양호한 결과를 보여 주었다. 이 알고리즘의 정확도는 통합된 상태에 의해서 손실된 각 개별 입력 프로세스와 개별 큐의 정보가 어느 정도 영향을 미치는가에 달려 있다고 할 수 있다.

참고 문헌

- [1] ATM Forum, Traffic Management Specification Ver. 4.0, April 1996.
- [2] S.. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transaction on Networking, 1(4), pp397-413, 1993.
- [3] 이호우, 대기행렬 이론-확률과정론 분석, 도서출판 기술, 1996.
- [4] H. Heffs and D.M. Lucantoni, "A Makrov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance," IEEE J. SAC 4, pp856-868, 1986.
- [5] A. Brandwajn and Y.-L.Jow, "An Approximation Method for Tandem Queues with Blocking," Oper. Res. 36, pp73-83, 1988.
- [6] P.J. Schweitzer and T. Altiok, "Aggregate Modelling of Tandem Queues without Intermediate Buffers," in: Perros and Altiok, eds., Queueing Networks with Blocking, pp47-72, North Holland, 1989.
- [7] Y. Takahashi, "A New Type Aggregation

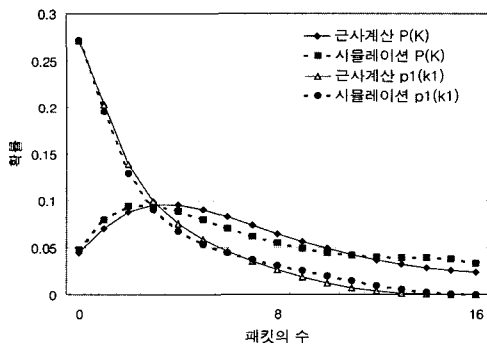


그림 5 근사 계산 알고리즘과 시뮬레이션 결과의 비교
(서비스 율=1.3)

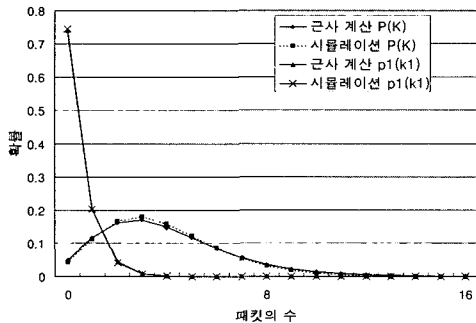


그림 6 근사 계산 알고리즘과 시뮬레이션 결과의 비교
(서비스 율=2.5)

- Method for Large Markov Chains and Its Application to Queueing Networks," in: Akiyama, ed., Teletraffic Issues in an Advanced Information Society, ITC-11, pp49~494, Elsevier, 1985.
- [8] S. Hong, "An Approximate algorithm for the analysis of the n heterogeneous IBP/D/1 queueing model," J. of KIMICS, 4(3), Sep. 2000.
- [9] H. Yanashita et al, "A discrete-time queueing model of the shared buffer ATM switch with bursy arrivals," Telecommunication Systems 2, pp1-20, 1993.
- [10] H. G. Perros and R. Onvural, "On the Superposition of Arrival Processes for Voice and Data," Fourth Int. Conf. on Data Communication Systems and Their Performance, Barcelona, June, pp341-357, 1990.
- [11] L. Kleinrock, Queueing Systems, Vol I: Theory, John Wiley & Sons, 1975.



홍 석 원

1979년 2월 서울대학교 물리학과(학사)
1988년 7월 North Carolina State University 전산학(석사). 1992년 7월 North Carolina State University 전산학(박사)
1993년 2월~1995년 2월 전자통신연구원 광대역 통신망 연구부 선임연구원. 1995년 3월~현재 명지대학교 컴퓨터 소프트웨어학과 부교수
관심분야는 망 구조 및 프로토콜, 성능 분석