Journal of the Korean Institute of Industrial Engineers
Vol. 31, No. 4, pp. 297-302, December 2005.

N-정책을 갖는 Push-Pull 대기행렬 시스템의 분석

김수연[1] · 김남기[2] · 김진동[3] · 이상민[4] · 채경철[4†]

[1]삼성화재 / [2]전남대학교 산업공학과 / [3]JP Morgan Chase Bank / [4]한국과학기술원 산업공학과

Analysis of the Push-Pull Queueing System under the N-Policy

Soo Yeon Kim[1] · Nam Ki Kim[2] · Jin Dong Kim[3] · Sang Min Lee[4] · Kyung Chul Chae[4]

[1]Samsung Fire & Marine Insurance, Seoul, 100-101
[2]Department of Industrial Engineering, Chonnam National University, Gwangju, 500-757
[3]JP Morgan Chase Bank, Seoul, 100-120
[4]Department of Industrial Engineering, KAIST, Daejeon, 305-701

Kopzon and Weiss adopted the notion of push-pull into the queueing system recently. We extend this queueing system to the (N_1,N_2)-policy version such that the original system corresponds to the special case $N_1=N_2=1$. For the extended system, we perform the cycle analysis and obtain the PGF of the stationary number of customers in the system.

***Keywords*:** M/G/1, N-policy, push-pull, queue length

1. 서 론

최근에 push-pull 개념을 대기행렬 모형에 도입한 논문이 발표되었는데, 이를 N-정책하의 대기행렬 모형으로 확장하고 분석하는 것이 본 논문의 목적이다. 지금까지 주로 생산관리 모형에 활용되던 push-pull 개념을(Bonney *et al.*, 1999) 처음으로 대기행렬 모형에 도입했다는 Kopzon and Weiss(2002)의 push-pull 대기행렬은 본 논문의 N-정책 중에서 “N=1”인 특수한 경우에 해당된다(이후 ‘Kopzon and Weiss’를 ‘K&W’라 칭함).

K&W가 제시한 push-pull 대기행렬을 보다 쉽게 이해할 수 있도록 다음과 같은 예를 든다. 두 명의 타이피스트(typist)가 각각 타이핑한 작업을 본인들이 교정하는 대신에 서로 상대방에게 넘긴다고 하자. 이는 교정 시에 본인의 오타는 놓치기 쉬운 반면에 상대방의 오타는 잘 찾을 수 있기 때문이다. K&W의 가정은 다음과 같다.

<가정 1> 두 타이피스트 T_1과 T_2가 타이핑할 일감은 무제한으로(즉, 쉼 없이 연속적으로) 공급된다.

<가정 2> 타이핑에 소요되는 시간들은 서로 독립인 지수분포를 따르고, T_1과 T_2의 평균 타이핑시간은 각각 λ_2^{-1}과 λ_1^{-1}이다.

<가정 3> 교정에 소요되는 시간들은 서로 독립인 일반분포를 따르고, T_1과 T_2의 평균 교정시간은 각각 μ_1^{-1}과 μ_2^{-1}이다. 그리고 교정시간은 타이핑시간과 독립이다.

<가정 4> $\rho_i = \lambda_i \mu_i^{-1} < 1, \quad i = 1, 2.$

<가정 5> 타이핑이든 교정이든 일단 시작한 작업은 도중에 중단하지 않는다(non-preemptive).

<가정 6> T_1과 T_2는 쉬지 않고 일(타이핑 또는 교정)을 계속 한다.

<가정 7> T_1과 T_2가 모두 교정을 하고 있는 상황은 금지한다. 즉, 최소한 한 명은 타이핑을 하고 있어야 한다.

위의 가정 중에서 <가정 6>까지는 다른 대기행렬 모형에서도 흔히 볼 수 있는 것이다. 특히, <가정 6>은 수율(throughput)을 최대화하기 위한 것임을 쉽게 알 수 있다. 반면에 <가정 7>

이 논문은 2004년도 학술진흥재단의 지원에 의하여 연구되었음(KRF-2004-015-C00084).
† 연락저자 : 채경철 교수, 305-701 대전시 유성구 구성동 373-1 한국과학기술원 산업공학과, Fax : 042-869-3110, E-mail : kcchae@kaist.ac.kr
2005년 2월 접수; 2005년 9월 수정본 접수; 2005년 10월 게재 확정.

은 다소 특이한데, 이 가정이 없으면 모형의 분석이(시뮬레이션으로는 가능하나 해석적으로는) 어려울 것이라는 것이 저자의 견해이다.

첫째로, <가정 6>과 <가정 7>을 동시에 만족시키기 위한 필요조건이 <가정 4>임을 쉽게 알 수 있다. 둘째로, <가정 7>을 만족시키기 위한 충분조건으로 K&W가 제시한 구체적인 방법은 다음과 같다. 시스템을 처음 가동시키는 시점에서 T_1과 T_2는 모두 타이핑을 시작하는데, 이때 교정할 일감은 하나도 없다([비고 1] 참조). T_2가 먼저 타이핑을 끝냈다고 하자. 즉, T_1이 교정할 일감이 먼저 발생했다고 하자. 그러면, T_1은 진행중이던 타이핑을 끝내는 즉시 교정을 시작한다(<가정 5> 참조). 그런데, 이때 유의할 점은 다음과 같다. T_1이 방금 타이핑을 끝낸 일감은 T_2에게 넘겨주지 않고 T_1이 임시로 보관한다. T_2는 교정할 일감을 넘겨받지 못했으므로 당분간 타이핑만 계속하고, 그동안 T_1은 교정만 계속한다. 그러다가 T_1이 교정할 일감이 다 떨어지는 순간 T_1은 다시 타이핑을 시작하는데, 이때 임시로 보관하고 있던 교정할 일감 하나를 T_2에게 넘긴다. 그러면, T_2는 진행중이던 타이핑을 끝내는 즉시 교정을 시작하는데, T_2 역시 방금 타이핑을 끝낸 일감을 T_1에게 넘겨주지 않고 임시로 보관한다. 이후 당분간 T_1은 타이핑만 계속하고 T_2는 교정만 계속하다가 T_2가 교정할 일감이 다 떨어지는 순간 T_2는 다시 타이핑을 하는데, 이때 임시로 보관하고 있던 교정할 일감을 T_1에게 넘기는 것이다.

K&W의 방법에서는 임시로 보관했다가 나중에 상대방에게 넘겨주는 일감의 수가 항상 1이다. 본 논문에서는 이를 확장하여 T_1과 T_2가 임시로 보관했다가 나중에 상대방에게 넘겨주는 일감의 수를 각각 N_2와 N_1이라 한다.

이러한 (N_1, N_2) -정책의 장점은 타이핑상태와 교정상태 간의 빈번한 교체를 막을 수 있다는 점인데, 특히 교체 시에 따르는 재가동(start-up) 비용이 발생하는 경우에 유리하다.

[비고 1] T_1과 T_2가 임시로 보관하고 있는 일감의 수가 모두 0인 경우는 초기상태에서만 발생한다(K&W, 2002), Lemma 2.1 참조).

2. 기존 연구결과의 요약

K&W의 방법에 따른 결과를 T_1의 관점으로 요약하되 편의상 1,2 단계로 나눈다. 1단계에서는 T_2가 교정하고 있는 시간구간들을 시간축에서 모두 제거하고 남은 구간들만 고려한다. 그리고, 2단계에서는 제거했던 시간구간들을 다시 시간축에 포함시킨다.

2.1 1단계 분석

시간축에서 T_2가 교정중인 구간들을 모두 제거하면 T_1은 준비기간(setup-time)을 갖는 M/G/1 대기행렬의 서버(server)가 되고 T_1이 교정할 일감은 고객이 되는데, 그 이유는 다음과 같다.

T_2가 교정중인 구간들을 제거하면 <가정 1,2,6,7>에 의해서 T_2는 연속적으로 타이핑만 하며 서버 T_1이 서비스(즉, 교정)할 일감의 도착과정은 도착률이 λ_1인 포아송과정이 된다. 그리고 <가정 3>에 의해서 T_1의 서비스시간은 (평균이 μ_1^{-1}) 인 i.i.d. 일반분포를 따르고 고객의 도착과정과 독립인데, <가정 4>에 의해서 이 시스템은 안정적(stable)이다.

서버가 연속적으로 서비스를 제공하는 기간을 바쁜기간이라 한다. 반면에, 서버가 바쁘지 않은 기간을 유휴기간이라 하는데, 하나의 유휴기간은 두 개의 구간으로 구성된다. 첫째 구간은 고객이 한 명도 없는 구간인데, 이는 실제로는 T_1이 보관중이던 일감을 T_2가 넘겨받는 시점으로부터 T_2가 진행중이던 타이핑을 끝낼 때까지 소요되는 시간이다. 그 다음, 실제로는 T_2가 타이핑을 중단하고 교정을 시작하지만 이 구간은 시간축에서 제외되었다(비고: T_1의 관점으로는 이 구간 동안 고객도 도착하지 않고 서비스도 제공하지 않으므로 모든 활동이 중지된 구간이다). 유휴기간의 둘째 구간은 실제로는 T_2가 보관했던 일감을 T_1이 넘겨받은 시점으로부터 T_1이 진행중이던 타이핑을 끝낼 때까지 소요되는 시간인데, 이는 T_1의 관점으로는 유휴기간 중에 첫 고객이 도착한 직후 바쁜 기간에 대한 준비작업을 하는 시간으로서 평균이 λ_2^{-1}인 지수분포를 따른다.

2.2 2단계 분석

1단계에서 제거했던 시간구간들을 다시 시간축에 포함시키기 위해서 정식으로 주기분석(cycle analysis)을 한다.

T_1의 관점에서 바쁜 기간을 B_1이라 하고 유휴기간의 첫째와 둘째 구간을 각각 I_1과 I_2라 하자. 그리고 제거했던 구간을 B_2라 하자. 그러면 하나의 재생(regeneration) 주기 C는 I_1, B_2, I_2, B_1의 순서로 구성된다.

[비고 2] 만약, 1단계에서 T_2를 서버라 하고 T_1이 교정중인 구간인 B_1을 시간축에서 제거하면, T_2의 바쁜 기간은 B_2이고 유휴기간의 첫째와 둘째 구간은 각각 I_2와 I_1이다.

T_1의 관점에서 안정상태(steady-state) 고객수의 PGF(probability generating function) $P_1(z)$는 관찰시점이 B_2에 속하는지 아닌지에 조건을 걸어서

$$P_1(z) = P\{B_2\}P_1(z|B_2) + P\{\overline{B_2}\}P_1(z|\overline{B_2}) \qquad (1)$$

로 표현할 수 있는데, 여기서 $P\{B_2\}$와 $P\{\overline{B_2}\}$는 각각 관찰시점이 B_2에 속할 확률과 아닐 확률이고, 이에 따른 조건부 PGF는 각각 $P_1(z|B_2)$와 $P_1(z|\overline{B_2})$이다.

B_2 동안에는 T_2가 임시로 보관중인 일감 하나가 있으므로

$$P_1(z|B_2) = z^1 = z \tag{2}$$

이다. 그리고, B_2를 제거한 1단계에서는 준비기간을 갖는 M/G/1 시스템이 되므로, 고객수 PGF를 T_1의 서비스시간의 LST(Laplace-Stieltjes transform)인 $S_1^*(\Theta)$로 표현하면 다음과 같다(Lee, 1998).

$$P_1(z|\overline{B_2}) = \left[\frac{(1-\rho_1)(1-z)S_1^*(\lambda_1-\lambda_1 z)}{S_1^*(\lambda_1-\lambda_1 z)-z}\right]\left[\frac{\lambda_2}{\lambda_2+\lambda_1-\lambda_1 z}\right] \tag{3}$$

$P\{B_2\}$와 $P\{\overline{B_2}\}$ $(=1-P\{B_2\})$는 재생보상정리(renewal reward theorem)로 구할 수 있는데, 이때 활용할 수 있는 알려진 결과는 다음과 같다. B_2를 제거한 1단계의 준비기간을 갖는 M/G/1 시스템에서 서버가 바쁠 확률은 $\rho_1(=\lambda_1\mu_1^{-1})$이므로, 2단계에서는

$$\lambda_1\mu_1^{-1} = E[B_1]/\{E[I_1]+E[I_2]+E[B_1]\} \tag{4}$$

이 된다. 식 (4)에 $E[I_i]=\lambda_i^{-1}$, $i=1,2,$ 를 대입하고 $E[B_1]$에 대해서 풀면 $E[B_1]=(\lambda_1+\lambda_2)/\{\lambda_2(\mu_1-\lambda_1)\}$을 얻으므로, 대칭성에 의해서([비고 2] 참조) $E[B_2]=(\lambda_2+\lambda_1)/\{\lambda_1(\mu_2-\lambda_2)\}$를 얻는다. 따라서, $E[C]$와 $P\{B_2\}$는 다음과 같다.

$$E[C] = \sum_{i=1}^{2}\{E[I_i]+E[B_i]\} = \frac{(\lambda_1+\lambda_2)(\mu_1\mu_2-\lambda_1\lambda_2)}{\lambda_1\lambda_2(\mu_1-\lambda_1)(\mu_2-\lambda_2)} \tag{5}$$

$$P\{B_2\} = E[B_2]/E[C] = \lambda_2(\mu_1-\lambda_1)/(\mu_1\mu_2-\lambda_1\lambda_2) \tag{6}$$

식 (2), (3), (6)을 식 (1)에 대입하고 간단히 하면 다음과 같다.

$$P_1(z) = \frac{\lambda_2(\mu_1-\lambda_1)}{\mu_1\mu_2-\lambda_1\lambda_2} \times \left[z + \frac{(1-z)S_1^*(\lambda_1-\lambda_1 z)}{S_1^*(\lambda_1-\lambda_1 z)-z}\cdot\frac{\mu_2-\lambda_2}{\lambda_2+\lambda_1-\lambda_1 z}\right] \tag{7}$$

3. (N_1, N_2)-정책을 갖는 Push-Pull 대기행렬의 분석

K&W의 방법을 확장하여 T_1과 T_2가 임시로 보관했다가 나중에 상대방에게 넘겨주는 일감의 수를 각각 N_2와 N_1이라 하자.

3.1 (N_1, N_2)-정책의 작동원리

2장의 "$N_1N_2=1$"인 경우에는 재생주기 C가 I_1, B_2, I_2, B_1의 순서로 구성되었으나, "$N_1N_2\neq 1$"인 경우는 순서가 확정적이 아니라 확률적이며 그 내막도 다소 복잡하다.

시스템을 처음 가동시키는 시점에서 T_1과 T_2는 모두 타이핑을 시작하는데, 이때 교정할 일감은 하나도 없다([비고 1] 참조). 이러한 초기상태는 T_1이 N_2개의 타이핑 작업을 끝내든지 또는 T_2가 N_1개의 타이핑 작업을 끝낼 때까지 지속된다. T_1이 N_2개를 타이핑하기 전에 T_2가 먼저 N_1개를 타이핑했다고 하자. 그러면 T_2는 N_1개의 일감을 T_1에게 넘겨주고 T_1은 진행중이던 타이핑을 끝내는 즉시 교정을 시작하는데, 이때 T_1이 타이핑한 일감들은 T_1이 임시로 보관한다.

[비고 3] T_2가 N_1개를 타이핑하는 동안 T_1이 타이핑한 개수는 N_2개 '미만'이다. 그러나 여기에 진행중이던 타이핑 1개를 더할 경우 T_1이 보관하는 일감의 수는 총 N_2개 '이하'가 된다.

T_1이 N_1개의 일감을 넘겨받는 시점으로부터 진행중이던 타이핑을 끝낼 때까지 소요되는 시간을 I_2라 하자. 무기억속성에 따라 I_2는 평균이 λ_2^{-1}인 지수분포를 따르는데, T_1의 관점으로 I_2는 바쁜기간에 앞선 준비기간에 해당된다. 따라서, I_2의 의미는 2장에서와 같으나, 다만 I_2의 시작시점에 1개가 아니라 N_1개의 일감이 있다는 점이 다르다. 이후 당분간 T_2는 타이핑만 하고 T_1은 교정만 하는데, 이는 T_1의 관점으로 바쁜 기간이므로 이 구간을 B_1이라 하자.

T_1이 교정할 일감이 떨어지는 시점에서 B_1이 끝나는데, 이때 T_1이 임시로 보관하고 있던 일감의 수를 K라 하면 [비고 3]에 의해서 "$K\le N_2$"이다. 만약 "$K=N_2$"이면 T_1은 T_2에게 N_2개의 일감을 넘겨주고, 이 순간 T_2의 관점으로 준비기간인 I_1이 시작되며, T_2가 진행중이던 타이핑을 끝내는 시점에 I_1이 끝나고 T_2의 바쁜 기간인 B_2가 시작된다. 그렇지만, "$K<N_2$"인 경우에는 T_1과 T_2가 모두 타이핑을 하는 구간이 발생하는데, 이 구간은 T_1이 'N_2-K'개를 타이핑할 때까지 또는 T_2가 N_1개를 타이핑할 때까지 지속된다. 그리고, T_1이 먼저 'N_2-K'개를 타이핑하면 이후 I_1과 B_2 구간이 발생하고, T_2가 먼저 N_1개를 타이핑하면 이후 I_2와 B_1 구간이 발생한다.

하나의 B_1이 끝나는 시점으로부터 다음 B_1이 끝날 때까지의 시간을 C_1이라 하자. T_1의 관점으로 바쁜 기간인 B_1 앞에는 준비기간인 I_2 하나가 반드시 있다. 그리고, 위에서 설명했듯이, C_1 동안에 B_2가 하나도 없을 수도 있고 여러 개가 있을 수도 있다. 물론, T_2의 관점으로 바쁜 기간인 B_2 앞에는

준비기간인 I_1 하나가 반드시 있다. 더욱이 C_1에 속했으되 I_1과 I_2도 아니고 B_1과 B_2도 아닌 구간들도 존재할 수 있다.

이와 같이 (N_1, N_2)-정책하에서는 C_1의 내막이 다소 복잡하다. 그렇지만, 2장에서와 같이 시간축에서 B_2들을 모두 제거하고 나머지만 고려하는 1단계 분석은 의외로 간단하다.

3.2 1단계 분석

T_2가 교정중인 B_2를 모두 제거하면, T_1은 준비기간을 갖는 M/G/1/ N_1-정책 대기행렬의 서버가 되고 T_1이 교정할 일감은 고객이 되는데 그 이유는 다음과 같다.

B_2를 모두 제거하면 <가정 1,6,7>에 의해서 T_2는 연속적으로 타이핑만 하므로, <가정 2>에 의해서 고객은 도착률이 λ_1인 포아송과정으로 도착한다. 그리고, <가정 3>에 의해서 서비스시간은 평균이 μ_1^{-1}인 i.i.d. 확률분포를 따르고, 고객의 도착과정과 독립이며, 서버가 1명이므로 기본적인 틀은 M/G/1 시스템인데, <가정 4>에 의해서 이 시스템은 안정적이다.

다음, B_2를 모두 제거한 시간축에서의 주기를 C_1'이라 하면 C_1'은 I_3, I_2, B_1의 순서로 구성되는데, I_3는 N_1명의 고객이 도착할 때까지 소요되는 시간이고, N_1명이 도착한 직후 바쁜 기간에 대비한 준비기간이 I_2이며, 준비기간이 끝난 후에는 바쁜 기간인 B_1이 따르는 것이다.

3.3 2단계 분석

T_1의 관점에서 안정상태 고객수의 PGF는 식 (1)로 표현할 수 있다. 그리고, "$N_1=1$" 경우에 유효한 식 (3)을 "$N_1\geq 2$" 경우로 확장한 결과는 다음과 같이 알려져 있다.

$$P_1(z|\overline{B_2})=\left[\frac{(1-\rho_1)(1-z)S_1^*(\lambda_1-\lambda_1 z)}{S_1^*(\lambda_1-\lambda_1 z)-z}\right] \times\left[\frac{\lambda_2-\dfrac{\lambda_2^2 z^{N_1}}{\lambda_2+\lambda_1-\lambda_1 z}}{(1-z)(\lambda_1+N_1\lambda_2)}\right] \tag{8}$$

[비고 4] 식 (8)은 Hur and Ahn(2005)이 제시한 '준비기간을 갖는 M^X/G/1/ N-정책 시스템'에 대한 결과에 "$X=1$"을 대입하여 얻은 것인데, "$X=1$"은 고객이 집단으로 도착하지 않고 한 명씩 도착함을 의미한다.

$P\{B_2\}$는 다음과 같이 구한다. 2장의 "$N_1N_2=1$" 경우에는 주기당 하나의 B_2가 있으므로 "$P\{B_2\}=E[B_2]/E[C]$"였으나 "$N_1N_2\neq 1$" 경우에는

$$P\{B_2\}=E[\#B_2]E[B_2]/E[C_1] \tag{9}$$

인데, 여기서 $\#B_2$는 C_1에 포함된 B_2의 개수이다. $E[\#B_2]$는 다음과 같다(유도과정은 부록에 있음).

$$E[\#B_2]=(\lambda_1+N_1\lambda_2)/(\lambda_2+N_2\lambda_1) \tag{10}$$

2.2절에서 $E[B_2]$를 계산할 때 활용한 식 (4)는 이제

$$\lambda_1\mu_1^{-1}=E[B_1]/\{E[I_3]+E[I_2]+E[B_1]\} \tag{11}$$

로 확장되는데, 이는 B_2를 제거한 1단계에서 서버가 바쁠 확률이 N_1-정책하에서도 여전히 $\rho_1(=\lambda_1\mu_1^{-1})$임을 의미한다 (Lee(1998)). I_3는 도착률이 λ_1인 포아송과정으로 N_1명의 고객이 도착할 때까지 걸리는 시간이므로 "$E[I_3]=N_1\lambda_1^{-1}$"이고, $E[I_2]$는 종전대로 λ_2^{-1}이다.

$E[B_1]$에 대한 방정식인 식 (11)에 이들을 대입하고 풀면 "$E[B_1]=(\lambda_1+N_1\lambda_2)/\{\lambda_2(\mu_1-\lambda_1)\}$"을 얻으므로, 대칭성에 의해서 "$E[B_2]=(\lambda_2+N_2\lambda_1)/\{\lambda_1(\mu_2-\lambda_2)\}$"를 얻는다. 그리고, $E[C_1]$은 다음과 같다.

$$\begin{aligned}E[C_1]&=E[I_3]+E[I_2]+E[B_1]+E[\#B_2]E[B_2]\\&=(\lambda_1+N_1\lambda_2)(\mu_1\mu_2-\lambda_1\lambda_2)/\{\lambda_1\lambda_2(\mu_1-\lambda_1)(\mu_2-\lambda_2)\}\end{aligned} \tag{12}$$

이들을 식 (9)에 대입하고 정리하면 다음과 같이 예상치 못했던 결과를 얻을 수 있다.

$$P\{B_2\}=\lambda_2(\mu_1-\lambda_1)/(\mu_1\mu_2-\lambda_1\lambda_2) \tag{13}$$

즉, 식 (13)은 (6)과 동일한데, 이는 N_1과 N_2가 $P\{B_2\}$에 영향을 전혀 미치지 않음을 의미한다.

반면에, 식 (2)는 다음과 같이 다소 복잡한 형태로 확장된다 (유도과정은 부록에 있음).

$$P_1(z|B_2)=\frac{\lambda_2 z}{\lambda_1+N_1\lambda_2}\left[\frac{1-z^{N_1}}{(1-z)N_1}+\frac{\lambda_1}{\lambda_2}\right] \tag{14}$$

식 (8), (9), (14)를 식 (1)에 대입하고 간단히 하면 다음과 같다.

$$P_1(z)=\left[\frac{\mu_2(\mu_1-\lambda_1)}{\mu_1\mu_2-\lambda_1\lambda_2}\right]\left[\frac{1}{\lambda_1+N_1\lambda_2}\right]\times\left[z\left\{\lambda_1+\frac{\lambda_2(1-z^{N_1})}{(1-z)N_1}\right\}+\left\{\frac{S_1^*(\lambda_1-\lambda_1 z)}{S_1^*(\lambda_1-\lambda_1 z)-z}\right\}\left\{\frac{\lambda_1(1-z)+\lambda_2(1-z^{N_1})}{\lambda_1(1-z)+\lambda_2}\right\}\right] \tag{15}$$

4. 결론 및 토의

본 논문에서는 K&W가 push-pull 개념을 대기행렬 모형에 도입하면서 분석의 편의상 사용했던 방법을 (N_1, N_2)-정책으로 확장하여 분석하였는데, K&W의 방법은 "$N_1=N_2=1$"인 특

수한 경우에 해당된다.

구체적으로, T_1의 관점에서 주기분석을 하였고 그 결과를 사용해서 T_1의 관점에서 안정상태 고객수의 PGF인 식 (15)의 $P_1(z)$를 얻었다. 식 (15)에 "$N_1=1$"을 대입하면 K&W의 결과인 식 (7)을 얻는다. T_2의 관점에서 안정상태 고객수의 PGF를 $P_2(z)$라 하면 이는 대칭성에 의해서 식 (15)의 우변에서 λ_1, λ_2, μ_1, μ_2, S_1, N_1을 각각 λ_2, λ_1, μ_2, μ_1, S_2, N_2로 대체하여 얻을 수 있다.

$P_1(z)$와 $P_2(z)$를 미분하여 기대치를 얻으면, 이를 사용하여 최적 (N_1, N_2)-정책을 얻을 수 있으나(Lee(1998)), 지면 관계상 추후과제로 남긴다. 최적 (N_1, N_2)-정책에 대한 저자의 추측(conjecture)은 다음과 같다. 식 (15)의 $P_1(z)$가 N_2와 무관하고 또한 대칭성에 의해서 $P_2(z)$가 N_1과 무관하므로, 최적 N_1-정책과 최적 N_2-정책을 각각 구한 다음 이들을 묶으면 최적 (N_1, N_2)-정책이 될 것으로 사료된다. 따라서, T_1 관점의 고객수와 T_2 관점의 고객수의 결합(joint) PGF는 불필요할 것으로 사료된다.

식 (15)의 $P_1(z)$가 N_2와 무관한 점에 대한 (결과론적인) 해석은 다음과 같다. "$E[B_2]=(\lambda_2+N_2\lambda_1)/\{\lambda_1(\mu_2-\lambda_2)\}$"이므로 $E[B_2]$는 '$\lambda_2+N_2\lambda_1$'에 정비례한다. 그런데 식 (10)에 의해서 $E[\#B_2]$는 '$\lambda_2+N_2\lambda_1$'에 반비례한다. 따라서, $E[\#B_2]E[B_2]$에서는 '$\lambda_2+N_2\lambda_1$'이 상쇄되어 N_2와 무관하게 된다. 이에 따라, 식 (12)의 $E[C_1]$과 식 (13)의 $P\{B_2\}$는 N_2와 무관하게 된다.

나아가서, $P\{B_1\}$과 $P\{B_2\}$는 N_1과 N_2 모두에 영향을 받지 않는데, 이는 1단계 분석에서 T_1이 바쁠 확률이 N_1과 무관하게 항상 $\rho_1(=\lambda_1\mu_1^{-1})$이었던 점과 유사하다.

마지막으로, <가정 7>이 없는 모형의 분석을 추후과제로 남긴다. 타이핑을 끝낸 일감을 임시로 보관하지 않고 바로 상대방에게 보내면 평균고객수와 평균대기시간을 감소시킬 수 있다. 하지만, T_1과 T_2가 모두 바쁜 구간이 발생하기 때문에 분석이 어려워지는데, 이는 T_1의 관점에서 바쁜 기간 중 일부 구간에서 고객이 도착하지 않는 상황이 발생하기 때문이다.

부록: 식 (10)과 (14)의 유도

식 (14)의 $P_1(z|B_2)$는 B_2 동안에 T_2가 임시로 보관중인 일감의 수의 PGF이다. 여기서 유의할 점은 두 가지인데, 첫째는 T_2가 보관 중인 일감수가 B_2 동안 변화하지 않는다는 점이고 둘째는 B_2 동안 T_1이 보관중인 일감은 없다는 점이다. 그리고 대칭성에 의해서, B_1 동안에는 T_1이 보관중인 일감수가 변화하지 않으며 이 동안 T_2가 보관중인 일감은 없다.

내재(embedded) 마코프체인을 다음과 같이 정의하자. T_i가 보관중인 (상대방이 교정할) 일감의 수를 n_i라 하고 (단, $i=1,2$), 마코프체인의 상태를 (n_1, n_2)로 정의한다. 그리고 상태를 관찰하는 내재점은 n_1과 n_2 중에서 하나가 증가한 직후로 정의한다. 단, 'N_1-1'에서 N_1으로 증가하거나 'N_2-1'에서 N_2로 증가한 시점의 경우에는 편의상 상대방에게 N_1개 또는 N_2개를 넘겨주기 직전으로 정의한다.

편의상, p와 q를 다음과 같이 정의한다.

$$p=\lambda_2/(\lambda_1+\lambda_2),\quad q=1-p \tag{A.1}$$

T_1과 T_2가 모두 타이핑을 하고 있을 때 T_1이 먼저 타이핑 하나를 끝낼 확률이 p이고 T_2가 먼저 끝낼 확률이 q이다 (Lee, 1998)).

초기상태(initial state)는 $(0,0)$이다. 그리고 그 다음 상태는 확률 p로 $(1,0)$이 되거나 확률 q로 $(0,1)$이 된다. 이후 $(0,0)$은 다시 방문하지 않으므로 $(0,0)$을 상태공간에서 제외한다([비고 1] 참조). "$0\le n_1\le N_2-1$, $0\le n_2\le N_1-1$, $n_1+n_2\ge 1$" 일 때, (n_1, n_2)이던 상태가 (n_1+1, n_2)로 전이할 확률은 p이고 (n_1, n_2+1)로 전이할 확률은 q이다.

그러다가 언젠가는 (N_2, n_2) 또는 (n_1, N_1)으로 전이하게 되는데, 이때 "$0\le n_2\le N_1-1$, $0\le n_1\le N_2-1$"이다. 상태 (N_2, n_2)의 의미는 다음과 같다. T_1이 보관하던 N_2개의 일감을 T_2에게 넘겨주면 T_2의 준비기간이 시작된다. 그런데, 준비기간이란 진행중이던 타이핑을 끝낼 때까지 걸리는 시간이므로, 준비기간이 끝나고 바쁜 기간인 B_2가 시작될 때 T_2가 보관하는 일감의 수는 'n_2+1'이 된다. 그리고 앞에서 언급했듯이 B_2 동안에 T_2가 보관하는 일감의 수는 계속 'n_2+1'이다. 또한, T_2의 준비기간과 바쁜 기간 동안 T_1이 타이핑한 일감은 T_1이 보관하지 않고 바로바로 T_2에게 보내기 때문에 이 동안 T_1이 보관하고 있는 일감의 수는 0이다. 따라서, 상태 (N_2, n_2)로 전이한 후 다음 방문하는 상태는 $(0, n_2+1)$인데, 이때 전이확률은 1이다. 그리고 대칭성에 의해서, 상태 (n_1, N_1)으로부터 $(n_1+1, 0)$로의 전이확률도 1이다.

안정상태에서 내재 마코프체인의 상태가 (n_1, n_2)일 확률을 $\Pi(n_1, n_2)$라 하자. 위에서 얻은 전이확률을 평형방정식(balance equation)에 대입해서 풀면 다음과 같다(비고: 평형방정식을 푸는 과정은 다소 복잡하나 평형방정식에 대입해서 확인하는 것은 간단함).

$$\Pi(N_2, n_2)=p\Pi(N_2, 0),\quad 0\le n_2\le N_1-1 \tag{A.2}$$

$$\Pi(n_1, N_1)=q\Pi(0, N_1),\quad 0\le n_1\le N_2-1 \tag{A.3}$$

$$\Pi(n_1, n_2)=\Pi(0, N_1)=\Pi(N_2, 0),$$
$$0\le n_1\le N_2-1,\ 0\le n_2\le N_1-1,\ 1\le n_1+n_2\le N_1+N_2-1 \tag{A.4}$$

식 (10)은 다음과 같이 얻은 것이다.

$$E[\#B_2] = \sum_{n_2=0}^{N_1-1} \Pi(N_2, n_2) / \sum_{n_1=0}^{N_2-1} \Pi(n_1, N_1) \tag{A.5}$$

식 (A.5)의 분모와 분자는 안정상태에서 임의의 전이 후에 각각 B_1과 B_2 구간이 발생할 확률이다.

마지막으로, 식 (14)는 다음과 같이 얻은 것이다.

$$\begin{aligned} P_1(z|B_2) &= \sum_{n_2=0}^{N_1-1} z^{n_2+1} \Pi(N_2, n_2) / \sum_{n_2=0}^{N_1-1} \Pi(N_2, n_2) \\ &= \left(qz + pz \sum_{n_2=0}^{N_1-1} z^{n_2}\right) / (q + pN_1) \end{aligned} \tag{A.6}$$

참고문헌

Bonney, M.C., Zhang, Z., Head, M.A., Tien, C.C., and Barson, R.J. (1999), Are Push and Pull Systems Really So Different?, *Int. J. Prodution Economics*, **59**, 53-64.

Hur, S. and Ahn, S. (2005), Batch Arrival Queues with Vacations and Server Setup, *Applied Mathematical Modelling*, **29**, 1164-1181.

Kopzon, A. and Weiss, G (2002), A Push-Pull Queueing System, *Operation Research Letters*, **30**, 351-359.

Lee, H.W. (1998), *Queueing Theory*, Sigma Press, Seoul, Korea.