

# 확률적 시간 넷과 최소 순회 시간 분석

임재걸\*, 심규박\*\*

## 요 약

'최소 순회 시간' 방법은 '시간 넷'(Timed Net)에서 모든 변천(transition)들을 최소한 한번 이상 격발하고 초기 마킹으로 되돌아오는데 걸리는 최소 시간을 찾아주는 수학적 방법으로, 컴퓨터 시스템 분석에 널리 사용된다. 시간 넷은 변천에 지연 시간이 연합된 변형된 페트리 넷으로, 실세계의 사건이 실행되는데 걸리는 시간이 일반적으로 확률적 시간인데 반하여 시간 넷에 사용되는 지연시간은 상수이기 때문에 최소 순회 시간 방법에 의한 분석 결과는 현실성이 부족하다. 그래서 본 논문은 사건의 실행 시간을 확률적 숫자로 표현하는 '확률적 시간 넷'을 제안하고 '확률적 시간 넷'의 최소 순회 시간 분석 방법을 소개한다. 또한 '확률적 시간 넷'의 적용 예로 위치 기반 서비스를 제공하는 멀티미디어 시스템을 '확률적 시간 넷'으로 모델링하고 분석한다. 기존의 최소 순회 시간 분석 결과의 전형적인 형태는 "최소 10 단위 시간 이상 걸린다."와 같은 형태인데 반하여 '확률적 시간 넷'의 분석 결과의 형태는 "10 단위 시간 이내일 확률이 85%이다."와 같은 확률적 형태이다.

## Stochastic Timed Net and Its Minimum Cycle Time Analysis

Jae-Geol Yim\*, Kyu-Bark Shim\*\*

## ABSTRACT

As a mathematical technique with which we can find the minimum duration time needed to fire all the transitions at least once and coming back to the initial marking in a timed net, the minimum cycle time method has been widely used in computer system analysis. A timed net is a modified version of a Petri net where a transition is associated with a delay time. A delay time used in a timed net is a constant even though the duration time associated with an event in the world is a stochastic number in general. In the consequence, the result of minimum cycle time analysis is not realistic. Therefore, we propose 'Stochastic Timed Net' where a transition can be associated with a stochastic number and introduce a minimum cycle time analysis method for 'Stochastic Timed Net.' As an example of the application of 'Stochastic Timed Net', we introduce a 'Stochastic Timed Net' model of a Location Based Service Providing Multimedia System and the result of minimum cycle time analysis of it. Whereas the typical form of the result of the existing minimum cycle time analysis is "It takes at least 10 time units", the typical form of the result of minimum cycle time analysis of a 'Stochastic Timed Net' is in the probability form such as "The probability of the events in which it finishes its job within 10 time units is 85%."

**Key words:** Petri Net(페트리 넷), Stochastic Petri Net(확률적 페트리 넷), Timed Net(시간 넷), Minimum Cycle Time(최소순회시간), System Analysis(시스템 분석)

\* 교신저자(Corresponding Author) : 심규박, 주소 : 경북 경주시 석장동 707(780-714), 전화 : 054)770-2245, FAX : 054)770-2520, E-mail : gpshim@dongguk.ac.kr

접수일 : 2005년 11월 1일, 완료일 : 2005년 12월 12일  
\* 정회원, 동국대학교 과학기술대학 컴퓨터멀티미디어학과 교수

(E-mail : yim@dongguk.ac.kr)

\*\* 정회원, 동국대학교 과학기술대학 정보통계학과 교수  
\* 이 논문은 2005년도 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2005-041-D00633).

### 1. 서론

페트리 넷은 1962년에 처음 소개된 이후 시스템 성능 테스트와 통신 규약의 일관성 및 타당성 테스트 등의 분야에 성공적으로 사용되었다. 또한 이는 분산 소프트웨어시스템, 분산데이터베이스 시스템, 병렬 시스템, 유연생산시스템, 사무자동화시스템, 형식언어 등을 비롯한 다양한 분야에서 모델링과 분석 도구로서 각광을 받고 있다[1].

페트리 넷 성능 분석 방법 중 반복적으로 수행되는 과업의 작업 시간을 측정하는 수단으로

‘최소 순회시간(minimum cycle time)’ 분석 방법이 있다. 페트리 넷의 변천(transition)에, 변천과 연관된 사건이 실행되는데 걸리는 시간을 첨부하여 놓은 것을 ‘시간 넷’이라 하고, ‘시간 넷’으로 생산공정의 모델을 구축하고, 주기적으로 수행되는 일련의 작업이 실행되는데 걸리는 시간을 ‘시간 넷(timed net)’을 분석하여 알아내는 방법이 Hillion[2]에 소개되었다.

최소 순회시간을 찾는 기존 알고리즘의 효율성을 개선하는 연구 결과가 Dasdan and Gupta[3]에 발표되었으며, 모든 장소의 입력 간선과 출력 간선이 정확히 하나인 특수한 페트리 넷의 최소 순회시간을 찾는 방법이 Nakamura and Silva[4]에 소개되었다. Rodriguez-Beltran and Ramfrez-Trevino [5]에는 순회시간이 주어졌을 때, 실제 순회시간이 주어진 순회시간이 되도록 초기 마킹을 결정하는 문제에 대한 해가 소개되었다.

실세계의 사건이 실행되는데 걸리는 시간은 일반적으로 상수가 아니라 평균 얼마와 같은 확률적인 숫자인 경우가 보통이다. 그러나 기존의 최소 순회 시간 분석에서는 변천에 연합된 수가 상수인 경우만 다루었다. 본 논문은 변천에 연합된 사건의 실행 시간이 정규분포, 감마 분포, 등 확률 분포를 이룰 경우에 최소 순회 시간을 분석하는 방법을 소개한다. 이를 위하여 3절에서 변천에 결합된 시간이 확률분포를 이루는 ‘확률적 시간 넷’을 정의하고, ‘확률적 시간 넷’의 최소 순회 시간 분석 방법을 소개하며, 4절에서 ‘확률적 시간 넷’을 이용하여 위치기반서비스(Location Based Service: LBS)를 제공하는 멀티미디어 시스템을 분석하는 적용 예를 제시한다.

### 2. 기존의 연구

여기서는 확률적 시간 넷을 고안하기에 앞서 그 기초가 되는 기존의 최소 순회시간 분석 방법과 확률적 시간 넷에 사용된 확률 이론을 소개한다. 2.1 절에서 최소 순회시간 방법에 대한 설명을 페트리 넷에 대한 기초 지식이 없더라도 이해할 수 있도록, 예를 통하여 설명하고[1,6,7,8], 2.2 절에서 통계 이론을 소개한다.

#### 2.1 페트리 넷과 최소 순회 시간 분석 방법

페트리 넷의 변천에 시간이 첨부되어 있는 것을 시간 넷이라 하며, 간단한 시간 넷의 예는 (그림 1)과 같다. (그림 1)은 데이터베이스 검색이나 갱신 작업을 실행하는 프로세스를 나타낸다.  $p_1$ 이라고 쓰인 원은 프로세스를 나타내며, 이 원 안에 쓰인  $k$ 는 현재 실행 중인 프로세스가  $k$ 개 있다는 뜻이다. 각 프로세스는 계산과 같은 고유의 작업을 수행하다가 데이터베이스에 접근하여 검색이나 갱신 작업을 한다. 갱신 작업의 시작을 나타내는 것이  $t_2$ 라고 쓰인 사각형이고, 검색 작업의 시작을 나타내는 것은  $t_1$ 이라고 쓰인 사각형이다. 갱신 작업 중인 프로세스는  $p_4$ 라고 쓰인 원 안에, 검색 중인 프로세스는  $p_2$ 라고 쓰인 원 안에 머문다.  $t_4$ 라고 쓰인 사각형과  $t_3$ 이라고 쓰인 사각형은 각각 갱신 작업과 검색 작업의 종료를 의미하며, 작업을 종료한 프로세스는 다시  $p_1$ 이라고 쓰인 원으로 되돌아간다.  $p_3$ 이라고 쓰인 원은 프로세스가 공유하는 데이터가 머무는 장소이며, 본 모델은 갱신 작업시 데이터베이스 전체를 독점한다고 가정하였다. 검색을 원하는 프로세스들은 가용 자원이 허락하는 한 동시에 검색을 수행할 수 있다. 가용자원의 수는

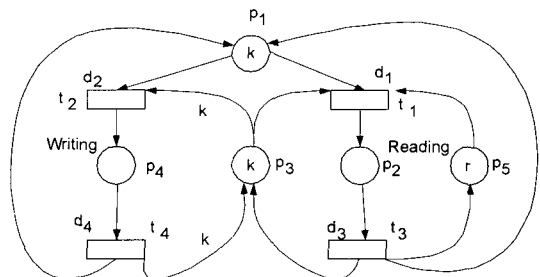


그림 1. 데이터베이스 검색과 갱신 작업을 수행하는 프로세스를 나타내는 시간 넷 모형

$p_5$ 라고 쓰인 원에 표시된다. 다음에서 이 시간 넷의 작동 원리를 설명하고자 한다.

페트리 넷은 이분유향그래프이다. 이분이란 정점이 두 가지 그룹으로 분할되고, 동일한 그룹에 속하는 정점들 간에는 간선이 존재하지 않는 것을 의미하며, 유향이란 간선에 방향이 있다는 것을 의미한다. (그림 1)의 정점들은 사각형 모양과 원 모양으로 분류된다. 사각형 모양의 정점을 변천 (transition)이라고 하고, 원 모양의 정점을 장소 (place)라고 일컫는다. 실세계의 현상을 페트리 넷으로 나타낼 때, 일반적으로 사건은 변천으로, 자원은 장소에 나타내는데, 사건이 수행되면 상태가 변천하고, 자원이 존재하는 곳은 장소임으로 페트리 넷의 정점들의 분류의 이름을 그렇게 명명한 것이다.

그래프의 정점들이 이름을 갖듯이 장소와 변천의 각 원소도 보통 고유의 이름을 갖는다. 실제 시스템을 나타내는 모델의 경우에는 정점과 연합된 시스템의 구성 요소를 나타내는 의미 있는 이름을 사용하지만, (그림 1)의 경우에는 무의미한 일련번호를 사용하여 장소는  $p_1, p_2, \dots, p_5$ 라 하고, 변천은  $t_1, t_2, t_3, t_4$ 라고 명명하였다. (그림 1)의 경우에는 모든 변천 각각에 지연시간이 부착되어 있으며, 그 값은 변수  $d_1, d_2, d_3, d_4$ 로 표기되었다. 그러나 지연시간이 0일 경우에는 표기를 생략할 수 있다.

장소는 일반적으로 자원을 나타내므로, 장소의 이름은 해당 자원 이름을 쓴다. 가용 자원의 분포를 나타내기 위하여 각 장소에 자원의 수를 표기하는데, (그림 1)의 경우에는 장소  $p_1$ 에 해당하는 자원  $k$ 개가 사용 가능함을  $p_1$ 의 내부에  $k$ 라는 레이블로 표기 하였으며, 'p1에 k개의 토큰이 있다'라고 말한다. 장소  $p_3$ 에도  $k$ 개의 토큰이 있음을 알 수 있다.

실세계에서 사건이 발생하여 상태가 변화하는 현상을 페트리 넷에서는 변천의 격발로 시뮬레이션한다. 격발 규칙은 다음과 같다.

1. 하나의 변천  $t$ 에 대하여, 입력 장소  $p$ 가 최소한  $W(p, t)$  만큼의 토큰을 갖고 있으면,  $t$ 는 장전되었다고 한다. 단,  $W(p, t)$ 는 간선  $(p, t)$ 의 레이블이다.
2. 장전된  $t$ 는 격발될 수도 있고, 아니 될 수도 있다.
3.  $t$ 의 격발은  $W(p, t)$  만큼의 토큰을 각 입력 장소  $p$ 에서 제거하고,  $W(t, p)$  만큼의 토큰을 각 출력 장소에 더하여 준다. 단,  $W(t, p)$ 는 간선  $(t, p)$ 의

레이블이다.

(그림 1)의 변천  $t_2$ 를 예로 들면,  $t_2$ 의 입력 장소는  $p_1$ 과  $p_3$  두 개다. 간선  $(p_1, t_2)$ 의 레이블은 그림에 명시되어 있지 않은데, 일반적으로 간선의 묵시적인 레이블은 1이므로, 간선  $(p_1, t_1)$ 의 레이블은 1이라고 할 수 있다. 한편, 간선  $(p_3, t_2)$ 의 레이블은  $k$ 라고 쓰여 있다. 격발 규칙에 의하여  $t_2$ 가 격발하려면  $p_3$ 에  $k$ 개의 토큰이 있어야 하고,  $t_2$ 가 격발하면  $p_3$ 에 있는  $k$ 개의 토큰이 모두 소비된다. 한편,  $t_1$ 의 격발은 장소  $p_1, p_3, p_5$ 에서 각각 하나씩의 토큰을 사용하므로  $t_1$ 은 계속 여러 번 격발이 가능하다. 이에 반하여,  $t_2$ 의 경우에는  $p_3$ 에  $k$ 개의 토큰이 있어야 함으로,  $p_2$ 에 토큰이 하나라도 있으면 격발이 불가능하고,  $t_2$ 가 격발하여  $p_4$ 에 토큰이 있으면  $t_1$ 의 격발이 불가능하다. 즉, 어떤 시점에서 갱신은 단 하나의 프로세스만 실행할 수 있는데 반하여, 검색은 다수의 프로세스가 동시에 실행 가능한 현상을 (그림 1)이 나타냄을 알 수 있다.

초기에 각 장소에 놓인 토큰의 수를 나열한 벡터를 초기마킹이라 한다. (그림 1)의 경우에 초기마킹  $M_0$ 은 다음과 같다:

$$M_0 = (k, 0, k, 0, r).$$

그래프 이론에서, 그래프를 나타내는 일반적인 방법으로 인접행렬(Incidence Matrix)을 사용한다. (그림 1)의 인접행렬은 다음과 같다:

$$A = \begin{matrix} & \begin{matrix} p_1 & p_2 & p_3 & p_4 & p_5 \end{matrix} \\ \begin{matrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{matrix} & \begin{bmatrix} -1 & 1 & -1 & 0 & -1 \\ -1 & 0 & -k & 1 & 0 \\ 1 & -1 & 1 & 0 & 1 \\ 1 & 0 & k & -1 & 0 \end{bmatrix} \end{matrix}$$

**[정의 1]**  $Ay=0$ 을 만족하는 벡터  $y$ 는 (식 1)의 성질을 만족하므로 S-불변(S-invariant)이라고 한다. (식 1)의  $M_0$ 은 초기마킹,  $M_i$ 는  $M_0$ 으로부터 도달 가능한 임의의 마킹이다.

$$y^T M_i = y^T M_0 \tag{1}$$

그림 1의 경우에는 다음과 같은 S-불변이 있다.

$$(0 \ 1 \ 0 \ 0 \ 1)^T = y_1,$$

$$(1 \ 1 \ 0 \ 1 \ 0)^T = y_2,$$

$$(0 \ 1 \ 1 \ k \ 0)^T = y_3.$$

[정의 2]  $A^T x = 0$ 을 만족하는 벡터  $x$ 를 T-불변(T-invariant)이라 한다.

인접행렬  $A$ 의 각 항  $a_{ij}$ 는 변천  $i$ 의 격발 결과 장소  $j$ 로 입력되는 토큰의 수와, 장소  $j$ 에서 소모되는 토큰의 수를 나타내므로, 각 변천을 T-불변의 해당 항만큼 격발하면 본래의 마킹으로 되돌아오는 성질이 있다.

[정리 1] 각 변천을 T-불변의 해당 항만큼 격발하면 본래의 마킹으로 돌아오는 성질이 있다.

(그림 1)의 경우에는 다음과 같은 T-불변들이 있다.

$$\begin{aligned} x_1 &= (1 \ 0 \ 1 \ 0)^T, \\ x_2 &= (0 \ 1 \ 0 \ 1)^T, \\ x_3 &= (1 \ 1 \ 1 \ 1)^T. \end{aligned}$$

T-불변의 정의에 의하여 정리 1의 역도 참이다. 즉, 주어진 마킹으로부터 격발을 시작하여 이 마킹으로 다시 되돌아오는 격발순서에 나타나는 변천들의 회수로 구성된 벡터  $x_j$ 는 T-불변이다.

[정리 2] 주어진 마킹으로부터 격발을 시작하여 이 마킹으로 다시 되돌아오는 격발순서에 나타나는 변천들의 회수로 구성된 벡터  $x_j$ 는 T-불변이다.

예를 들어, (그림 1)에서  $t_1$ 과  $t_3$ 을 차례로 격발하면 다시 (그림 1)로 되돌아온다. 따라서 이 격발순서의 격발 회수 벡터,  $(1 \ 0 \ 1 \ 0)$ 은 T-불변이다.

인접행렬의 음수 항만 절대 값으로 바꾸고, 나머지 항은 0으로 대체하여 얻은 행렬, 즉 각 장소로 들어오는 간선의 무게만 표기한 행렬을  $A^-$  행렬이라 하며, (그림 1)의 경우에는 다음과 같다:

$$A^- = \begin{matrix} & p_1 & p_2 & p_3 & p_4 & p_5 \\ \begin{matrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{matrix} & \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & k & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{matrix}$$

(그림 1)의 각 변천에 연관된 지연 시간을 나타내는 대각행렬  $D$ 는 다음과 같다.

$$D = \begin{matrix} & t_1 & t_2 & t_3 & t_4 \\ \begin{matrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{matrix} & \begin{bmatrix} d_1 & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 \\ 0 & 0 & d_3 & 0 \\ 0 & 0 & 0 & d_4 \end{bmatrix} \end{matrix}$$

최소 순회시간을 찾는 식은 Murata[1]에 소개되었고, (식 2)와 같다. 단,  $x$ 는  $x_3$ 과 같이 양수 항으로 구성된 T-불변이다.

$$\begin{aligned} & \text{최소 순회 시간} \\ & = \text{Max} \{y_k^T (A^-)^T D x / y_k^T M_0\} \end{aligned} \quad (2)$$

(그림 1)의 시간 넷에 (식 2)를 적용하면, 다음과 같다.

$$(A^-)^T D x = [d_1 + d_2 \ d_3 \ d_1 + kd_2 \ d_4 \ d_1]^T$$

$$y_1 = (0 \ 1 \ 0 \ 0 \ 1)^T \text{ 일때,}$$

$$y_1^T (A^-)^T D x = d_1 + d_3,$$

$$y_1^T M_0 = (0 \ 1 \ 0 \ 0 \ 1)(k \ 0 \ k \ 0 \ r)^T = r$$

$$\text{이므로, } y_1^T (A^-)^T D x / y_1^T M_0 = (d_1 + d_3)/r \text{ 이다.}$$

$$y_2 = (1 \ 1 \ 0 \ 1 \ 0)^T \text{ 일때,}$$

$$y_2^T (A^-)^T D x = d_1 + d_2 + d_3 + d_4,$$

$$y_2^T M_0 = k$$

$$\text{이므로, } y_2^T (A^-)^T D x / y_2^T M_0 = (d_1 + d_2 + d_3 + d_4)/k \text{ 이다.}$$

$$y_3 = (0 \ 1 \ 1 \ k \ 0)^T \text{ 일때,}$$

$$y_3^T (A^-)^T D x = d_1 + d_3 + k(d_2 + d_4),$$

$$y_3^T M_0 = k$$

$$\text{이므로,}$$

$$y_3^T (A^-)^T D x / y_3^T M_0 = d_2 + d_4 + (d_1 + d_3)/k \text{ 이다.}$$

따라서 최소 순회 시간은 다음과 같다.

$$\text{Max}(d_1 + d_3)/r, (d_1 + d_2 + d_3 + d_4)/k,$$

$$d_2 + d_4 + (d_1 + d_3)/k = d_2 + d_4 + (d_1 + d_3)/k$$

즉, 모든 변천을 정확히 한 번씩 격발하고 초기 마킹으로 돌아오는데 걸리는 시간은  $d_2 + d_4 + (d_1 + d_3)/k$  이다. 기존의 연구에서는  $d_1, \dots, d_4$ 가 각각 상수임으로 계산 결과도 상수가 된다.

## 2.2 확률밀도함수

본 절에서는 일반적으로 많이 나타나는 정규분포, 감마(Gamma) 분포, 그리고  $\chi^2$  분포에 대한 기초이

론을 정리하여 소개한다.

1) 정규분포

두 개의 독립적인 확률변수  $X_1, X_2$ 가 각각 평균  $\mu_1, \mu_2$ , 분산  $\sigma_1^2, \sigma_2^2$ 인 정규분포를 따른다면 각각의 확률밀도함수는 다음과 같다.

$$P_{X_1}(x_1) = \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left[-\frac{(x_1 - \mu_1)^2}{2\sigma_1^2}\right],$$

$$-\infty < x_1 < \infty \quad (3)$$

$$P_{X_2}(x_2) = \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left[-\frac{(x_2 - \mu_2)^2}{2\sigma_2^2}\right],$$

$$-\infty < x_2 < \infty \quad (4)$$

두 개의 독립적인 확률변수  $X_1, X_2$ 의 합  $X_1 + X_2$ 의 분포는 다음과 같다.

$$P_{X_1+X_2}(x_1+x_2) = \frac{1}{2\pi\sigma_1\sigma_2} \exp\left[-\left\{\frac{(x_1 - \mu_1)^2}{2\sigma_1^2} + \frac{(x_1 - \mu_2)^2}{2\sigma_2^2}\right\}\right]$$

$$-\infty < x_1 + x_2 < \infty \quad (5)$$

이들의 합  $X_1 + X_2$ 의 평균과 분산은 다음과 같다.

- ① 평균 :  $E(X_1 + X_2)$   
 $= E(X_1) + E(X_2) = \mu_1 + \mu_2$
- ② 분산:  $Var(X_1 + X_2)$   
 $= Var(X_1) + Var(X_2) = \sigma_1^2 + \sigma_2^2$

2) Gamma분포

두 개의 독립적인 확률변수  $X_1, X_2$ 는 각각 모수  $(a_1, \beta)$ 와  $(a_2, \beta)$ 를 갖는 Gamma분포를 따른다고 하자. 이 때, Gamma 분포의 확률밀도함수는 다음과 같다.

$$P_{X_1}(x_1) = \frac{x_1^{a_1-1} \exp[-x_1/\beta]}{\beta^{a_1} \Gamma(a_1)}, \quad a_1 > 0, \beta > 0 \quad (6)$$

$$P_{X_2}(x_2) = \frac{x_2^{a_2-1} \exp[-x_2/\beta]}{\beta^{a_2} \Gamma(a_2)}, \quad a_2 > 0, \beta > 0 \quad (7)$$

두 개의 독립적인 확률변수  $X_1, X_2$ 의 합  $X_1 + X_2$ 의 분포는 모수  $\beta$ 가 같다는 가정 하에서 다음과 같은 Gamma 분포의 확률밀도함수이다.

$$P_{X_1+X_2}(x_1+x_2) = \frac{(x_1+x_2)^{a-1} \exp[-(x_1+x_2)/\beta]}{\beta^a \Gamma(a)},$$

$$, \quad a > 0, \beta > 0 \quad (8)$$

여기서,  $a = a_1 + a_2$ 이다.

이 때, 합  $X_1 + X_2$ 의 평균과 분산은 다음과 같다.

- ① 평균 :  $E(X_1 + X_2) = a\beta$
- ② 분산 :  $Var(X_1 + X_2) = a\beta^2$

3)  $\chi^2$ 분포

두 개의 독립적인 확률변수  $X_1, X_2$ 는 각각 모수  $\frac{m_1}{2}$  과  $\frac{m_2}{2}$  를 갖는  $\chi^2$ 분포를 따른다고 하자. 이 때,  $\chi^2$ 분포는 Gamma 분포의 특수한 형태로서 확률밀도함수는 다음과 같다.

$$P_{X_1}(x_1) = \frac{x_1^{\frac{m_1}{2}-1} \exp[-x_1/2]}{\beta^{\frac{m_1}{2}} \Gamma(\frac{m_1}{2})}, \quad 0 \leq x_1 < \infty \quad (9)$$

$$P_{X_2}(x_2) = \frac{x_2^{\frac{m_2}{2}-1} \exp[-x_2/2]}{\beta^{\frac{m_2}{2}} \Gamma(\frac{m_2}{2})}, \quad 0 \leq x_2 < \infty \quad (10)$$

두 개의 독립적인 확률변수  $X_1, X_2$ 의 합  $X_1 + X_2$ 의 분포는 다음과 같은  $\chi^2$  분포의 확률밀도함수이다.

$$P_{X_1+X_2}(x_1+x_2) = \frac{1}{2^{\frac{m_1+m_2}{2}} \Gamma(\frac{m_1+m_2}{2})} \times (x_1+x_2)^{\frac{m_1+m_2}{2}-1} \exp[-(x_1+x_2)/2]$$

$$, \quad 0 \leq x_1 + x_2 < \infty \quad (11)$$

이 때, 합  $X_1 + X_2$ 의 평균과 분산은 다음과 같다.

- ① 평균 :  $E(X_1 + X_2) = m_1 + m_2$
- ② 분산 :  $Var(X_1 + X_2) = 2(m_1 + m_2)$

### 3. 확률적 시간 넷

실세계의 사건이 실행되는 데 걸리는 시간은 일반적으로 상수가 아니라 평균 얼마와 같은 확률적 수입에도 불구하고, 기존의 최소 순회 시간 분석 방법은 상수만을 다룬다. 본 절에서는 시간넷의 변천에 확률적 시간을 연합한 페트리 넷을 ‘확률적 시간넷’이라 하여 <표 1>과 같이 정의하고, 확률적 시간 넷의 최소 순회시간을 시간을 분석하는 방법을 제안하고자 한다.

확률적 시간 넷의 예로 (그림 2)를 들 수 있다. (그림 2)는 (그림 1)의 시간 넷과 동일하나 변천에 연합된 지연시간이 ‘정규분포(10, 2)’와 같은 확률 분포 시간이라는 점이 다르다.

표 1. 확률적 시간 넷의 정의

확률적 시간 넷 N은 6가지 요소로 구성된다.  
 $N = (P, T, F, W, DT, M_0)$ ,  
 단,  
 $P = \{p_1, p_2, \dots, p_m\}$ 은 장소(place)라는 유한 집합,  
 $T = \{t_1, t_2, \dots, t_n\}$ 은 변천(transition)이라는 유한 집합,  
 $F$ : 유향간선이라는  $(P \times T)$ 와  $(T \times P)$ 의 합집합의 부분집합,  
 $W: F \rightarrow \{1, 2, 3, \dots\}$ 은 유향간선의 가중치를 결정하는 무게 함수,  
 $DT: T \rightarrow \{\text{정규}(\mu_1, \sigma_1^2), \text{감마}(a_1, \beta), \chi^2(m_1)\}$ ,  
 단  $\mu_1, \sigma_1^2, a_1, \beta, m_1$ 은 실수이며, 변천의 지연 시간을 결정하는 함수,  
 $M_0: P \rightarrow \{0, 1, 2, 3, \dots\}$ 은 초기에 각 장소에 놓인 토큰의 수를 결정하며, 초기 마킹(marking)이라고 함.  
 단, P와 T의 교집합은 공집합이고, P와 T의 합집합은 공집합이 아님.

확률적 시간 넷의 최소 순회 시간 분석 방법은 기존의 방법과 동일하나, 지연시간의 합을 구하는 단계에서 기존의 방법은 상수를 다루는데 반하여 확률적 시간 넷의 경우에는 통계적 확률 분포를 다룬다는 점이 다르다. 예를 들어, (그림 2)의 경우, 분석 결과는 다음과 같다:

$$\begin{aligned} & \text{Normal}(10, 2) + \text{Normal}(20, 2) \\ & + \{ \text{Normal}(10,2) + \text{Normal}(3, 0.2) \} / k \\ & = \text{Normal}(30, 4) + \text{Normal}(13, 2.2) / k. \end{aligned}$$

k가 비교적 큰 수일 경우, 위의 값은 Normal(30, 4)가 된다. Normal(30, 4)로부터 최소 순회 시간은 평균 30이고 분산이 4인 정규분포를 이루는 것을 알 수 있다. 사용자가 요구하는 처리 시간이 30 단위 시간이라면 기존의 분석 방법은 이 시스템이 요구 사항을 만족한다고 결론을 내릴 것이다. 그러나 확률적 시간 넷의 경우에는 반드시 그렇지 않다. 예를 들어, 사용자가 요구 하는 처리 시간이 95% 확률 범위 내에서 30 단위 시간일 것을 요구 한다면 확률적 시간 넷의 경우에 요구 사항을 만족시키지 못함으로 시스템의 성능을 향상시켜야 한다는 결론을 내린다.

지연 시간이 감마분포를 이룰 경우를 고려하여 보자. 두개의 독립적인 변천  $t_2$ 와  $t_4$ 에 의해 지연 시간이 각각 Gamma(6,3), Gamma(10,3) 분포를 따른다면 (그림 2)의 분석 결과는 (식 8)에 의해 Gamma(16,3)이 된다. 따라서 정규분포의 경우와 마찬가지로 사용자가 요구하는 처리 시간이 95% 확률 범위에서 48 단위 시간이라면 이 경우에도 사용자의 요구를 만족하지 못한다는 결론을 내린다.

지연 시간이  $\chi^2$  분포를 이루는 경우를 고려하여 보자. 두개의 독립적인 변천  $t_2$ 와  $t_4$ 에 의해 지연 시간이 각각  $\chi^2(6)$ ,  $\chi^2(13)$ 이라면 (그림 2)의 분석 결과는 (식 9)에 의해  $\chi^2(19)$ 가 된다. 따라서 정규분포의 경우와 마찬가지로 사용자가 요구하는 처리 시간이 95% 확률 범위에서 19 단위 시간이라면 이 경우에도 사용자의 요구를 만족하지 못한다는 결론을 내린다.

### 4. LBS 시스템에의 적용 예

위치기반서비스 가입자는 서비스 응답 시간이 일정 시간 이내이기를 기대하며, 응답 시간이 너무 느린 경우가 빈번히 발생하면 결국 탈퇴를 하게 된다.

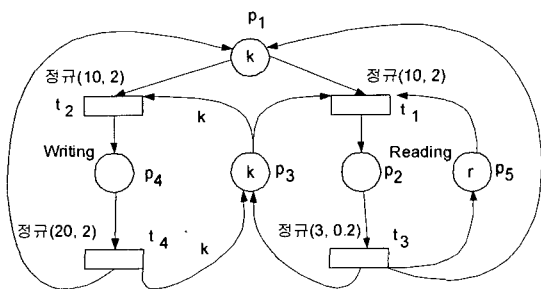


그림 2. 확률적 시간 넷의 예







결과적으로,  $(A^{-1})^T D x = (3d_1 \ 0 \ 3d_1 \ 2d_2 \ d_3 \ d_3 \ 0 \ 2d_4 \ 2d_4 \ 0 \ 2d_5 \ 2(d_2+d_5) \ 0)^T$ 이다.

$$\begin{aligned} y_1^T(A^{-1})^T D x / y_1^T M_0 &= 3d_1/n, \\ y_2^T(A^{-1})^T D x / y_2^T M_0 &= 2(d_2+d_5)/r, \\ y_3^T(A^{-1})^T D x / y_3^T M_0 &= d_3/m, \\ y_4^T(A^{-1})^T D x / y_4^T M_0 &= 2d_4/k, \\ y_5^T(A^{-1})^T D x / y_5^T M_0 &= (3d_1 + 2d_2 + d_3 + 2d_4 + 2d_5)/n. \end{aligned}$$

따라서 (그림 4)에 대한 최소 순회 시간은 다음과 같다:

$$\text{Max}\{3d_1/n \ 2(d_2+d_5)/r, \ d_3/m, \ 2d_4/k, \ (3d_1 + 2d_2 + d_3 + 2d_4 + 2d_5)/n\} \quad (\text{식 5})$$

지속시간  $d_1$ 은 사용자가 선택한 메뉴를 판단하여 위치기반 서비스 서버로 송신할 것인지를 판단하는 시간이고,  $d_2$ 는 단말기에서 통합플랫폼으로 메시지를 전달하는 통신 시간이며,  $d_5$ 는 통합플랫폼에서 단말기로 결과를 전달하는 통신시간이다.  $d_3$ 는 위치제공자의 처리시간인데, 위치제공자의 처리장치의 수가 증가하면 처리 시간이 짧아질 수 있다.  $d_4$ 는 콘텐츠 제공자의 처리시간이며, 이 시간도 역시 처리장치의 수가 증가하면 짧아질 수 있다.

예를 들어, 지연 시간과 자원의 수가 다음과 같은 경우를 고려하여 보자.

$$\begin{aligned} d_1 &= 50, \quad d_2 = 100, \quad d_3 = 500, \quad d_4 = 1,000, \\ d_5 &= 100, \quad n = 1,000 \quad k = 1, \quad m=1, \quad r=1. \end{aligned}$$

$$\begin{aligned} \text{최소 순회 시간은 다음과 같다.} \\ \text{MAX}\{0.15, 400, 500, 2,000, 3.05\} &= 2,000 \end{aligned}$$

최소 순회 시간 2,000의 3 명의 가입자를 서비스할 때 이 LBS 시스템의 반응시간이 2,000 단위 시간이며, 이는 콘텐츠 제공자에 의하여 결정된다는 것이다. 따라서 사용자가 요구하는 반응시간이 2,000 단위 시간이라면 본 시스템은 서비스의 질과 관련한 사용자의 요구 사항을 만족한다.

지연시간이 통계적 확률분포를 이루는 경우를 고려해 보자.  $d_4$ 가 평균이 1,000이고 분산이 50인 정규분포를 이룬다고 하자. 사용자가 요구하는 반응시간이 90% 범위 내에서 2,000 단위 시간이라면 이 시스템의 응답시간은 사용자의 요구사항을 만족시키지

못하기 때문에 콘텐츠 제공자를 개선하여 지연 시간을 짧게 하든지, 처리장치를 늘리는 방법을 고려해야 한다.

### 5. 결 론

실세계에서는 어떤 사건이 실행되는 데 소요되는 시간이 일반적으로 상수가 아니라 통계적 확률분포를 이룬다. 그러나 기존의 최소 순회 시간 분석에서는 시간 넷의 변천에 연합된 지연 시간이 상수인 경우만 다루고 있다. 따라서 기존의 방법으로는 '95% 확률 범위에서 응답 시간이 300 단위 시간일 것'과 같은 확률적인 분석 기준을 뒷받침할 수가 없다.

본 논문은 시간 넷의 변천에 연합된 지연 시간이 확률분포를 하는 확률적 시간 넷을 제안하고, 이것을 이용한 최소 순회 시간 분석 방법을 소개하였다. 제안하는 방법은 기존의 최소 순회 시간 분석 방법과 비슷하지만 제안하는 방법의 분석 결과는 기존의 방법의 결과와 크게 다르다. 즉, 기존의 방법의 분석 결과는 상수인데 반하여, 제안하는 방법의 분석 결과는 '주어진 시스템의 처리 시간이 95% 범위에서 300 단위 시간이다'와 같은 확률적인 형태를 띈다.

확률적인 분석 결과는 일상적으로 사용되는 퍼지 문장으로 표현된 사용자의 요구 사항을 분석하는 데에도 적용될 수 있다. 예를 들어, 다음과 같은 형태의 요구 사항을 분석 처리 할 수도 있다: '처리 시간이 대체로 300 단위 시간이라야 한다,' '처리 시간이 대부분 300 단위 시간이라야 한다,' '처리 시간이 거의 모든 경우에 300 단위 시간이라야 한다.'

최소 순회시간 분석 결과는 일목요연한 하나의 수학적식으로 표현되며, 시스템의 병목을 정확히 지적하여 준다. 따라서 시스템의 성능을 개선하려면 바로 그 병목을 개선하면 된다. 반대로 시스템의 성능이 충분할 경우에는, 병목을 기준으로 다른 처리 과정에 할당된 자원들은 절감하여도 전체시스템의 성능에 아무런 영향을 주지 않는다는 의미를 분석 결과가 나타내기 때문에 자원을 절감하는 방안도 시사한다. 즉, 최소 순회시간 분석 방법은 사용자 요구 사항을 만족하면서, 경비를 최소화하는 최적의 시스템 설비 구현 방안을 찾아주는 도구로 사용될 수 있다. 본 연구는 이러한 최소 순회 시간 분석을 지연 시간이 확률분포를 나타내는 경우로 확장하여 실용성을 보장

하였다. 향후에는 확률적인 지연 시간을 바탕으로 퍼지 시스템을 분석하는 방안을 고안하고자 한다.

### 참 고 문 헌

- [1] Murata, T., "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, Vol. 77. No. 4, pp. 541-580, April 1989.
- [2] Hillion, Hervé P., "Timed Petri Nets and Application to Multi-Stage Production Systems.," *Lecture Notes in Computer Science*, Vol. 424; *Advances in Petri Nets 1989*, Berlin, Germany, Springer-Verlag, pp. 281-305, 1990.
- [3] Dasdan, A. and R.K. Gupta, "Faster maximum and minimum mean cycle algorithms for system performance analysis," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 17, No. 10, pp. 889-899, 1998.
- [4] Nakamura, M. and M. Silva, "Cycle time computation in deterministically timed weighted marked graphs," *Proceedings. ETFA '99. 7th IEEE International Conference on Emerging Technologies and Factory Automation 1999*, Vol. 2, pp. 1037-1046, Oct. 18-21, 1999.
- [5] Rodriguez-Beltran, J. and A. Ramfrez-Trevino, "Minimum initial marking in timed marked graphs," *2000 IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 4, pp. 3004-3008, Oct. 8-11, 2000.
- [6] Peterson, J. L., *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, N.J., 1981, ISBN: 0-13-661983-5.
- [7] Reisig, W., *Petri Nets, An Introduction, EATCS, Monographs on Theoretical Computer Science*, W.Brauer, G. Rozenberg, A. Salomaa (Eds.), Springer Verlag, Berlin, 1985.
- [8] Desel, J., Wolfgang Reisig, and Grzegorz Rozenberg (Eds.), *Lectures on Concurrency and Petri Nets, Advances in Petri Nets, Lecture Notes in Computer Science*, Vol. 3098, Springer-Verlag, 2004, ISBN: 3-540-22261-8.
- [9] Kotz, J., "Distributions in Statistics - Continuous Univariate Distributions-1,2", John Wiley & Sons, 1970, ISBN 0-471-44626-2.
- [10] Krishnamurthy, N., "Using SMS to Deliver Location-based Services," *Proceedings of 2002 IEEE International Conference on Personal Wireless Communications (Proceedings of ICPWC'2002)*, Dec. 15-17, 2002, pp. 177-181.

#### 임 재 결



1974년 인천교육대학 졸업  
 1981년 동국대학교 전자계산학과 졸업  
 1987년 일리노이대학교 시카고 캠퍼스 컴퓨터과학 석사  
 1990년 일리노이대학교 시카고 캠퍼스 컴퓨터과학 박사  
 1992년~현재 동국대학교 과학기술대학 컴퓨터멀티미디어 학과 교수  
 관심분야: 시스템 설계 및 분석, 인공지능, 페트리 넷 이론 및 응용.

#### 심 규 박



1984년 동국대학교 통계학과 졸업  
 1986년 동국대학교 대학원 통계학과 이학석사  
 1993년 동국대학교 대학원 통계학과 이학박사  
 1994년~현재 동국대학교 과학기술대학 정보통계학과 교수  
 관심분야: 전산통계, 신뢰도검정, 통계 자료분석.